

NFR Testing for EktaFund Donation Website

IT-314 Software Engineering
Group 19

Group Mentor: Diti Soni

Group Members:

202201315	Dani Jal Nileshbhai
202201281	Shah Sakshi Dharmeshkumar
202201354	Ram Kulkarni
202201292	Devamm Patel
202201275	Kunj Mahendra Bhuva
202201271	Rishabh Jain
202201362	Tanay Kewalramani
202201303	Pandya Ansh Ashutoshbhai
202201264	Yadav Alpeshkumar Banshbahadur
202201363	Mangukiya Harshkumar Ashwinbhai

Contents

1	Introduction	1
1.1	Purpose of the Document	1
1.2	Overview of Ekta Group Donation System	1
2	Non-Functional Requirements (NFR)	2
2.1	Performance	2
2.2	Reliability	2
2.3	Scalability	2
2.4	Usability	2
2.5	Data Privacy	2
2.6	Mobile Accessibility	2
3	Performance Testing using Jmeter	3
3.0.1	Overview of JMeter	3
3.1	Sprint 1: User Registration and Authentication	4
3.2	Sprint 1 NFR Testing Results	5
3.2.1	Test Link	5
3.2.2	Testing Images	5
3.2.3	Performance Graph	10
3.3	Sprint 2: Admin Verification and NGO Listing	13
3.3.1	Test Link	14
3.3.2	Testing Images	14
3.3.3	Performance Graph	17
3.4	Sprint 3: Donation Management	19
3.4.1	Test Link	20
3.4.2	Testing Images	20
3.4.3	Performance Graph	24
3.5	Sprint 4: User Feedback and Support	26
3.5.1	Test Link	27
3.5.2	Testing Images	27
3.5.3	Performance Graph	28
4	Usabilities and Accessibilities	29
4.1	Usability Testing	29
5	Security Testing using Pentest-tool	32
5.1	Overview of Pentest-Tools	32

Chapter 1

Introduction

1.1 Purpose of the Document

The purpose of this document on Non-Functional Requirements (NFR) testing is to evaluate the system's performance, security, reliability, and usability across development sprints. By organizing results with graphs, JMeter images, and CSV data, it provides clear evidence of the system's ability to meet these standards. This sprint-wise approach highlights incremental improvements and identifies areas for enhancement, supporting stakeholders in making informed decisions about the system's readiness and quality.

1.2 Overview of Ekta Group Donation System

The Ekta Group Donation Website aims to connect donors with individuals and organizations in need of donations. The platform allows donors to browse various donation categories, make secure donations, and track their donation history. Beneficiaries can list their needs, and admins can manage requests, ensuring donations are allocated effectively.

Chapter 2

Non-Functional Requirements (NFR)

2.1 Performance

System responses should be efficient, especially during high-traffic events.

2.2 Reliability

Ensure consistent uptime and data reliability.

2.3 Scalability

Accommodate an increasing number of users and donation requests.

2.4 Usability

Design a user-friendly interface that is easy to navigate for all ages.

2.5 Data Privacy

Protect donor and beneficiary information in compliance with data privacy regulations.

2.6 Mobile Accessibility

Make the platform mobile-friendly for ease of access.

Chapter 3

Performance Testing using Jmeter

3.0.1 Overview of JMeter

Apache JMeter is an open-source, Java-based performance testing tool designed to measure the performance and load-handling capacity of web applications and services. It simulates multiple users accessing an application simultaneously to identify bottlenecks and ensure reliability under varying workload

Key Features of JMeter:

- Simulates real-world usage scenarios by generating concurrent user requests.
- Provides extensive support for protocol testing, including HTTP, HTTPS, FTP, and JDBC database connections.
- Generates detailed performance metrics such as response times, throughput, and error rates.
- Includes graphical representation of test results via graphs, tables, and logs.
- Scalable to distributed environments for larger test scenarios.

JMeter is widely used in software development and quality assurance processes to ensure that applications meet non-functional requirements, such as performance, scalability, and reliability. It allows for testing of individual components as well as end-to-end systems under load conditions, making it a crucial tool for identifying and resolving performance issues.

3.1 Sprint 1: User Registration and Authentication

Duration: 2 weeks

Stories:

- **Donor Registration**

- Backend Story: User Registration API
- Tasks:
 - * Implement API for user registration.
 - * Validate input fields.
 - * Hash and securely store passwords.
 - * Include optional fields like contact numbers.
 - * Set up user roles and permissions.

- **NGO Registration**

- Backend Story: NGO Registration API
- Tasks:
 - * Implement API for NGO registration.
 - * Store organization details, mission statements, and legal documents.
 - * Create a profile for each NGO.

Estimated Effort: 42.9 ~ 43 FP

Functionality and Effort Calculation

Functionality	UFP	AFP	Total FP
API for user registration	$(1 \times 4) + (1 \times 5) + (1 \times 4) + (1 \times 7) + (0 \times 5) = 20$	0.65	13
Validate input fields	$(1 \times 4) + (0 \times 5) + (0 \times 4) + (0 \times 7) + (0 \times 5) = 4$	0.65	2.6
Hash and securely store passwords	$(1 \times 4) + (0 \times 5) + (0 \times 4) + (1 \times 7) + (0 \times 5) = 11$	0.65	7.15
Include optional fields	$(1 \times 4) + (0 \times 5) + (0 \times 4) + (1 \times 7) + (0 \times 5) = 11$	0.65	7.15
Set up roles & permissions	$(1 \times 4) + (1 \times 5) + (1 \times 4) + (1 \times 7) + (0 \times 5) = 20$	0.65	13

3.2 Sprint 1 NFR Testing Results

3.2.1 Test Link

For reference, the test was conducted using the following links: <https://demo-backend-y7ap.onrender.com//api/donors/register>, <https://demo-backend-y7ap.onrender.com//api/ngos/register>, <https://demo-backend-y7ap.onrender.com//api/forget-password/reset>

3.2.2 Testing Images

Here are the images showing the setup and results:

