



BLOGGING PLATFORM

CONNECTCO



SOFTWARE ENGINEERING (IT - 314)

Prof. Saurabh Tiwari

Group 23

- 202201287 GHORI ZEEL JIVRAJBHAI (**Group Leader**)
- 202201291 PATEL NAISARGI NILANGBHAI
- 202201272 SIDDHANT GUPTA
- 202201322 ADITYA IYER
- 202201338 KRITARTH JOSHI
- 202201312 DESAI PRITISH
- 202201299 AKSHAT JINDAL
- 202201324 KASHYAP GAJERA
- 202201267 CHAUHAN HEMAN JASHVANTBHAI
- 202201265 CHAUDHARI PRACHI HASMUKHBHAI

IT313 Software Engineering

Non-Functional Testing



Software Engineering Project

Group 23

December 1, 2024

Non-functional testing evaluates the quality attributes of a software system that are not directly tied to its functional requirements. These attributes include:

- **Performance**
- **Reliability**
- **Scalability**
- **Usability**
- **Maintainability**
- **Security**

The primary objectives of non-functional testing are to enhance the product's usability, efficiency, maintainability, and portability; reduce risks and costs associated with production; streamline installation, setup, execution, management, and monitoring processes; and gather metrics for internal research and development. Additionally, it seeks to deepen understanding of the product's behavior and the technologies involved.

Types of Non-Functional Testing:

- **Load and Stress Testing**
- **Performance Testing**
- **Security Testing**

- **Compatibility Testing**
- **Reliability and Availability Testing**

1. Load and Stress Testing

Here we have performed the load testing **in Jmeter**.

In this scenario, load testing was conducted using JMeter. A JMeter Test Plan requires a listener to display the performance test results.

Role of Listeners in JMeter:

- Listeners capture server responses during test execution and present the results in various formats such as trees, tables, graphs, and log files.
- They also allow saving these results for future reference.
- JMeter offers several types of listeners, including **Summary Report, Aggregate Report, Aggregate Graph, View Results Tree, and View Results in Table**, among others.

Detailed Explanation of Summary Report Parameters:

- **Label:** Represents the name or URL of a specific HTTP(s) request. If the "Include group name in label?" option is selected, the Thread Group name is prefixed to each label.
- **Samples:** Indicates the number of virtual users executing a request.
- **Average:** The mean time taken by all samples to complete the request for a specific label.
- **Min:** The shortest response time recorded for a specific label.
- **Max:** The longest response time recorded for a specific label.
- **Std. Dev.:** Reflects the deviation of sample response times from the average. Lower standard deviation indicates more consistent data. Ideally, it should be less than or equal to half the average response time for a label.
- **Error%:** The percentage of failed requests for a specific label.
- **Throughput:** Represents the number of requests processed by the server per time unit (seconds, minutes, or hours). It is measured from the first sample's start to the last sample's end. Higher throughput indicates better performance.
- **KB/Sec:** The volume of data downloaded from the server during the test, expressed in kilobytes per second. It is essentially the throughput measured in KB/s.

Test Scenario:

HTTP GET Request for various web pages:

1. **Dashboard**

2. Profile
3. Blogg Page
4. Login
5. Landing Page

1. Dashboard Page

Summary Report

Name:
Summary Report - Dashboard

Comments:

Write results to file / Read from file

Filename:
Browse...
Log/Display Only:
☐ Errors
☐ Successes
Configure

| Label | # Samples | Average | Min | Max | Std. Dev. | Error % | Throughput | Received KB/sec | Sent KB/sec | Avg. Bytes |
|--------------|-----------|---------|-----|-------|-----------|---------|------------|-----------------|-------------|------------|
| HTTP Request | 400 | 3255 | 39 | 10611 | 2974.83 | 0.00% | 33.7/sec | 235.45 | 4.70 | 7157.0 |
| TOTAL | 400 | 3255 | 39 | 10611 | 2974.83 | 0.00% | 33.7/sec | 235.45 | 4.70 | 7157.0 |

| Number of Users | Ramp-up Period | Loop Count | Total Samples (Users*Loops) | Average Error(In %) | Throughput (Response/s) |
|-----------------|----------------|------------|-----------------------------|---------------------|-------------------------|
| 200 | 2 | 2 | 400 | 0 | 33.7 |
| 400 | 2 | 2 | 800 | 7.12 | 11.1 |
| 600 | 2 | 2 | 1200 | 44.8 | 14 |

