
PROJECT: EXPENSIFY

IT314 – SOFTWARE ENGINEERING

GROUP 26

TASK : Non Functional Testing



Done by: Kkavy Dave – 202201421

GROUP MEMBERS:

ID	NAME
202201410	SIDDHANT KOTAK
202201419	SHAH MAAHI BIJAL
202201421	KKAVY KAPIL DAVE
202201440	MUNJAPARA DHARMIL JAYANTILAL
202201446	SONI DEVARSH SUSHANT
202201449	LATHIGARA PRIYANSH HARESHBHAI
202201452	DETROJA AVI ASHISHBHAI
202201456	MANAV TIRATH PAREKH
202201458	MAKWANA SRUSHTI NARESHKUMAR

Non-Functional Testing

Non-functional testing for the **Expensify Expense Tracker App** focuses on evaluating system characteristics that are not directly linked to specific functionalities but are critical for ensuring superior performance, reliability, and user experience.

This testing examines aspects like responsiveness, scalability, security, usability, and adaptability under various conditions. By evaluating these non-functional aspects, the process strengthens the app's reliability and optimizes its capabilities to consistently meet or exceed user expectations, ensuring ease of use for users.

Non-Functional Requirements

1. **Performance:** The app must provide quick responses to user actions and ensure efficient data processing.
2. **Reliability:** Guarantee high uptime and data accuracy to prevent interruptions in expense tracking and financial management.
3. **Scalability:** The app should seamlessly support an increasing number of users as its adoption grows.
4. **Usability:** The interface should be intuitive and simple, making it easy for all users to navigate and utilize its features effectively.
5. **Mobile Accessibility:** Ensure the app is optimized for smooth performance and usability on mobile devices.

Objectives

1. Non-Functional Testing enhances the configuration, implementation, oversight, and monitoring of the application.
2. It improves the app's usability, effectiveness, maintainability, and portability.
3. It ensures an enhanced and seamless user experience, making the app easy to use for all users.

Usability Testing

This testing evaluates how simple it is to use the app, focusing on its design and interface. It includes assessments of user experience, accessibility, and adaptability to different user needs.

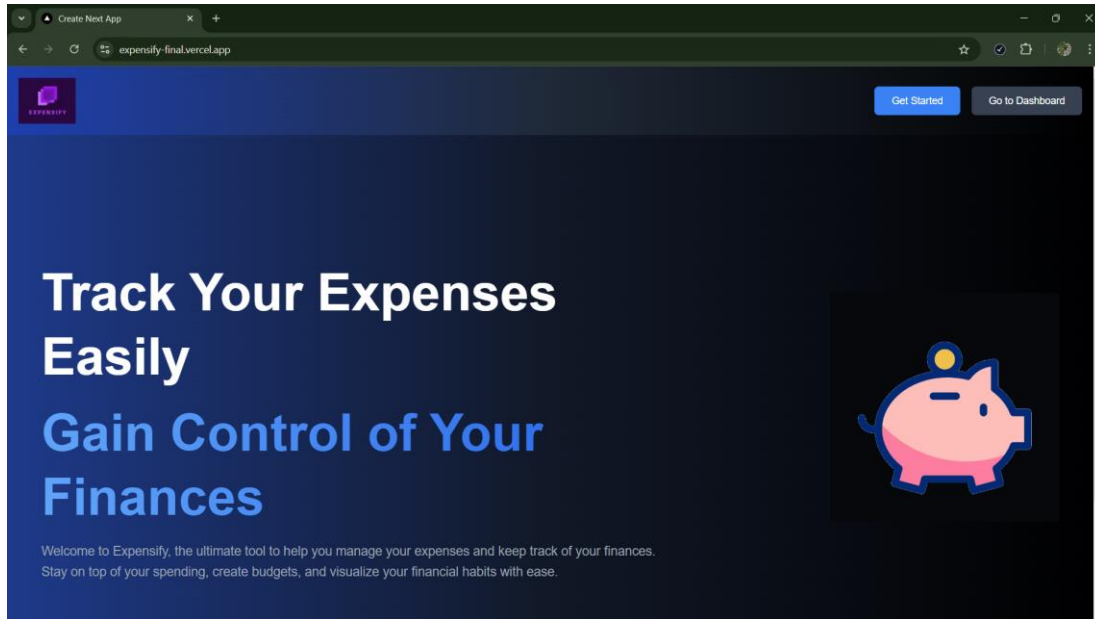
As demonstrated through GUI and acceptance testing, **Expensify** is highly user-friendly, thanks to its intuitive interface and thoughtful design.