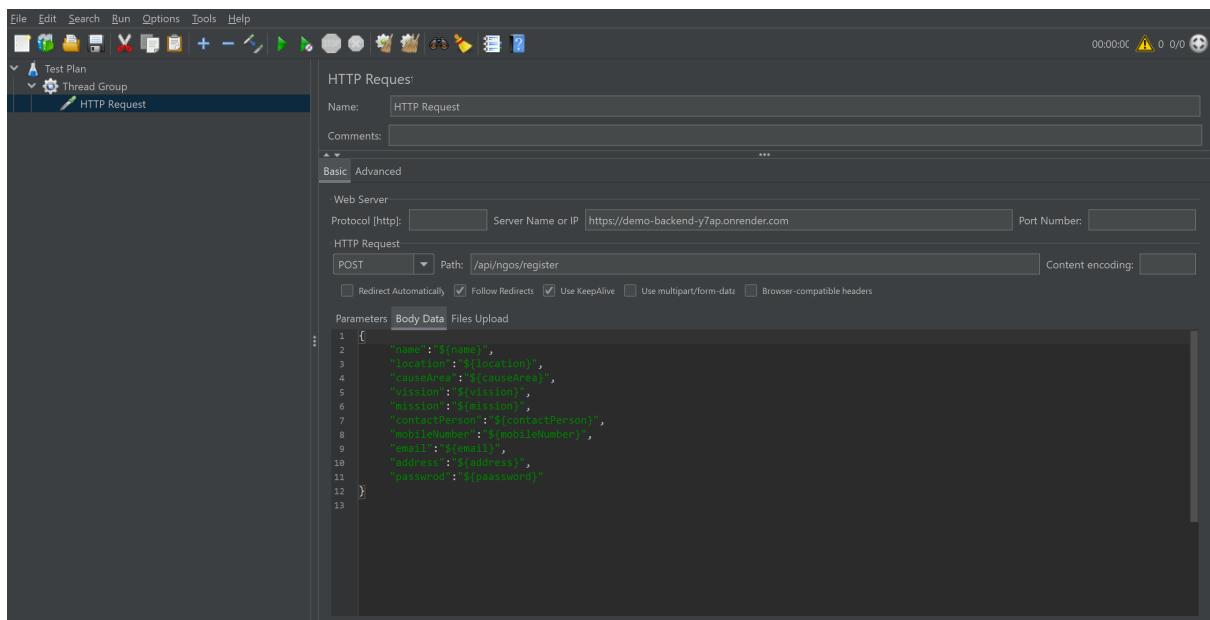


Figure 3.1: Image showing the initial setup of Sprint 1 for NGO register testing

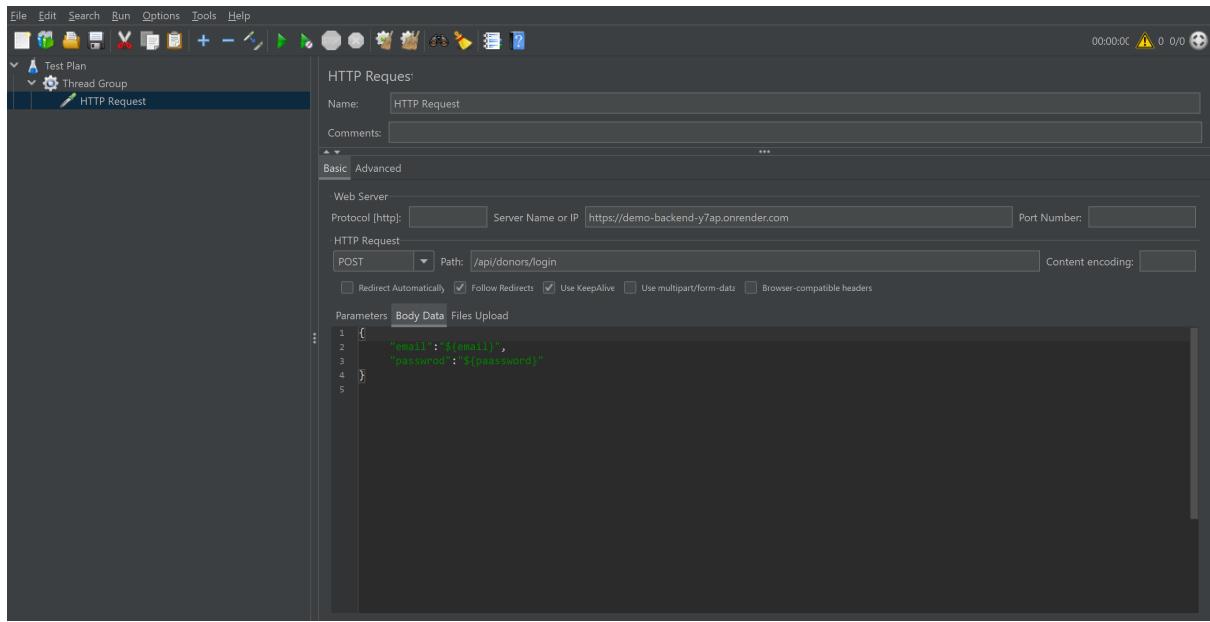


Figure 3.2: Image displaying Sprint 1 testing Donor Login

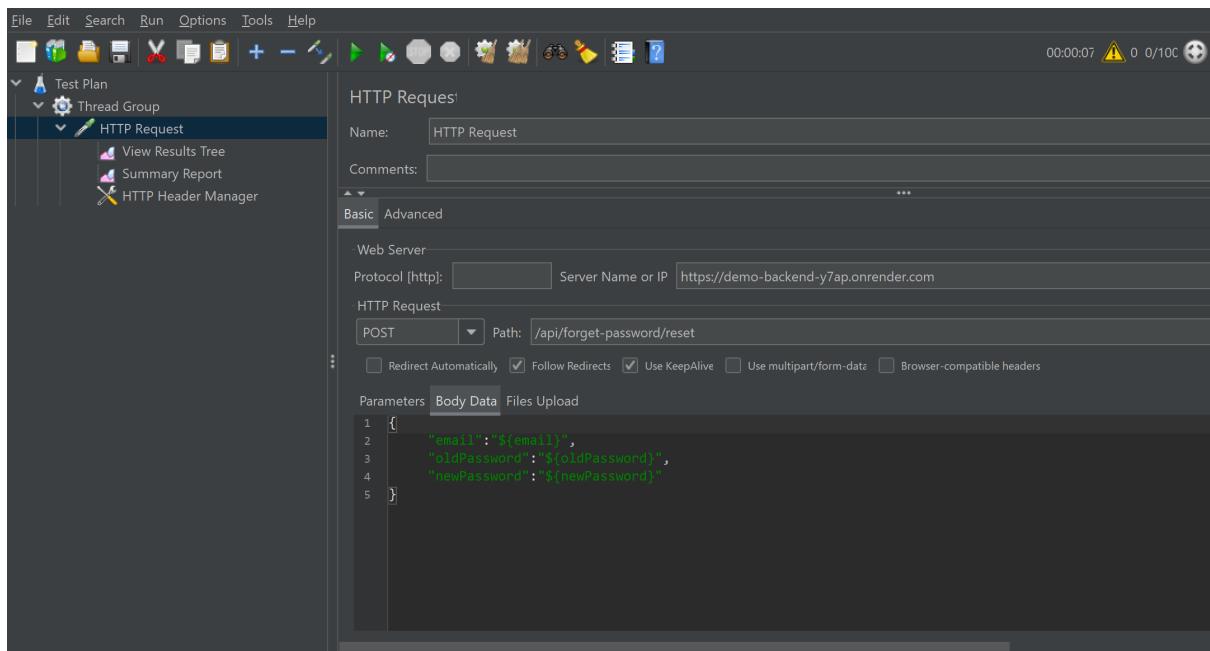


Figure 3.3: Image showing the initial setup of Sprint 1 testing for donor register

View Results in Table

Name:	<input type="text" value="View Results in Table"/>	Comments:	<input type="text" value="Write results to file / Read from file"/>	Filename:	<input type="text"/>	<input type="button" value="Browse..."/>	Log/Display Only:	<input type="checkbox"/> Errors	<input type="checkbox"/> Successes	<input type="button" value="Configure"/>
Sample #	Start Time	Thread Name	Label	Sample Time(...)	Status	Bytes	Sent Bytes	Latency	Connect Time(...)	
21	01:48:53.611	Thread Group ...	HTTP Request	93	✖	376	278	92	24	
22	01:48:53.725	Thread Group ...	HTTP Request	203	✖	389	293	203	2	
23	01:48:53.797	Thread Group ...	HTTP Request	459	✓	390	296	459	5	
24	01:48:53.916	Thread Group ...	HTTP Request	355	✓	390	300	355	2	
25	01:48:54.000	Thread Group ...	HTTP Request	475	✓	390	297	474	2	
26	01:48:54.115	Thread Group ...	HTTP Request	422	✓	390	299	422	1	
27	01:48:54.320	Thread Group ...	HTTP Request	532	✓	390	297	532	1	
28	01:48:54.223	Thread Group ...	HTTP Request	646	✓	390	298	646	2	
29	01:48:54.427	Thread Group ...	HTTP Request	466	✓	390	298	466	2	
30	01:48:54.521	Thread Group ...	HTTP Request	440	✓	390	300	440	1	
31	01:53:16.906	Thread Group ...	HTTP Request	62	✖	376	278	62	5	
32	01:53:17.025	Thread Group ...	HTTP Request	153	✖	389	293	153	2	
33	01:53:17.120	Thread Group ...	HTTP Request	446	✓	390	296	446	2	
34	01:53:17.205	Thread Group ...	HTTP Request	401	✓	390	300	401	2	
35	01:53:17.301	Thread Group ...	HTTP Request	613	✓	390	297	613	2	
36	01:53:17.417	Thread Group ...	HTTP Request	566	✓	390	299	566	2	
37	01:53:17.511	Thread Group ...	HTTP Request	683	✓	390	298	683	2	
38	01:53:17.618	Thread Group ...	HTTP Request	587	✓	390	297	587	2	
39	01:53:17.714	Thread Group ...	HTTP Request	499	✓	390	298	499	2	
40	01:53:17.814	Thread Group ...	HTTP Request	491	✓	390	300	491	1	

Scroll automatically? Child samples? No of Samples: 40 Latest Sample: 491 Average: 216 Deviation: 245

Figure 3.4: Image showing the initial setup of Sprint 1 testing outcomes for ngo register