2. Profile Page

| Label | # Samples | Average | Min | Max | Std. Dev. | Error % | Throughput | Received KB/sec | Sent KB/sec | Avg. Bytes |
|--------------|-----------|---------|-----|------|-----------|---------|------------|-----------------|-------------|------------|
| HTTP Request | 400 | 1383 | 52 | 7010 | 1179.70 | 0.00% | 36.5/sec | 235.76 | 5.03 | 6615.0 |
| TOTAL | 400 | 1383 | 52 | 7010 | 1179.70 | 0.00% | 36.5/sec | 235.76 | 5.03 | 6615.0 |

| Number of Users | Ramp-up Period | Loop Count | Total Samples (Users*Loops) | Average Error(In %) | Throughput (Response/s) |
|-----------------|----------------|------------|-----------------------------|---------------------|-------------------------|
| 200 | 2 | 2 | 400 | 0 | 36.5 |
| 400 | 2 | 2 | 800 | 8.25 | 11.6 |
| 600 | 2 | 2 | 1200 | 56.08 | 14 |

3. Blog Page

| Label | # Samples | Average | Min | Max | Std. Dev. | Error % | Throughput | Received KB/sec | Sent KB/sec | Avg. Bytes |
|--------------|-----------|---------|-----|------|-----------|---------|------------|-----------------|-------------|------------|
| HTTP Request | 400 | 1103 | 308 | 3002 | 651.57 | 0.00% | 80.6/sec | 573.52 | 11.66 | 7282.4 |
| TOTAL | 400 | 1103 | 308 | 3002 | 651.57 | 0.00% | 80.6/sec | 573.52 | 11.66 | 7282.4 |

| Number of Users | Ramp-up Period | Loop Count | Total Samples (Users*Loops) | Average Error(In %) | Throughput (Response/s) |
|-----------------|----------------|------------|-----------------------------|---------------------|-------------------------|
| 200 | 2 | 2 | 400 | 0 | 80.6 |
| 400 | 2 | 2 | 800 | 13 | 9.3 |
| 600 | 2 | 2 | 1200 | 65.08 | 13.9 |

4. Login Page

| Label | # Samples | Average | Min | Max | Std. Dev. | Error % | Throughput | Received KB/sec | Sent KB/sec | Avg. Bytes |
|--------------|-----------|---------|-----|------|-----------|---------|------------|-----------------|-------------|------------|
| HTTP Request | 400 | 1077 | 35 | 4730 | 818.02 | 0.00% | 61.3/sec | 426.21 | 8.33 | 7114.0 |
| TOTAL | 400 | 1077 | 35 | 4730 | 818.02 | 0.00% | 61.3/sec | 426.21 | 8.33 | 7114.0 |

| Number of Users | Ramp-up Period | Loop Count | Total Samples (Users*Loops) | Average Error(In %) | Throughput (Response/s) |
|-----------------|----------------|------------|-----------------------------|---------------------|-------------------------|
| 200 | 2 | 2 | 400 | 0 | 61.3 |
| 400 | 2 | 2 | 800 | 9.62 | 11.7 |
| 600 | 2 | 2 | 1200 | 48.5 | 12.7 |