Compatibility Check

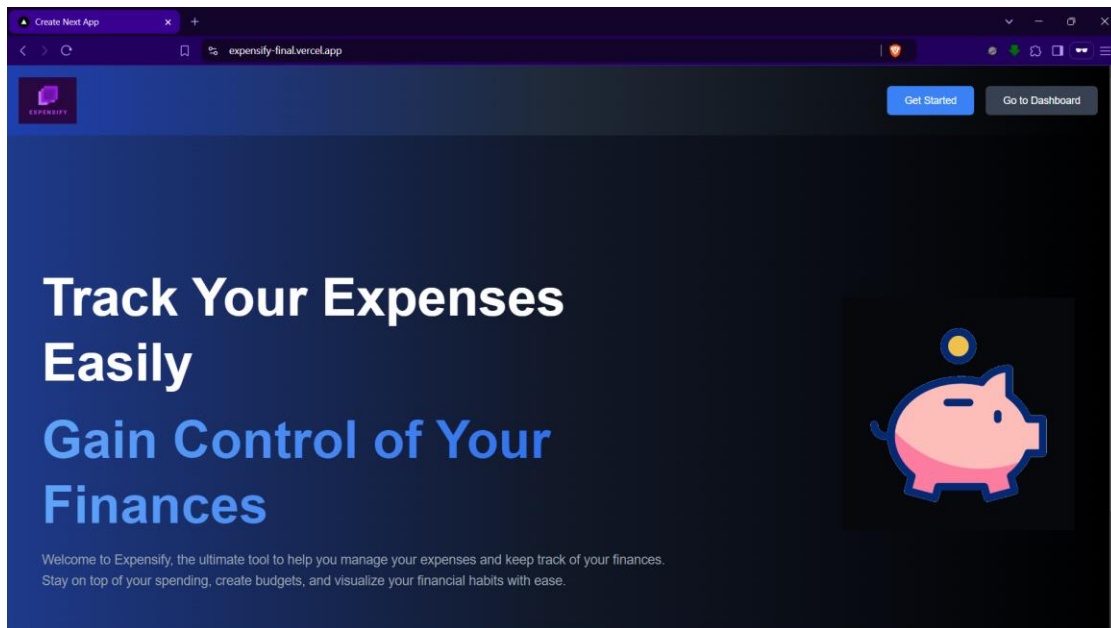
We have ensured that the app functions flawlessly across all major web browsers, including Edge, Chrome, and Brave, on windows along with android and iOS systems ensuring accessibility and convenience for users.

1. Compatibility Testing

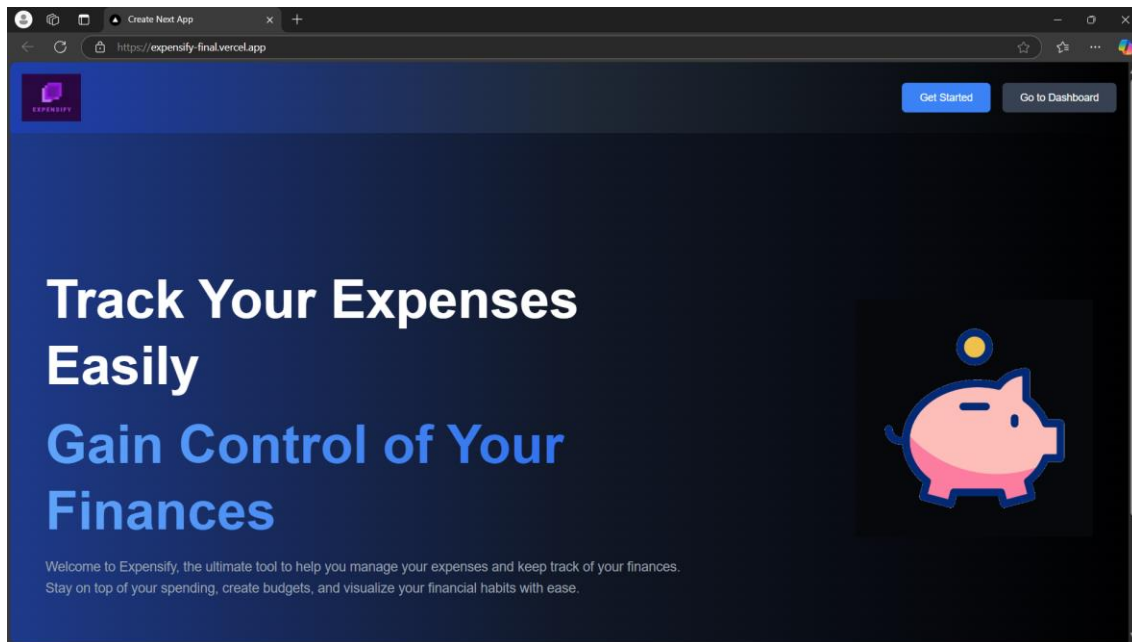
Chrome (Windows)