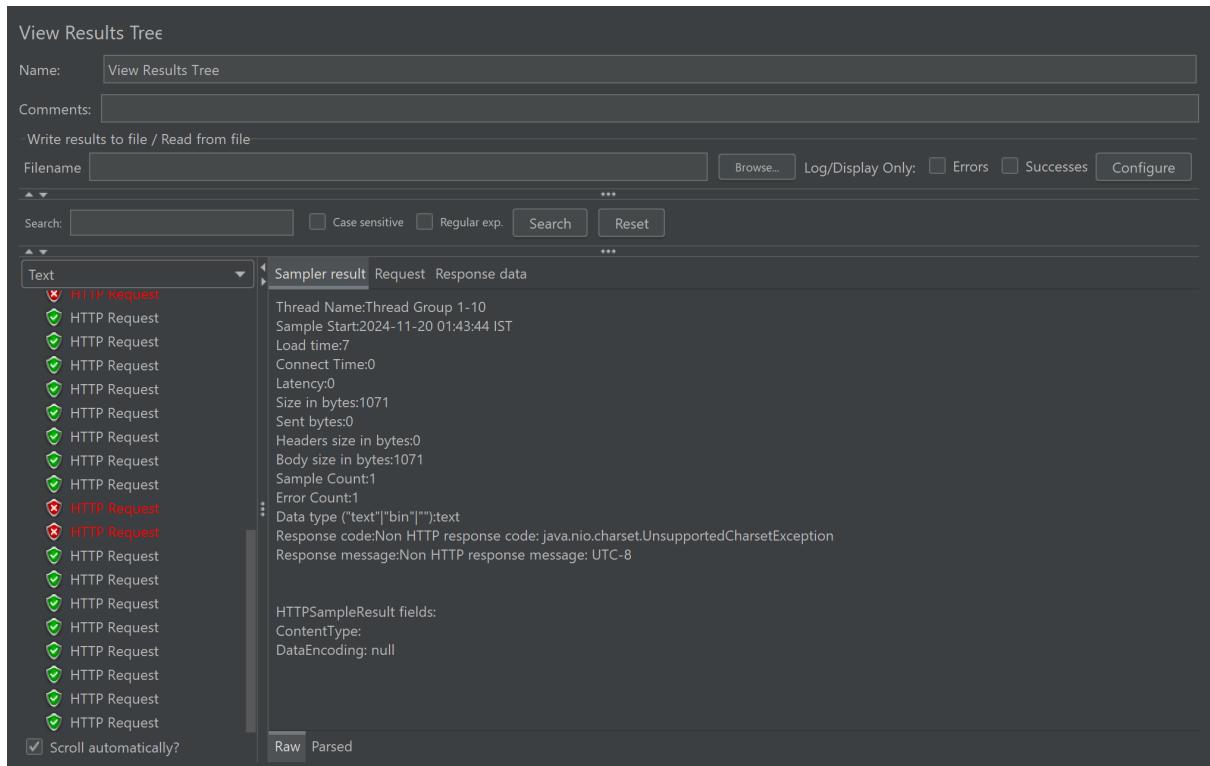


Figure 3.5: Image showing the initial setup of Sprint 1 testing for donor register

3.2.3 Performance Graph

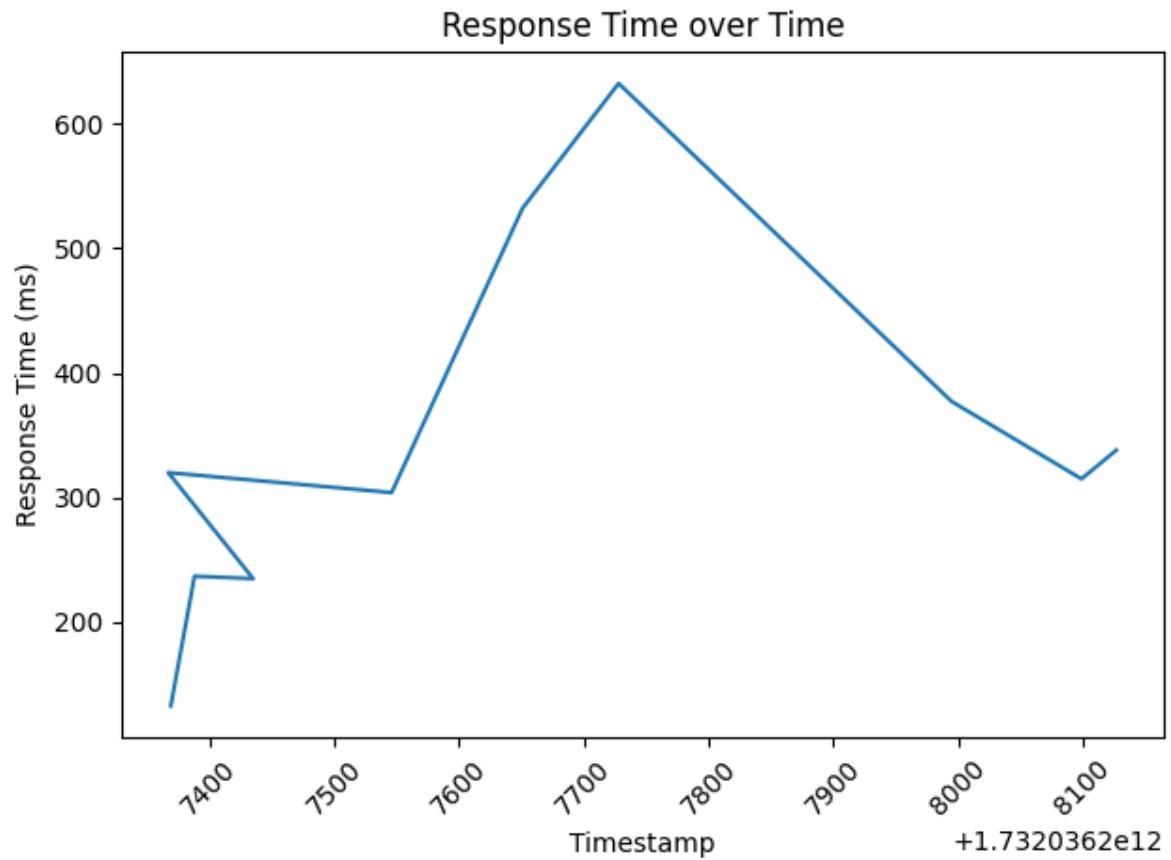


Figure 3.6: Graph displaying performance metrics for Sprint 1 for creating 10 users simultaneously

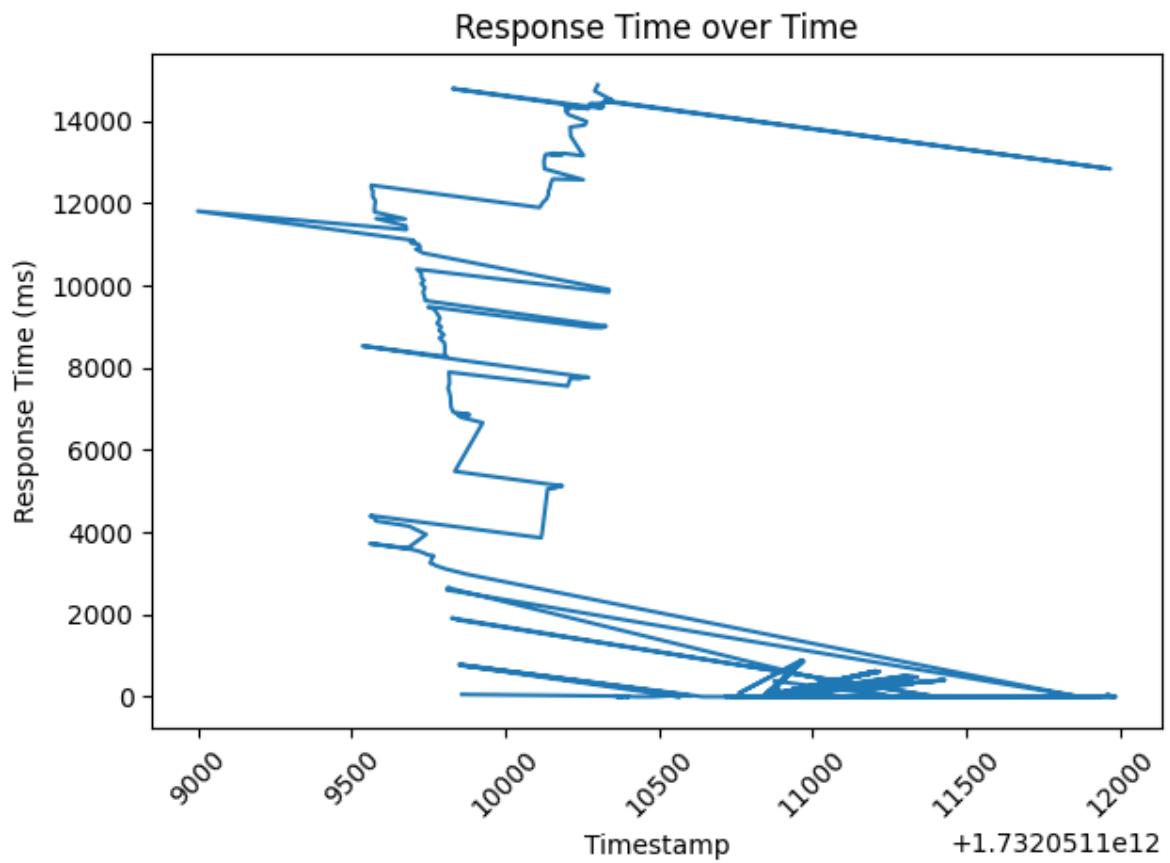


Figure 3.7: Graph displaying performance metrics for Sprint 1 for creating 1000 users simultaneously