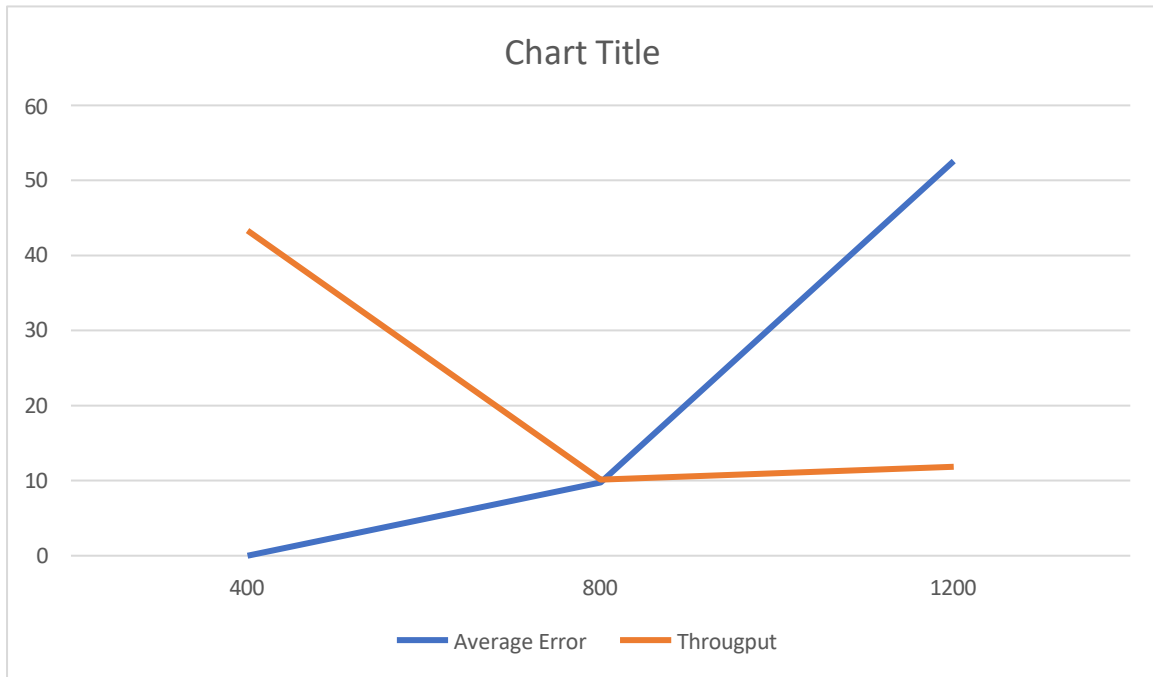
5. Landing Page

| Label | # Samples | Average | Min | Max | Std. Dev. | Error % | Throughput | Received KB/sec | Sent KB/sec | Avg. Bytes |
|--------------|-----------|---------|-----|-------|-----------|---------|------------|-----------------|-------------|------------|
| HTTP Request | 400 | 17568 | 191 | 85835 | 12829.99 | 0.00% | 4.5/sec | 100.76 | 0.63 | 22711.3 |
| TOTAL | 400 | 17568 | 191 | 85835 | 12829.99 | 0.00% | 4.5/sec | 100.76 | 0.63 | 22711.3 |

| Number of Users | Ramp-up Period | Loop Count | Total Samples (Users*Loops) | Average Error(In %) | Throughput (Response/s) |
|-----------------|----------------|------------|-----------------------------|---------------------|-------------------------|
| 200 | 2 | 2 | 400 | 0 | 4.5 |
| 400 | 2 | 2 | 800 | 10.9 | 7 |
| 600 | 2 | 2 | 1200 | 48.45 | 4.6 |

Average Analysis:

| Number of Users | Ramp-up Period | Loop Count | Total Samples (Users*Loops) | Average Error(In %) | Throughput (Response/s) |
|-----------------|----------------|------------|-----------------------------|---------------------|-------------------------|
| 200 | 2 | 2 | 400 | 0 | 43.32 |
| 400 | 2 | 2 | 800 | 9.78 | 10.14 |
| 600 | 2 | 2 | 1200 | 52.58 | 11.84 |



2. PERFORMANCE TESTING

Objective: To assess and test system's responsiveness, scalability and performance under various conditions

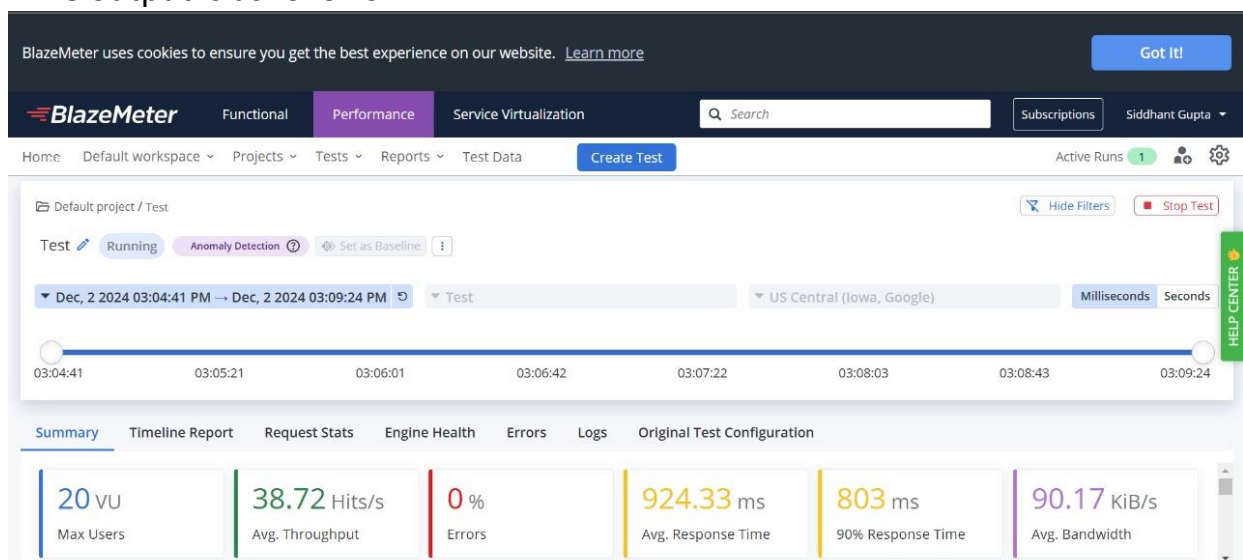
Tools used: **Blaze-Meter**

The test is conducted under the following conditions:

- Duration: 8 minutes
- Test Type: JMeter
- No. of CPUs: 2
- Fixed Location

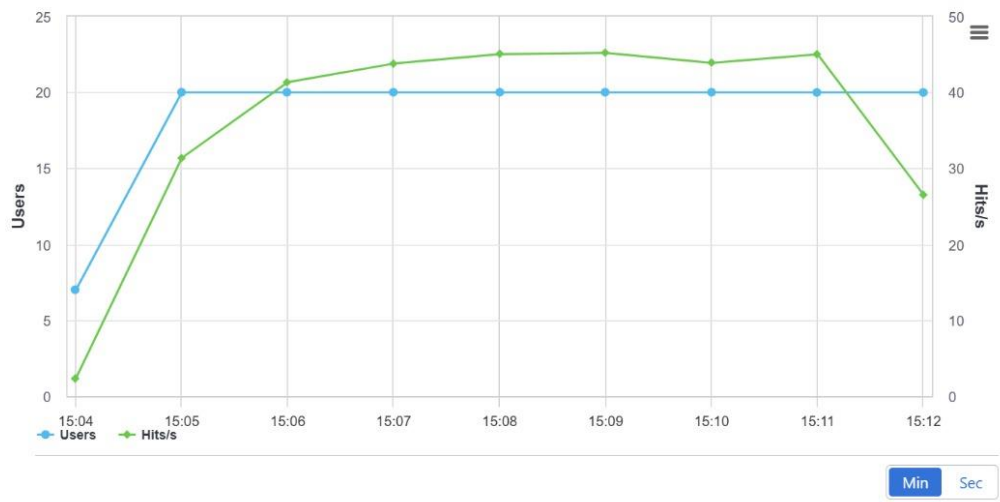
| | | | | | |
|----------|--------------------------|----------------|-----------------------------------|---------------|--|
| Duration | 8 minutes | Test Type | JMeter | Tags | <input type="text" value="Define Tags..."/> |
| Started | Dec 02, 2024, 3:04:41 PM | Response Codes | 2xx 3xx | Report Notes: | <input type="text" value="Enter report notes..."/> |
| ETA | in 13 minutes | Locations | US Central (Iowa, Google) | | <input type="button" value="Save Note"/> |

The output is as follows:

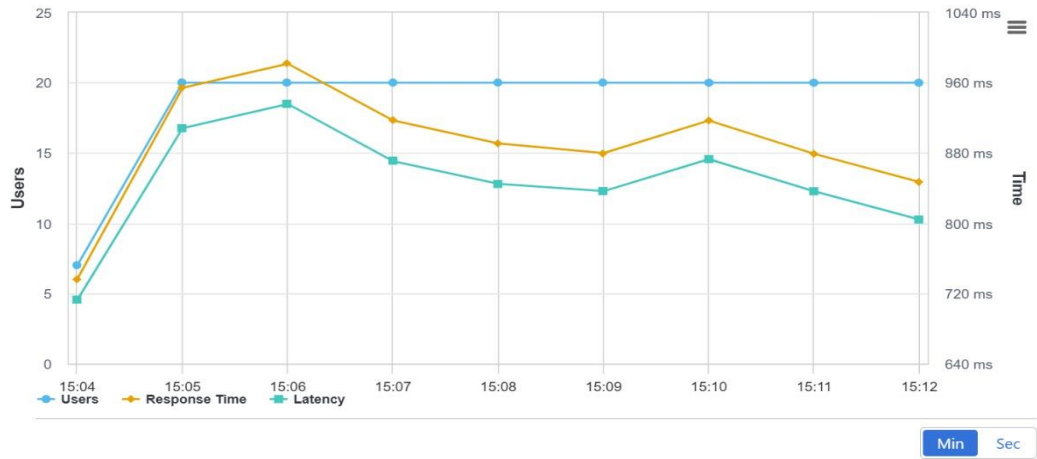


- The website can handle up to 38.72 hits / sec.
- The error rate according to the no. of users is quite low, giving an average bandwidth of 90.17 KB/s.
- Average response time of all users is 803 ms