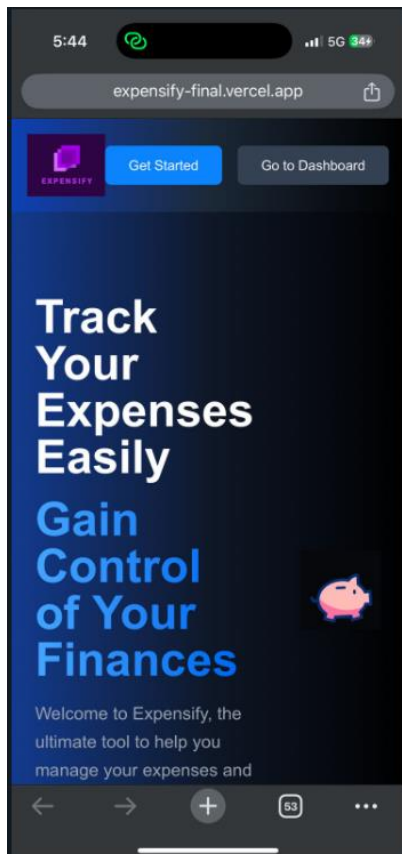
Brave (Windows)



Microsoft Edge (Windows)



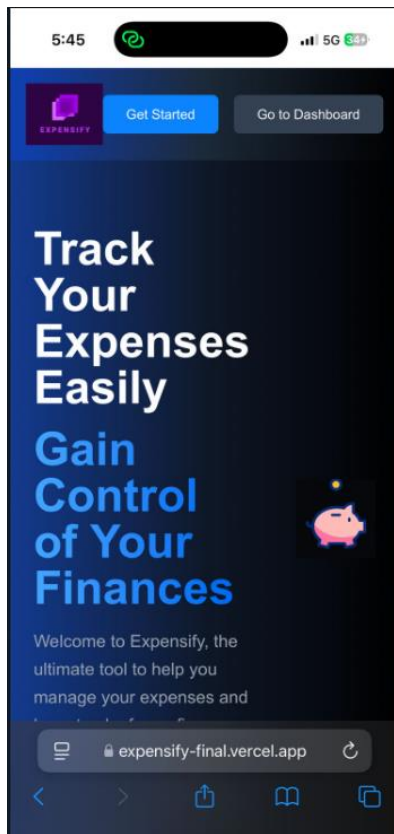
Chrome(iOS)



Chrome (Android)



Safari (iOS)



2. Security Testing

Authentication and Authorization:

Our system integrates **Clerk authentication** to manage user access. Clerk provides a secure and seamless experience through options like passwordless login (magic links and OTPs), social logins (Google, GitHub, etc.), and multi-factor authentication (MFA). Only authenticated users can access sensitive areas, ensuring robust data protection.

Data Security:

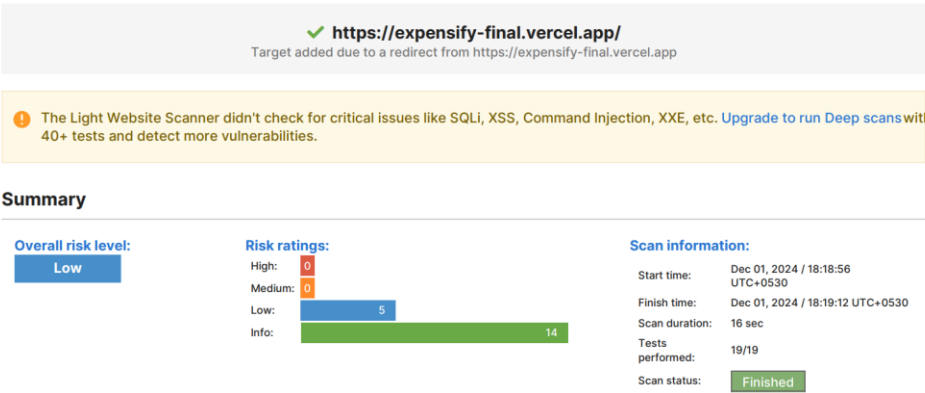
Clerk ensures the encryption of sensitive user data, such as session tokens, and supports advanced security protocols like WebAuthn for biometric logins. These measures safeguard user credentials and prevent unauthorized access.