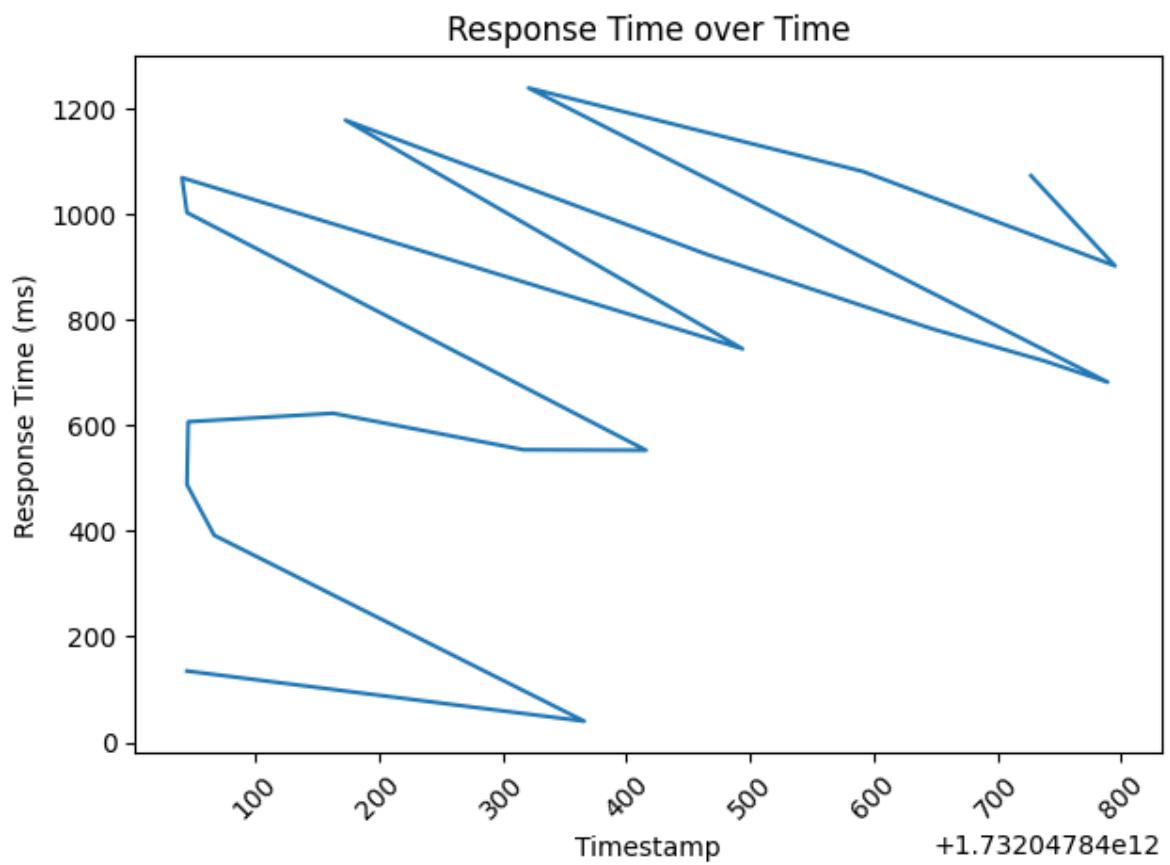


Figure 3.8: Graph displaying performance metrics for Sprint 1 for Forget-password

3.3 Sprint 2: Admin Verification and NGO Listing

Duration: 2.5 weeks

Stories:

- **Admin Verification**

- Backend Story: Verification API
- Tasks:
 - * Implement verification process for NGO documents.
 - * Include status updates and notifications.

- **Search and Filter NGOs**

- Backend Story: NGO Listing and Filtering API
- Tasks:
 - * Create API to retrieve NGO profiles with filtering options.
 - * Optimize database queries for search and filter operations.

Estimated Effort: 49 FP

Functionality and Effort Calculation

Functionality	UFP	AFP	Total FP
Verification NGO	$(1 \times 4) + (1 \times 5) + (1 \times 4) + (1 \times 7) + (0 \times 5) = 20$	0.65	13
Status updates & notifications	$(1 \times 4) + (1 \times 5) + (1 \times 4) + (1 \times 7) + (0 \times 5) = 20$	0.65	13
Retrieve NGO profile	$(1 \times 4) + (1 \times 5) + (1 \times 4) + (1 \times 7) + (0 \times 5) = 20$	0.65	13
Validity for changes & integrity	$(1 \times 4) + (0 \times 5) + (1 \times 4) + (1 \times 7) + (0 \times 5) = 15$	0.65	9.75
NGO to update projects	$(1 \times 4) + (1 \times 5) + (0 \times 4) + (1 \times 7) + (0 \times 5) = 16$	0.65	10.4

3.3.1 Test Link

For reference, the test was conducted using the following links: <https://demo-backend-y7ap.onrender.com//api/admin/verify-ngo>, <https://demo-backend-y7ap.onrender.com//api/admin/utils/notification>, <https://demo-backend-y7ap.onrender.com//api/donor/location>

3.3.2 Testing Images

Here are the images showing the setup and results:

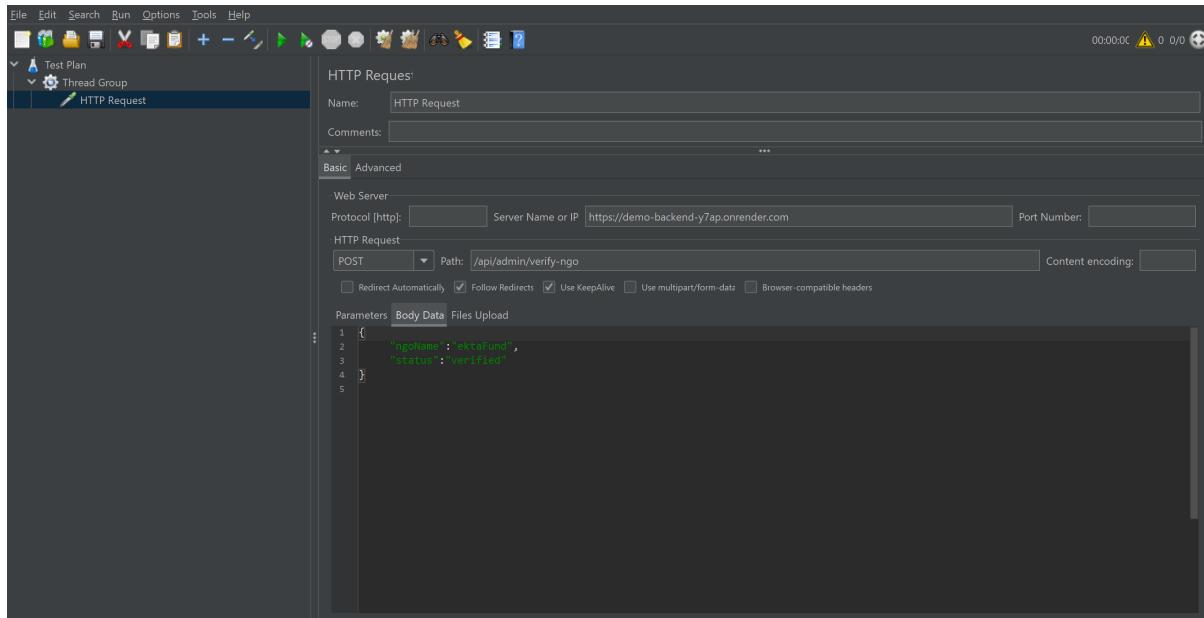


Figure 3.9: Image showing the initial setup of Sprint 2 testing for NGO verification