TIMELINE GRAPHS
Concurrent Users and Hits/s

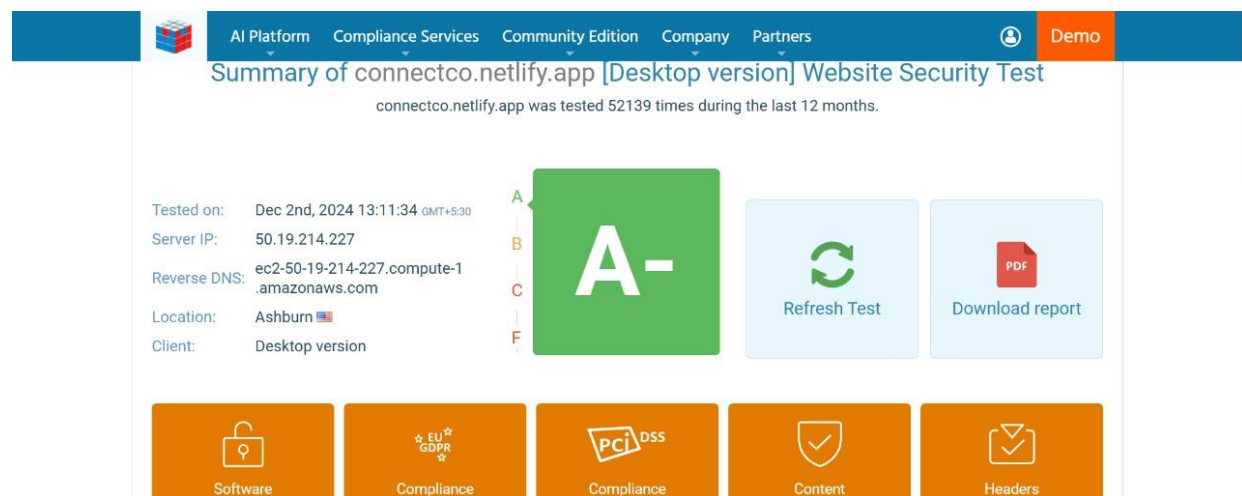


TIMELINE GRAPHS
Concurrent Users and Response Time



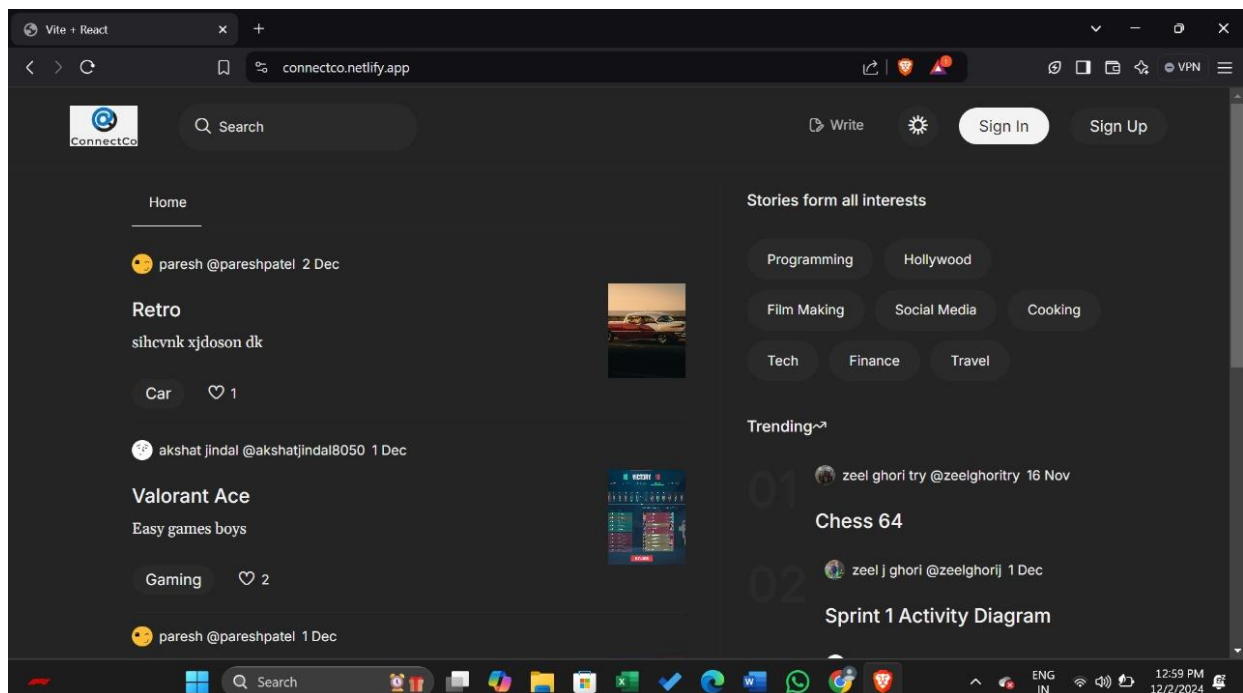
3. Security

Security testing is a critical process aimed at identifying and addressing vulnerabilities in a software system to ensure the protection of data and resources from potential threats or unauthorized access. This type of testing evaluates various aspects, including authentication, authorization, data integrity, and confidentiality, to safeguard systems against cyberattacks such as SQL injection, cross-site scripting, and malware exploitation. Tools like **ImmuniWeb** play a pivotal role in this domain, offering comprehensive security solutions that integrate automated vulnerability scanning with human-led penetration testing. ImmuniWeb provides real-time risk scoring, detailed insights, and compliance monitoring to help organizations fortify their digital