Session Management:

Sessions are managed securely with token lifecycles, supporting features like automatic refresh tokens for uninterrupted user experience. Clerk also integrates easily with modern frameworks like Next.js to handle session persistence.

Security Testing:

We have also scanned our website using a third-party tool, <https://pentest-tools.com/>, to identify and address potential vulnerabilities, ensuring an additional layer of security.



🚩 Website is accessible.

🚩 Nothing was found for vulnerabilities of server-side software.

🚩 Nothing was found for client access policies.

🚩 Nothing was found for robots.txt file.

🚩 Nothing was found for use of untrusted certificates.

🚩 Nothing was found for enabled HTTP debug methods.

🚩 Nothing was found for enabled HTTP OPTIONS method.

🚩 Nothing was found for secure communication.

🚩 Nothing was found for directory listing.

🚩 Nothing was found for missing HTTP header - Strict-Transport-Security.

🚩 Nothing was found for domain too loose set for cookies.

🚩 Nothing was found for HttpOnly flag of cookie.

🚩 Nothing was found for Secure flag of cookie.

Scan coverage information

List of tests performed (19/19)

- ✓ Starting the scan...
- ✓ Checking for missing HTTP header - Referrer...
- ✓ Checking for missing HTTP header - X-Content-Type-Options...
- ✓ Checking for unsafe HTTP header Content Security Policy...
- ✓ Checking for website technologies...
- ✓ Checking for missing HTTP header - Content Security Policy...
- ✓ Checking for vulnerabilities of server-side software...
- ✓ Checking for client access policies...
- ✓ Checking for robots.txt file...
- ✓ Checking for absence of the security.txt file...
- ✓ Checking for use of untrusted certificates...
- ✓ Checking for enabled HTTP debug methods...
- ✓ Checking for enabled HTTP OPTIONS method...
- ✓ Checking for secure communication...
- ✓ Checking for directory listing...
- ✓ Checking for missing HTTP header - Strict-Transport-Security...
- ✓ Checking for domain too loose set for cookies...
- ✓ Checking for HttpOnly flag of cookie...
- ✓ Checking for Secure flag of cookie...

Scan parameters

target: https://expensify-final.vercel.app/
scan_type: Light
authentication: False

Scan stats

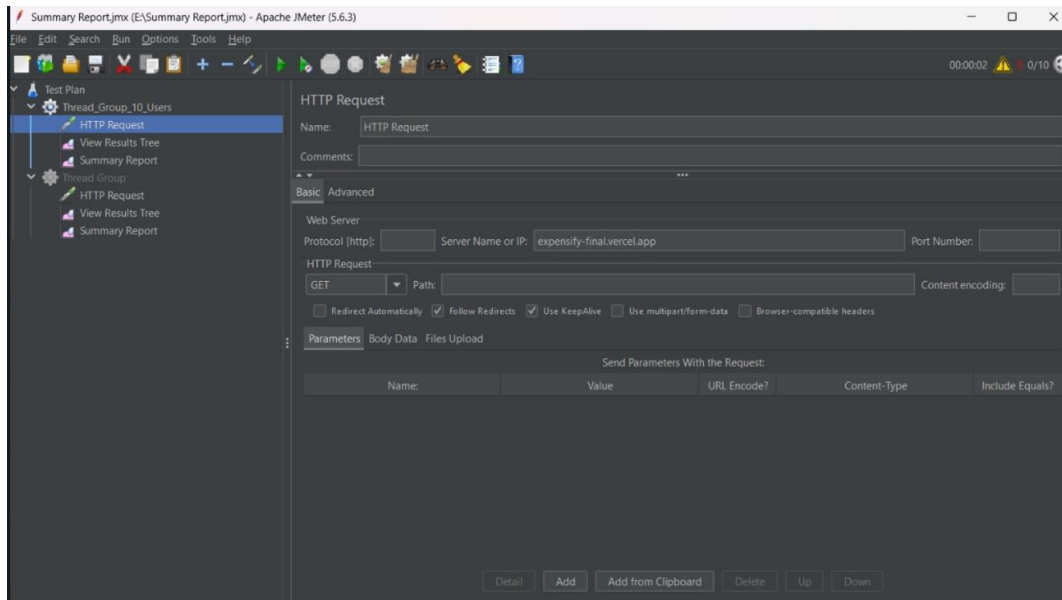
Unique Injection Points Detected:	4
URLs spidered:	24
Total number of HTTP requests:	34
Average time until a response was received:	125ms

