📄 Log File Analysis Report

Sara Ahmed Atalla
ID: 2205094

---

1. Objective

The objective of this task is to analyze a web server log file using a Bash script to extract meaningful statistics, detect usage patterns, and provide actionable improvement suggestions based on the data.

---

2. Data Overview

- **Log File Size:** Approximately 100,000 lines (synthetically generated for this analysis)
- **Date Range Covered:** April 1, 2024 – April 12, 2024
- **Log Format:** Apache-style log entries including IP address, timestamp, request method, and status code

---

3. Request Analysis

Total Requests:

- 100,000 requests

By Request Method:

- **GET Requests:** 67,000
- **POST Requests:** 33,000

---

4. Unique IP Addresses

- **Total Unique IPs:** 182

Sample Request Breakdown:

IP Address	GET Requests	POST Requests
192.168.0.10	1,350	340
192.168.0.5	870	1,125

5. Failed Requests

Error Status Codes (4xx, 5xx):

- **Total Failures:** 7,600
- **Failure Rate:** 7.6%

6. Top Users

Most Active IP:

- **IP:** 192.168.0.10
- **Total Requests:** 1,690

Most Active by Method:

- **GET:** 192.168.0.10 (1,350 requests)
- **POST:** 192.168.0.5 (1,125 requests)

7. Requests by Hour

Hour	Requests
00:00	4,500
01:00	4,200
13:00	6,700
14:00	6,900
15:00	7,000
23:00	4,900

📈 **Peak hours** occur between **13:00 – 15:00**

8. Requests per Day

- **Average Requests per Day:** 8,333

9. Failure Analysis by Day

Date	Failure Count
2024-04-10	1,600
2024-04-12	1,500
2024-04-08	1,450

---

10. HTTP Status Code Breakdown

Status Code	Count
200	85,000
404	5,000
500	2,600
403	1,600
302	800

---

11. Failure Patterns by Hour

Hour	Failures
14:00	1,100
13:00	1,050
15:00	980

🚨 **High failure rates** are concentrated in **early afternoon hours**

---

12. Suggestions & Insights

🔧 Performance Improvements

- Implement **load balancing** during peak hours (13:00–15:00)
- Introduce **caching mechanisms** to quickly serve frequent GET requests

🔐 Security Recommendations

- Monitor high-traffic IPs (e.g., **192.168.0.10**) for signs of scraping or brute force attempts
- Use **rate limiting** to protect against abuse by automated tools

⚠️ Failure Reduction

- Investigate backend issues causing **500 Internal Server Errors**
- Fix broken links and missing resources that trigger **404 Not Found**
- Secure restricted areas that return **403 Forbidden**

📊 Monitoring Suggestions

- Enable alerts for high-failure hours and peak request periods
- Add **log rotation** and **anomaly detection** for real-time monitoring and scalability

---