infrastructure while ensuring adherence to regulatory standards. Its AI-powered platform streamlines the security testing process, making it an efficient and effective choice for businesses aiming to mitigate security risks proactively.



4. Compatibility Testing

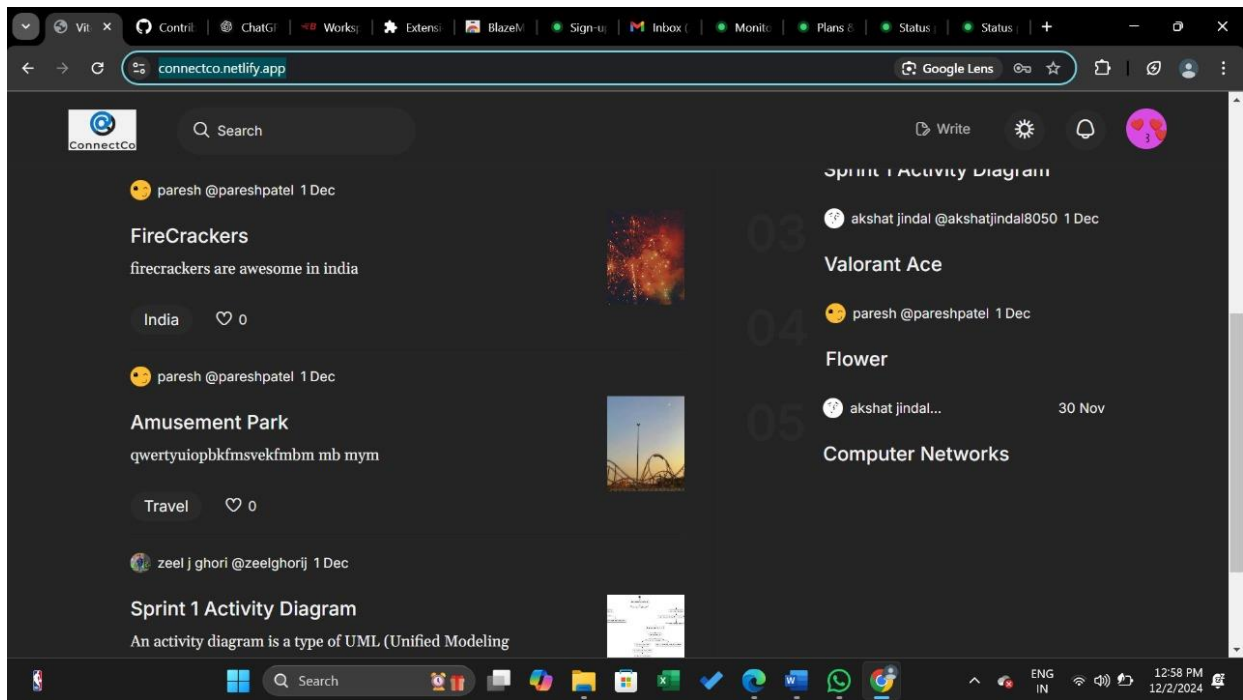
Compatibility Testing is a software testing process designed to confirm that an application or system functions correctly across diverse environments, including different hardware, operating systems, browsers, networks, and devices. Its primary objective is to ensure the software is compatible with external factors it may encounter in real-world scenarios. This testing helps uncover issues such as rendering problems, performance discrepancies, or feature failures caused by differences in configurations or platforms. By addressing these concerns, compatibility testing enhances user experience and ensures the application is accessible to a wider audience.