3. Load Testing and Stress Testing (ApacheJmeter)

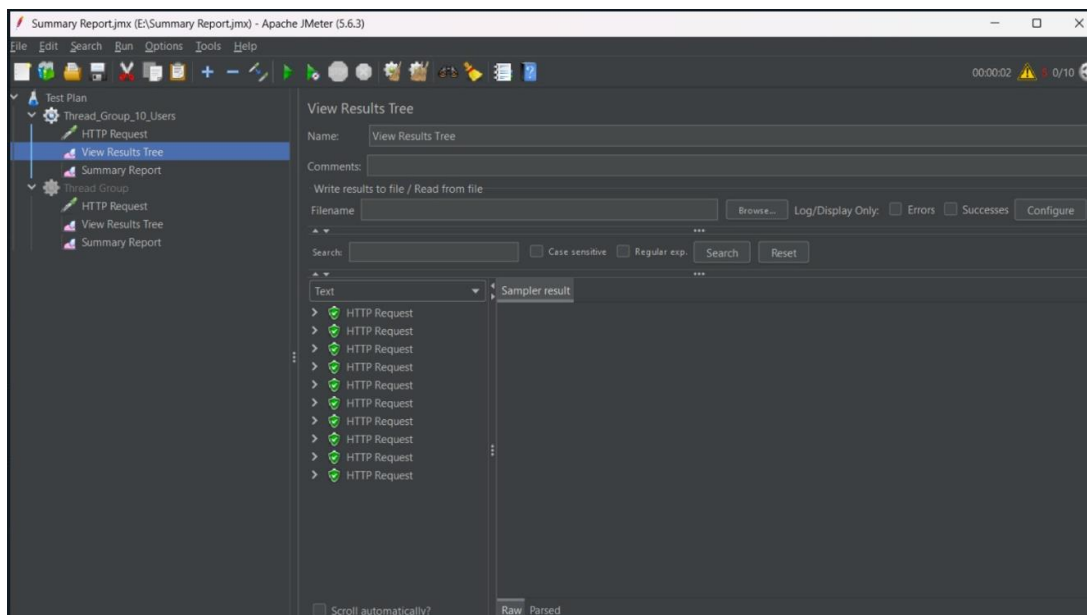
- Testing Home Module

GET request to PATH 'expensify-final.vercel.app/'

Scenario: We have accessed the home page of Expensify with the same user from multiple hosts (10, 100 and 500) and will be measuring the load balance and throughput of the system.



Summary Outputs for the following and results



Summary Reportjmx (E:\Summary Reportjmx) - Apache JMeter (5.6.3)

File Edit Search Run Options Tools Help

00:00:39 0/100

Test Plan

- Thread_Group_10_Users
 - HTTP Request
 - View Results Tree
 - Summary Report
- Thread_Group_100_Users
 - HTTP Request
 - View Results Tree
 - Summary Report
- Thread_Group_500_Users
 - HTTP Request
 - View Results Tree
 - Summary Report
- Thread_Group_1000_Users
 - HTTP Request
 - View Results Tree
 - Summary Report

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

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

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received K...	Sent KB/sec	Avg. Bytes
HTTP Reque...	100	19771	2110	38549	17621.83	0.00%	2.6/sec	36.13	0.64	14483.8
TOTAL	100	19771	2110	38549	17621.83	0.00%	2.6/sec	36.13	0.64	14483.8

☐ Include group name in label? ☒ Save Table Header

Summary Reportjmx (E:\Summary Reportjmx) - Apache JMeter (5.6.3)