View Results in Table

Name:	View Results in Table								
Comments:									
<input type="checkbox"/> Write results to file / Read from file <input type="text"/> Filename <input type="button" value="Browse..."/> Log/Display Only: <input type="checkbox"/> Errors <input type="checkbox"/> Successes <input type="button" value="Configure"/>									
Sample #	Start Time	Thread Name	Label	Sample Time(...)	Status	Bytes	Sent Bytes	Latency	Connect Time(...)
1	01:13:02.017	Thread Group ...	HTTP Request	807	✓	389	306	797	64
2	01:15:09.416	Thread Group ...	HTTP Request	94	✗	634	240	94	12
3	01:16:12.254	Thread Group ...	HTTP Request	81	✗	634	240	81	7
4	01:19:39.218	Thread Group ...	HTTP Request	165	✗	496	240	165	8
5	01:22:35.013	Thread Group ...	HTTP Request	24901	✓	865	240	24900	19

Scroll automatically? Child samples?
 No of Samples 5 Latest Sample 24901 Average 5209 Deviation 9849

Figure 3.10: Image displaying Sprint 2 testing outcomes

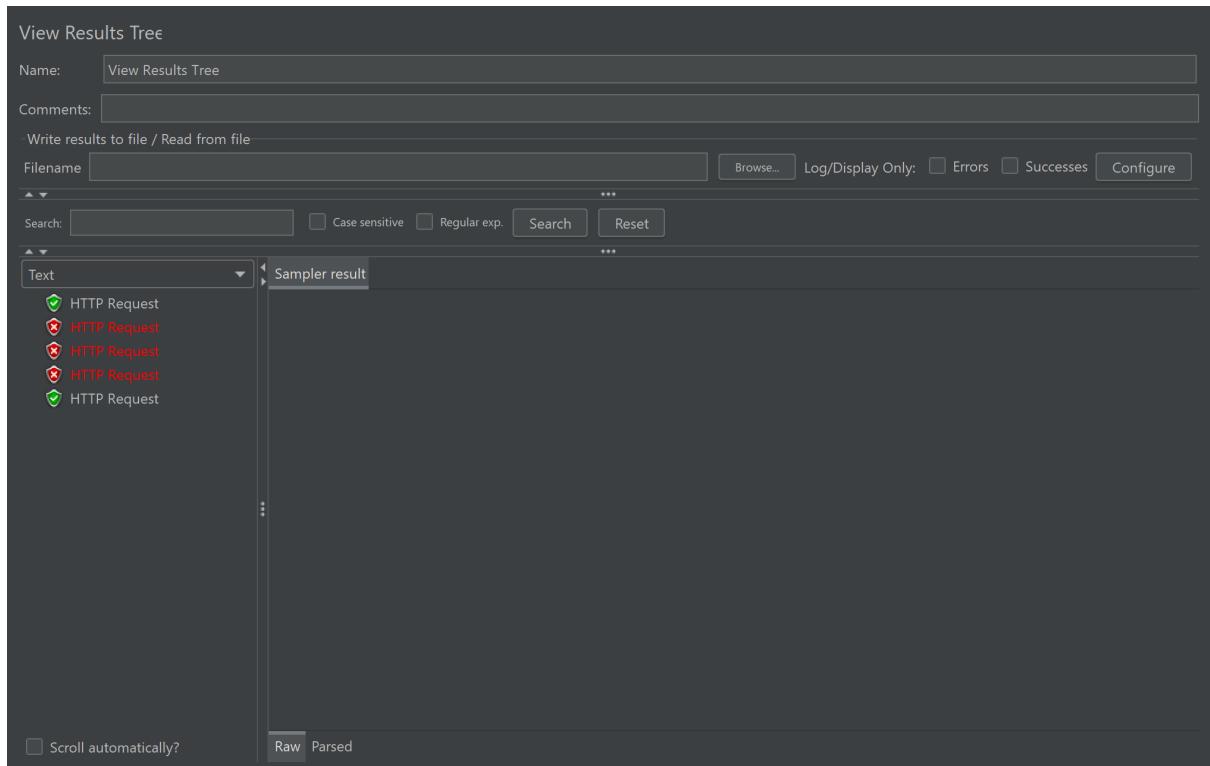


Figure 3.11: Image displaying Sprint 2 testing outcomes

3.3.3 Performance Graph

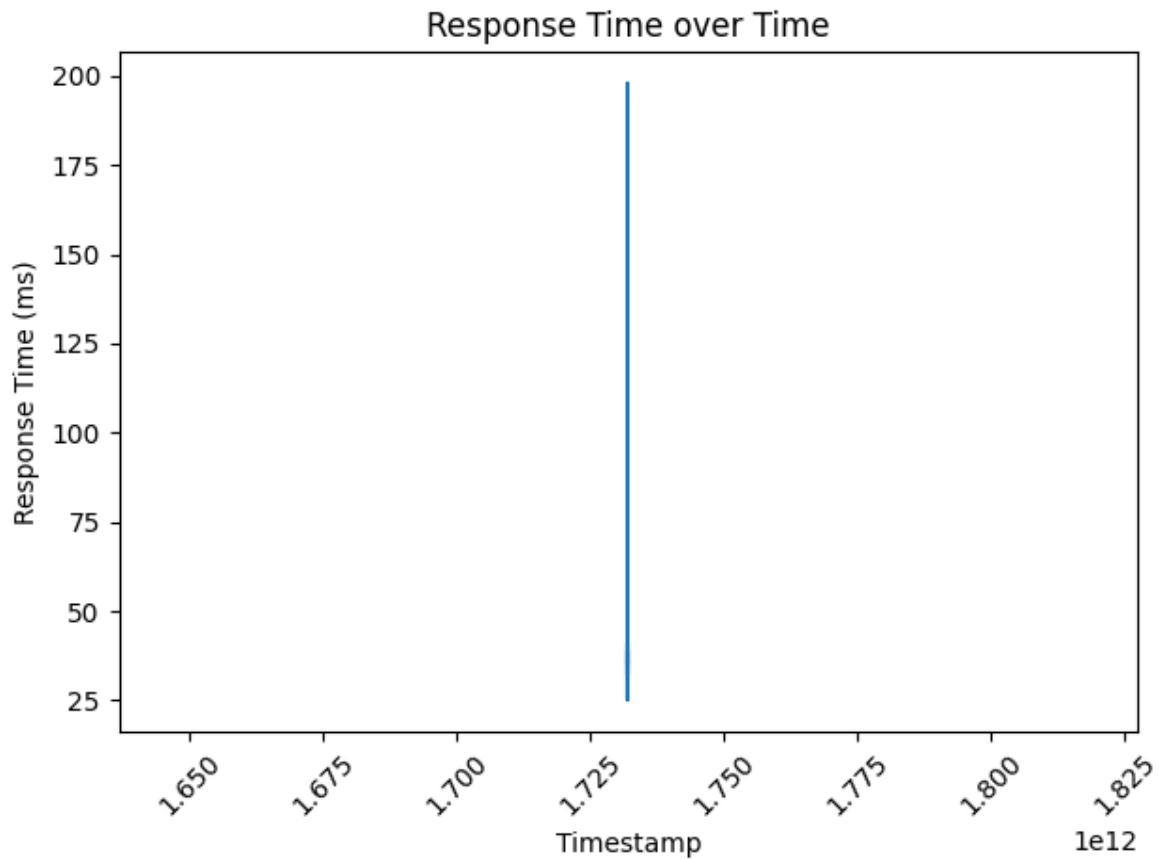


Figure 3.12: Graph displaying performance metrics for Sprint 2 for searching NGO by it's location