Resolution- 1920*1080

Browser:- Brave

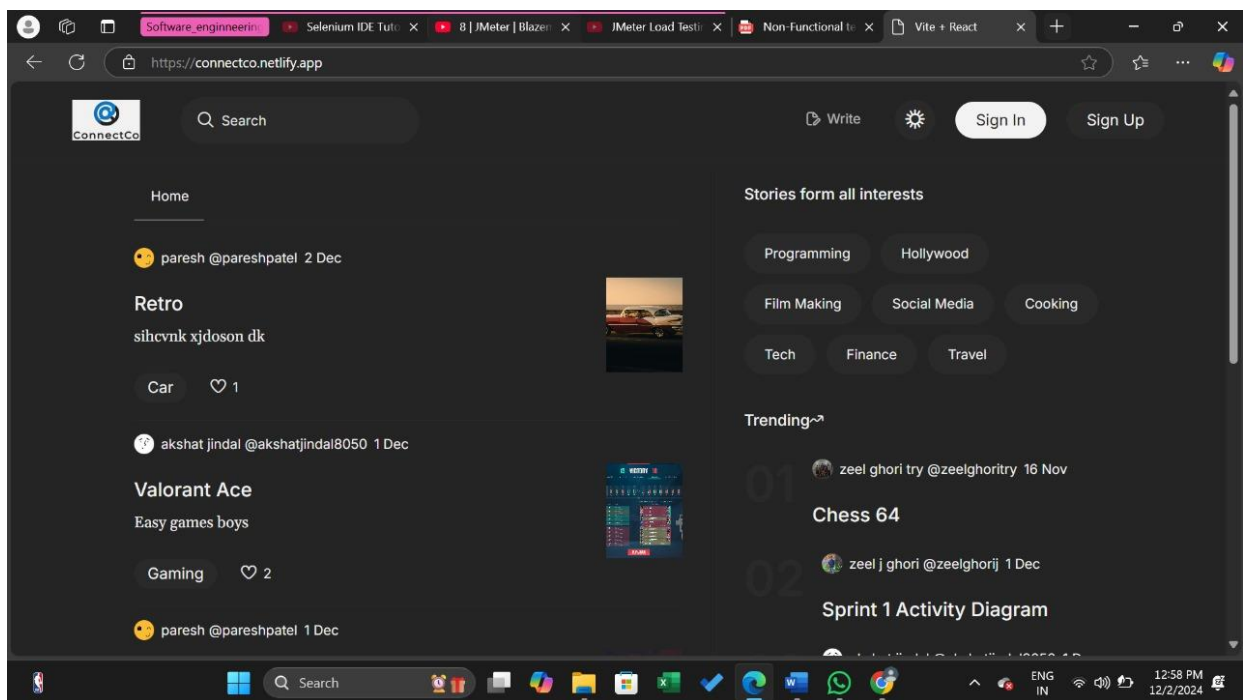
System : Windows 10 64- bit



Resolution- 1920*1080

Browser:- Chrome

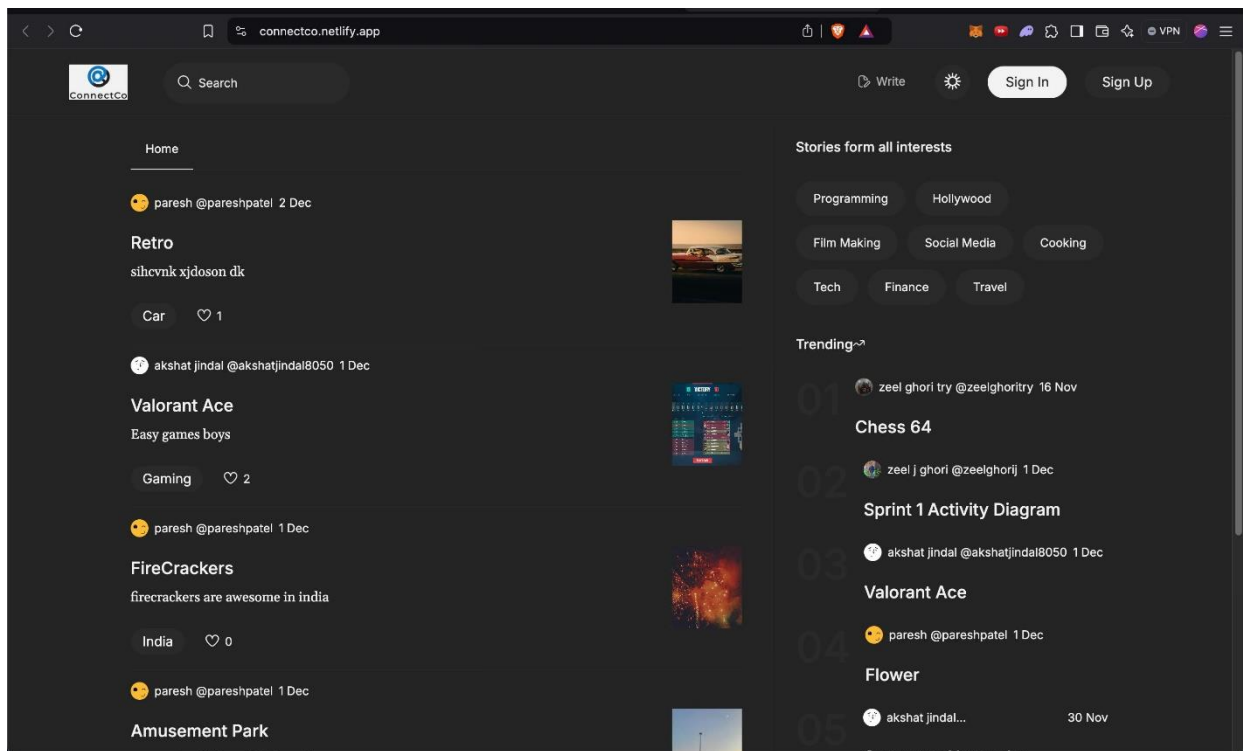
System : Windows 10 64- bit



Resolution- 1920*1080

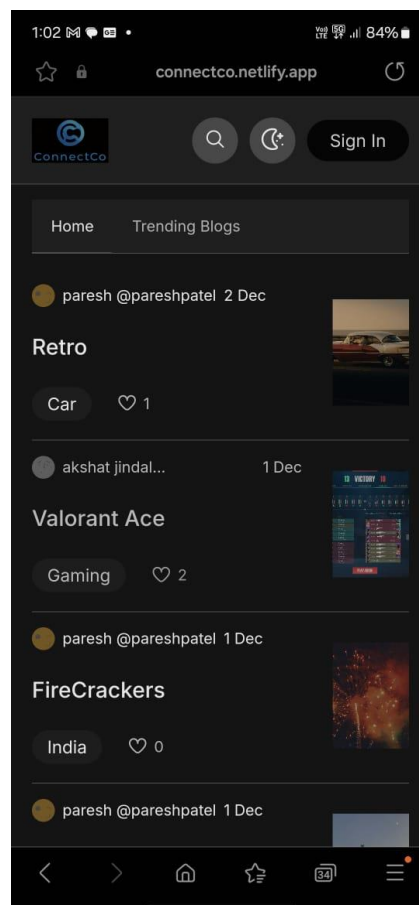
Browser:- Internet Explorer

System : Windows 10 64- bit



Resolution- 2560*1664

Version:- safari 17.3 (19617.2.4.11.8)



Resolution- 1080 x 2340 pixels,

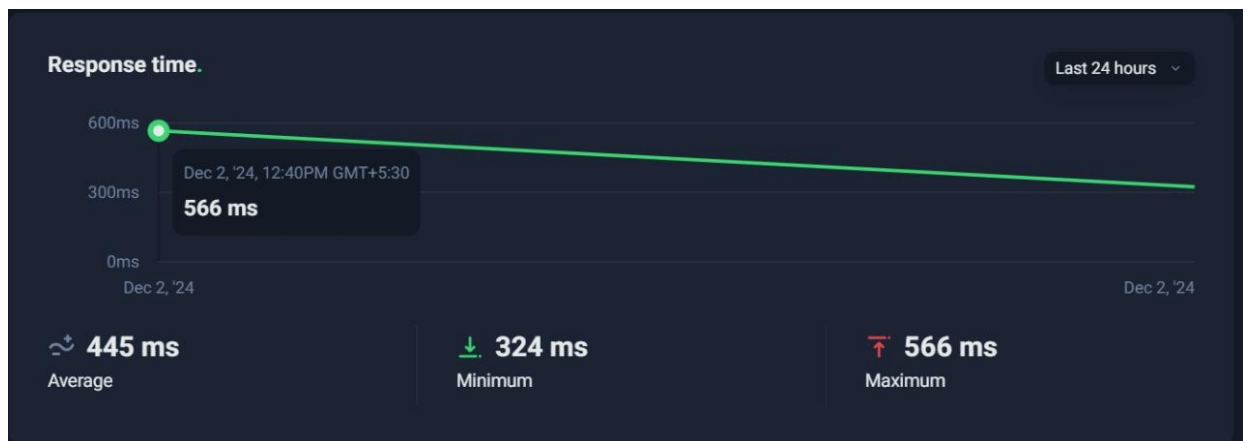
Version: Samsung Internet

5. Availability and Reliability Testing

Objective: To verify the system's reliability and availability under varying conditions.

Tool Used: UpTime Robot

The monitoring process was successfully implemented, ensuring the website remained operational 24/7. Throughout the testing period, no issues were detected, even with continuous usage by multiple users over extended durations. The results confirm the website's stability and performance under constant load.



HTTP Request v/s Time

- No major drops of response time throughout 24 hours.
- Users: 10-20, working parallel