File Edit Search Run Options Tools Help

00:01:16 0/500

Test Plan

- Thread_Group_10_Users
 - HTTP Request
 - View Results Tree
 - Summary Report
- Thread_Group_100_Users
 - HTTP Request
 - View Results Tree
 - Summary Report
- Thread_Group_500_Users
 - HTTP Request
 - View Results Tree
 - Summary Report

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

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

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received K...	Sent KB/sec	Avg. Bytes
HTTP Reque...	500	43183	560	75956	16959.29	71.00%	6.6/sec	39.15	0.48	6105.6
TOTAL	500	43183	560	75956	16959.29	71.00%	6.6/sec	39.15	0.48	6105.6

☐ Include group name in label? ☒ Save Table Header

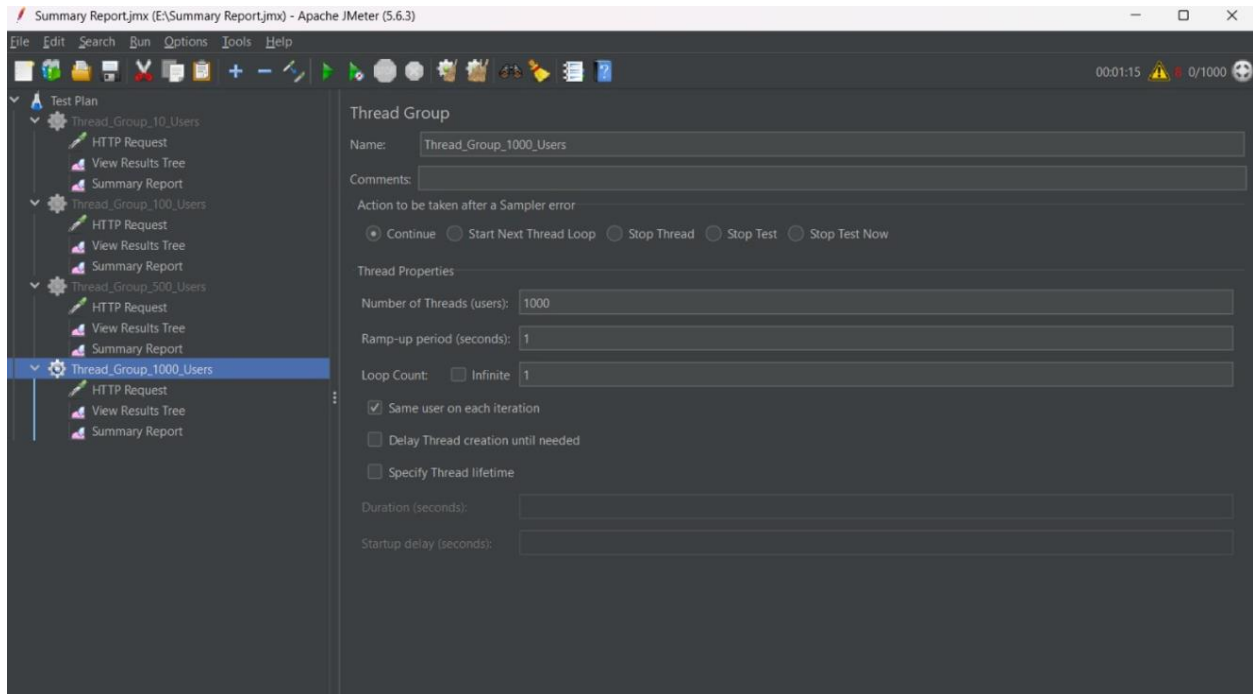
Analysis

The summary report shows that with 500 concurrent users, the home page has a throughput of 6.6 seconds and an error rate of 71%, indicating it cannot handle this level of traffic. With 100 concurrent users, the throughput is 52.3 seconds with a 0% error rate, demonstrating the system can manage this load comfortably. For 10 concurrent users, the throughput is 3.8 seconds with an error rate of 0%.