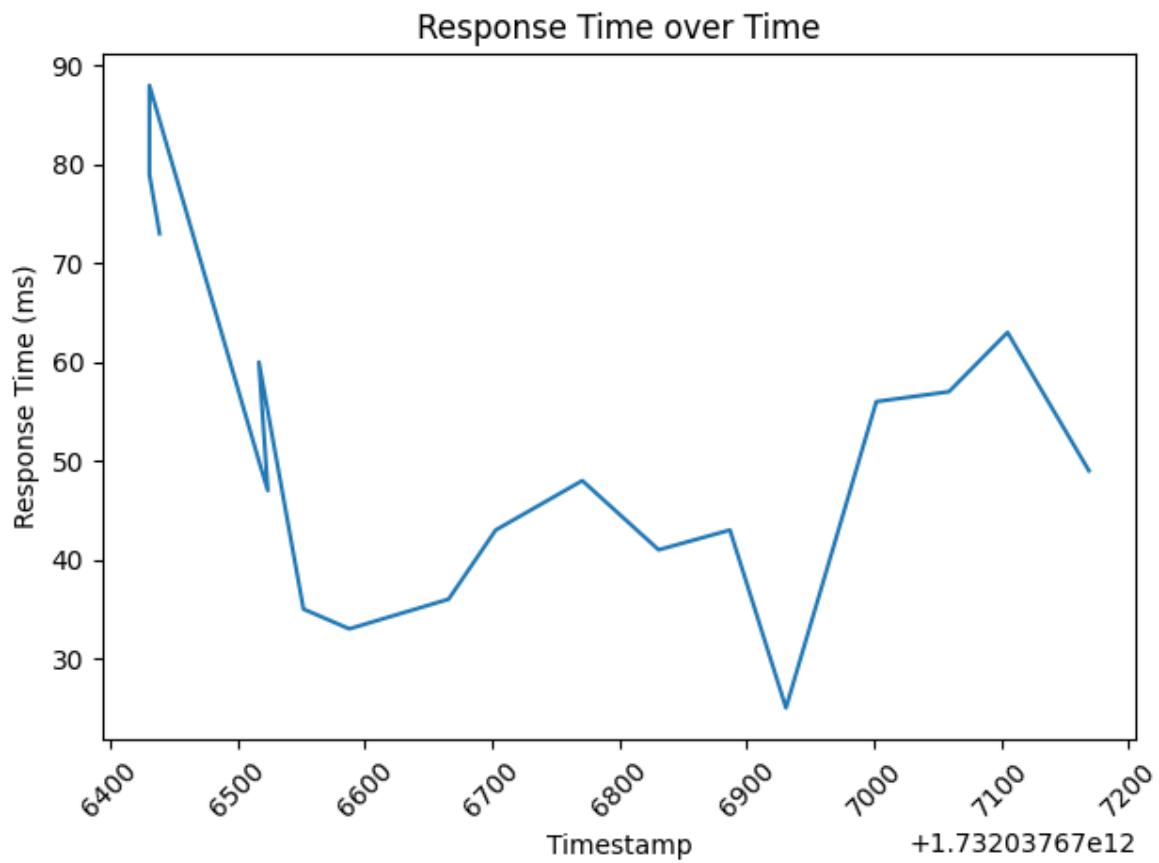


Figure 3.13: Graph displaying performance metrics for Sprint 2 for Admin Pending-verification

3.4 Sprint 3: Donation Management

Duration: 3 weeks

Stories:

- **Donation Process**

- Backend Story: Payment Processing API
- Tasks:
 - * Integrate payment gateways.
 - * Ensure secure processing and storage of payment information.
 - * Generate donation receipts and transaction IDs.

- **NGO Donation Management**

- Backend Story: Bill Generation API
- Tasks:
 - * Create a billing system for receipts.
 - * Store transaction records securely.
 - * Provide downloadable donation reports.

Estimated Effort: 61 FP

Functionality and Effort Calculation

Functionality	UFP	AFP	Total FP
Integrated payment gateways	$(1 \times 4) + (1 \times 5) + (1 \times 4) + (0 \times 7) + (1 \times 5) = 18$	0.65	11.7
Generate donation receipts	$(1 \times 4) + (1 \times 5) + (1 \times 4) + (1 \times 7) + (0 \times 5) = 17$	0.65	11.05
Create a billing system for receipts	$(2 \times 4) + (2 \times 5) + (1 \times 3) + (1 \times 7) + (0 \times 5) = 23$	0.65	15.6
Store transaction records securely	$(0 \times 3) + (1 \times 4) + (0 \times 3) + (1 \times 7) + (0 \times 5) = 11$	0.65	7.15
Provide donation reports	$(2 \times 4) + (1 \times 5) + (1 \times 4) + (1 \times 7) + (0 \times 5) = 23$	0.65	15.6

3.4.1 Test Link

For reference, the test was conducted using the following links: <https://demo-backend-y7ap.onrender.com//api/donations/donate>, <https://demo-backend-y7ap.onrender.com//api/donation/services/payment>, <https://demo-backend-y7ap.onrender.com//api/donor/services/receiptGenerator>

3.4.2 Testing Images

Here are the images showing the setup and results:

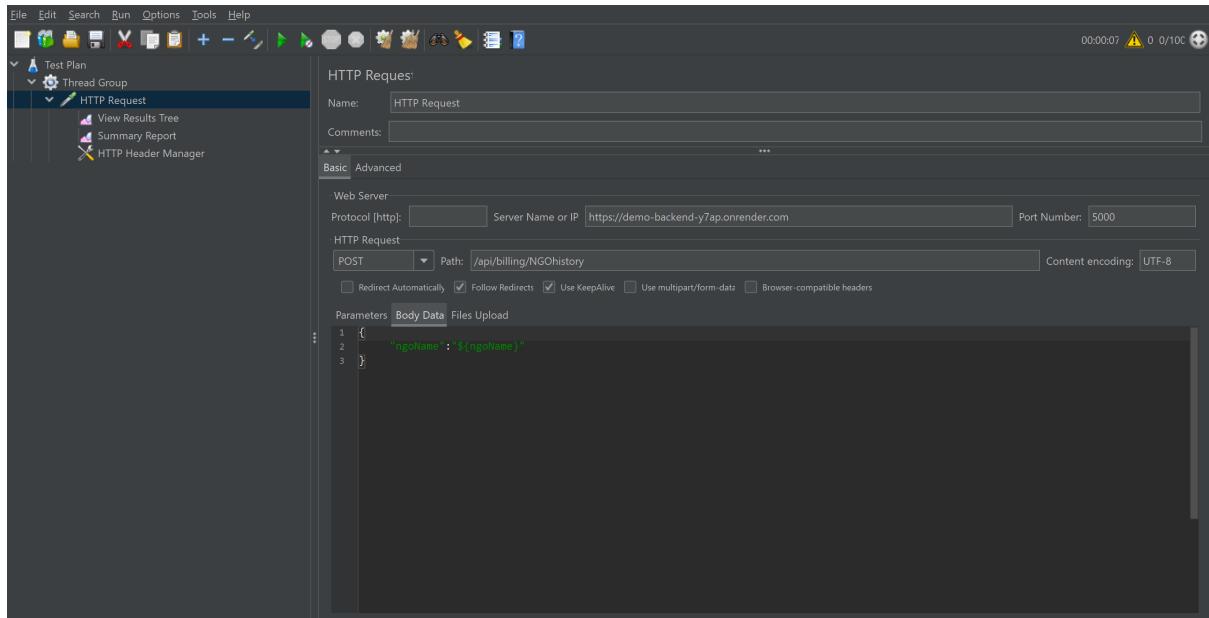


Figure 3.14: Image displaying Sprint 3 set up for the ngohistories

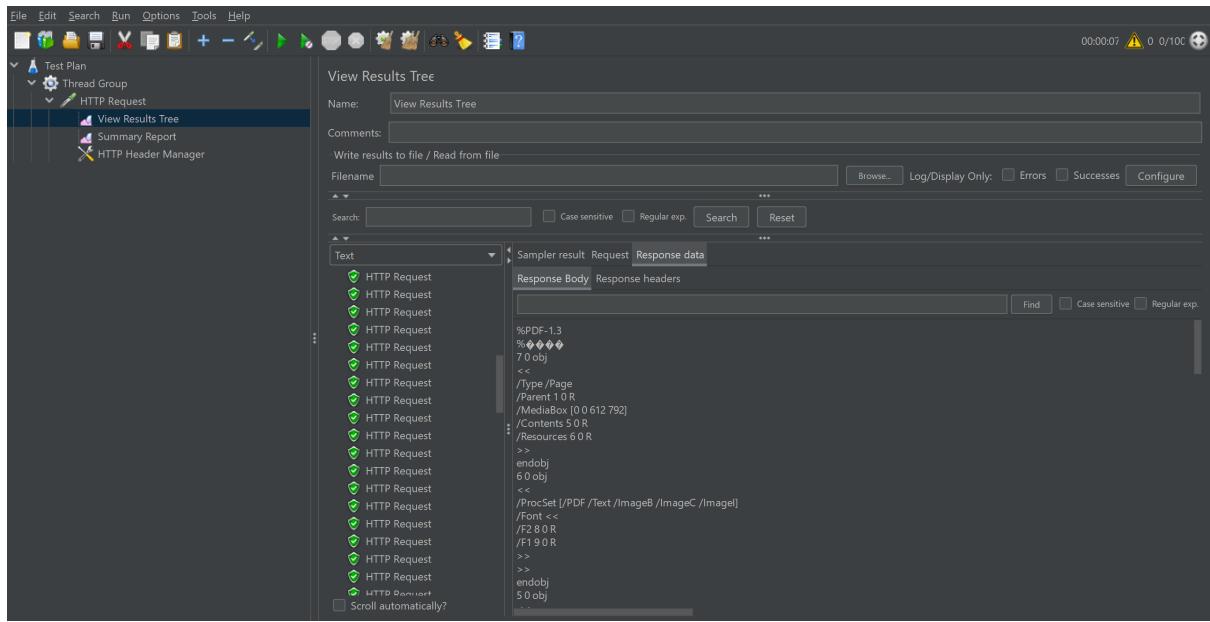


Figure 3.15: Image displaying Sprint 3 testing outcomes for ngohistroies

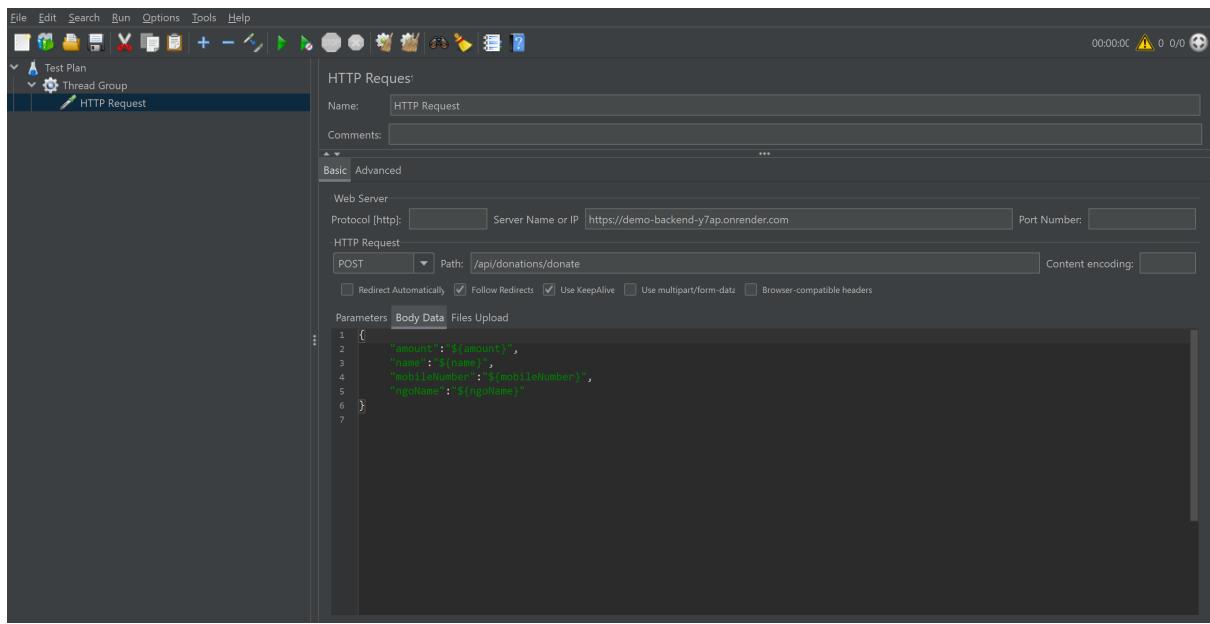


Figure 3.16: Image displaying Sprint 3 testing inputs of the payment

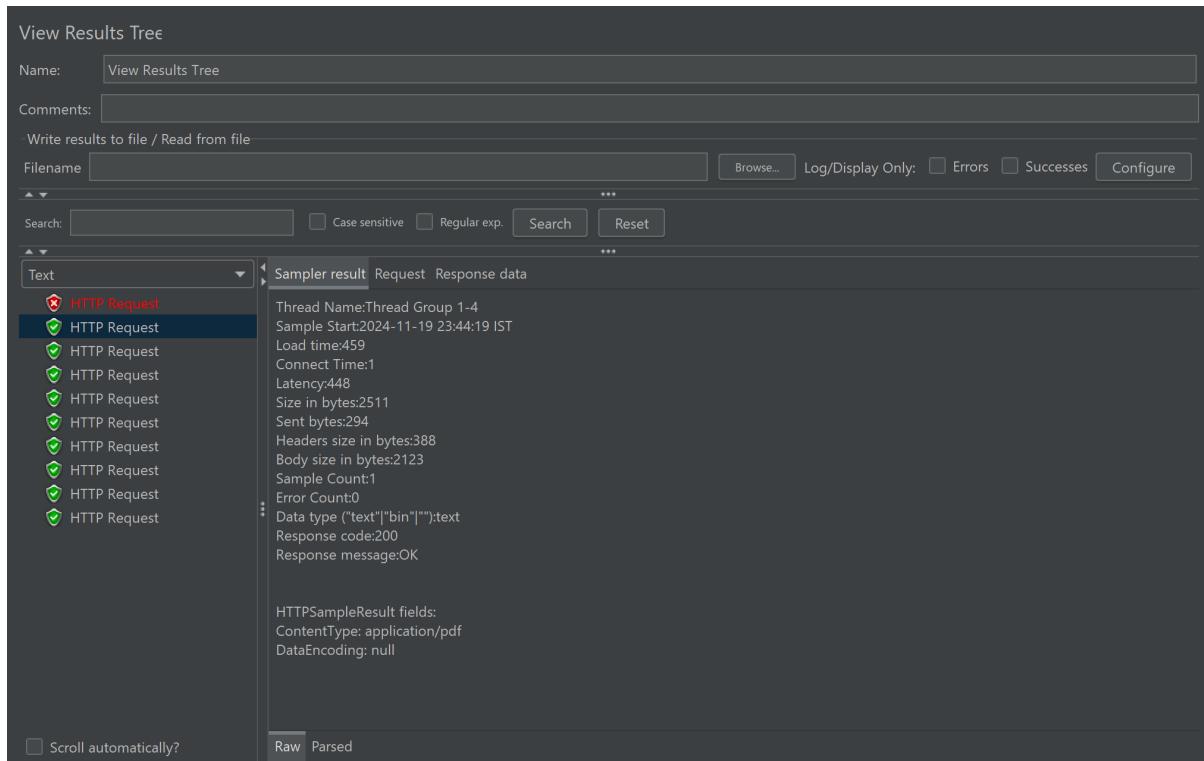


Figure 3.17: Image displaying Sprint 3 testing outcomes for payment

3.4.3 Performance Graph

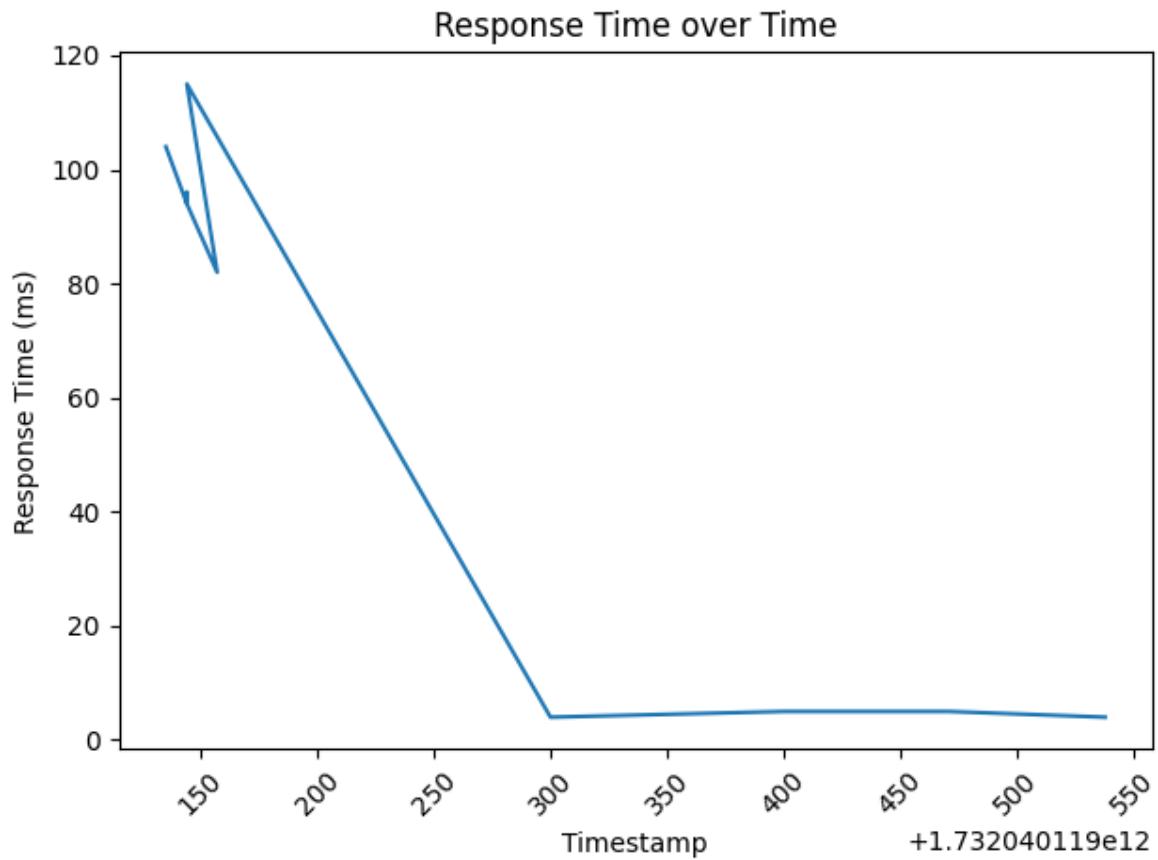


Figure 3.18: Graph displaying performance metrics for Sprint 3 for Payment to NGO