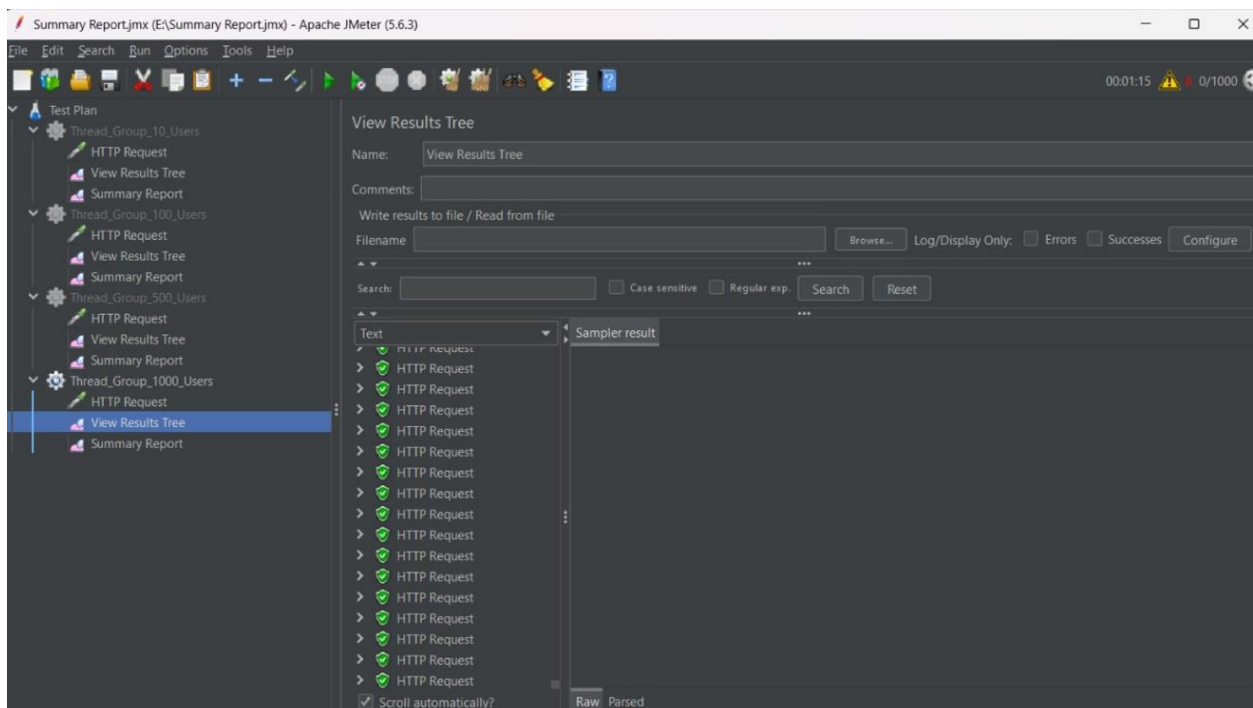
- Testing Home module with stress testing (1000 users)

GET request to PATH 'expensify-final.vercel.app/'

Scenario: We have accessed the home page of Expensify with the same user from multiple hosts (1000) and will be measuring the load balance and throughput of the system



Summary outputs for 1000 concurrent users



Summary Report.jmx (E:\Summary Report.jmx) - Apache JMeter (5.6.3)

File Edit Search Run Options Tools Help

00:01:15 0/1000

Test Plan

- Thread_Group_10_Users
 - HTTP Request
 - View Results Tree
 - Summary Report
- Thread_Group_100_Users
 - HTTP Request
 - View Results Tree
 - Summary Report
- Thread_Group_500_Users
 - HTTP Request
 - View Results Tree
 - Summary Report
- Thread_Group_1000_Users
 - HTTP Request
 - View Results Tree
 - Summary Report

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

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

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received K...	Sent KB/sec	Avg. Bytes
HTTP Reque...	1000	43224	36897	75285	5600.49	94.00%	13.3/sec	67.25	0.36	5192.0
TOTAL	1000	43224	36897	75285	5600.49	94.00%	13.3/sec	67.25	0.36	5192.0

☐ Include group name in label? ☒ Save Table Header

Analysis

The summary report shows that with 1000 concurrent users, the home page has a throughput of 13.3 seconds and an error rate of 94%, indicating it cannot handle this level of traffic

4. Reliability Check

The monitoring dashboard for the Neon database used in Expensify provides a comprehensive view of the system's reliability. Metrics like RAM and CPU usage indicate efficient resource allocation, ensuring that the application remains stable under varying workloads. The connections count graph tracks active and idle connections, confirming that the database handles traffic effectively without bottlenecks. The absence of deadlocks highlights that the database processes queries without conflicts, ensuring consistent and uninterrupted operations. The database size monitoring showcases proper storage management, preventing unexpected storage issues. Additional metrics, such as rows inserted/updated/deleted and local file cache hit rate, demonstrate data processing efficiency and quick retrieval speeds, further contributing to reliability.