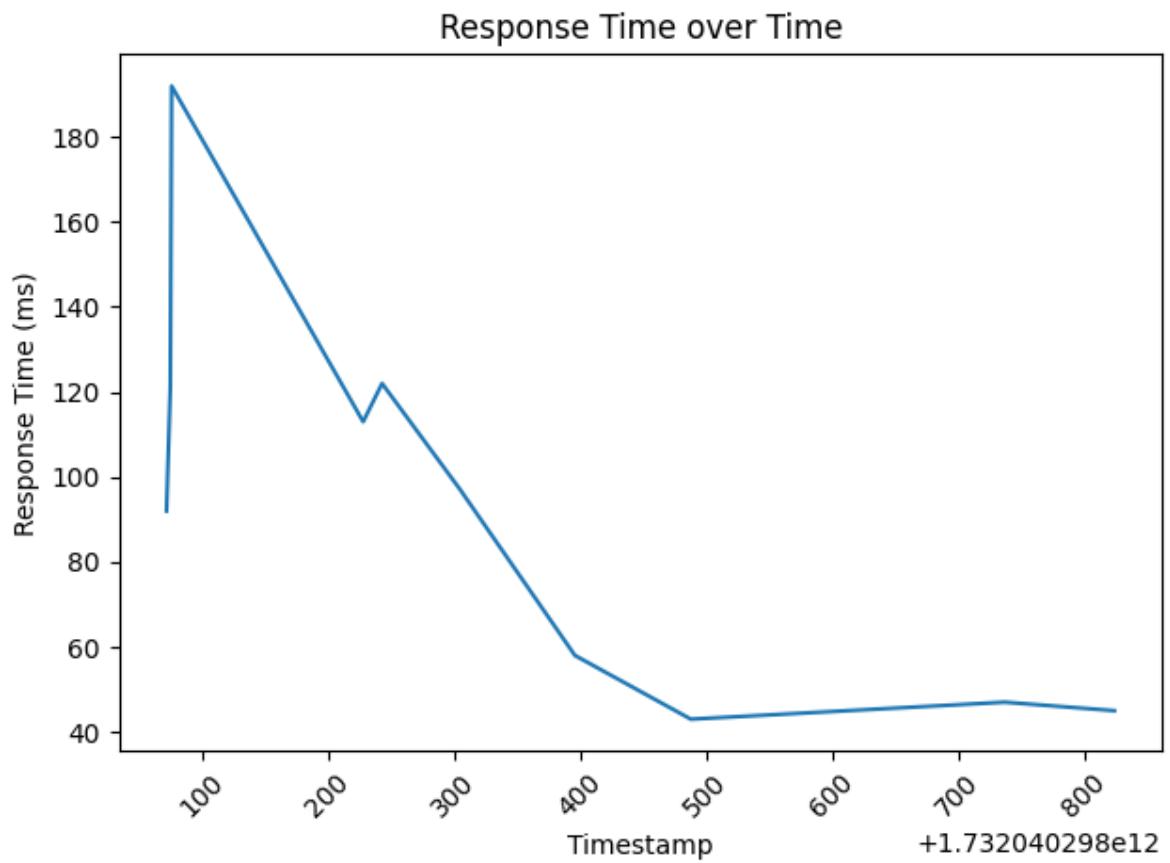


Figure 3.19: Graph displaying performance metrics for Sprint 3 for Payment NGO

3.5 Sprint 4: User Feedback and Support

Duration: 1.5 weeks

Stories:

- **Donor Feedback**

- Backend Story: Review System API
- Tasks:
 - * Implement API for creating and fetching reviews.
 - * Allow NGOs to reply to reviews.
 - * Ensure reviews are moderated by admins.

- **NGO Response**

- Backend Story: Review Response API
- Tasks:
 - * Enable NGOs to respond to feedback.

Estimated Effort: 33.15 FP

Functionality and Effort Calculation

Functionality	UFP	AFP	Total FP
Review system	$(1 \times 4) + (1 \times 5) + (1 \times 4) + (1 \times 7) + (0 \times 5) = 20$	0.65	13
Allow NGOs to reply	$(1 \times 4) + (0 \times 5) + (0 \times 4) + (1 \times 7) + (0 \times 5) = 11$	0.65	7.15
Reviews moderation by admins	$(1 \times 4) + (1 \times 5) + (1 \times 4) + (1 \times 7) + (0 \times 5) = 20$	0.65	13

3.5.1 Test Link

For reference, the test was conducted using the following links: <https://demo-backend-y7ap.onrender.com//api/donations/donate>, <https://demo-backend-y7ap.onrender.com//api/donation/services/payment>, <https://demo-backend-y7ap.onrender.com//api/donor/services/receiptGenerator>

3.5.2 Testing Images

Here are the images showing the setup and results:

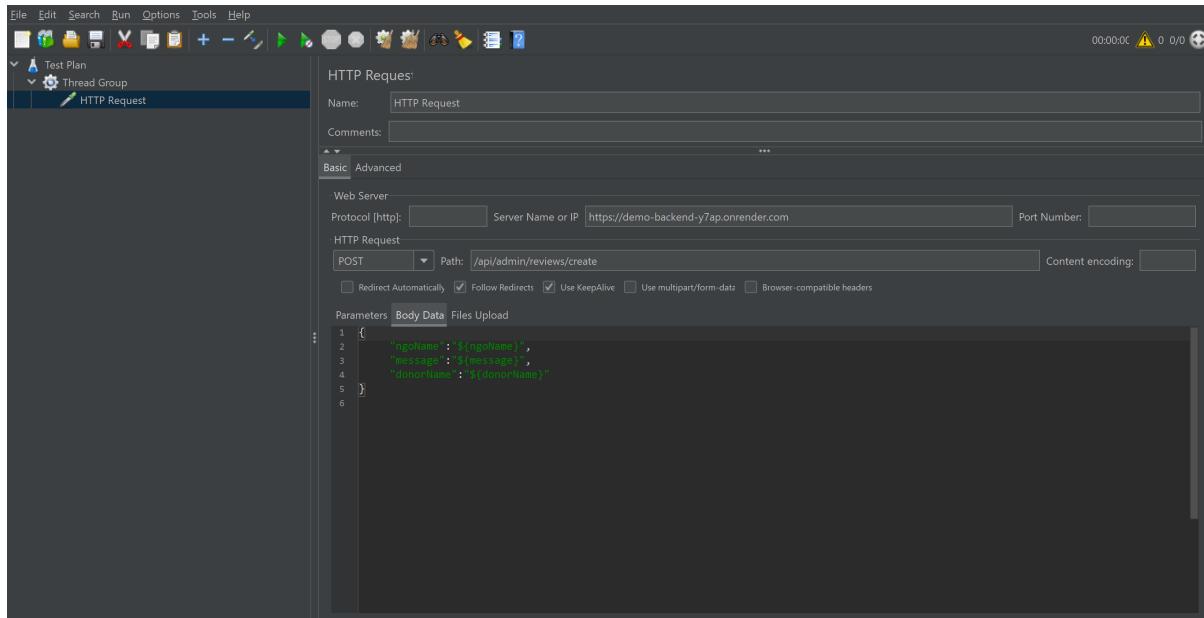


Figure 3.20: Image showing the initial setup of Sprint 4 testing for review showing

3.5.3 Performance Graph

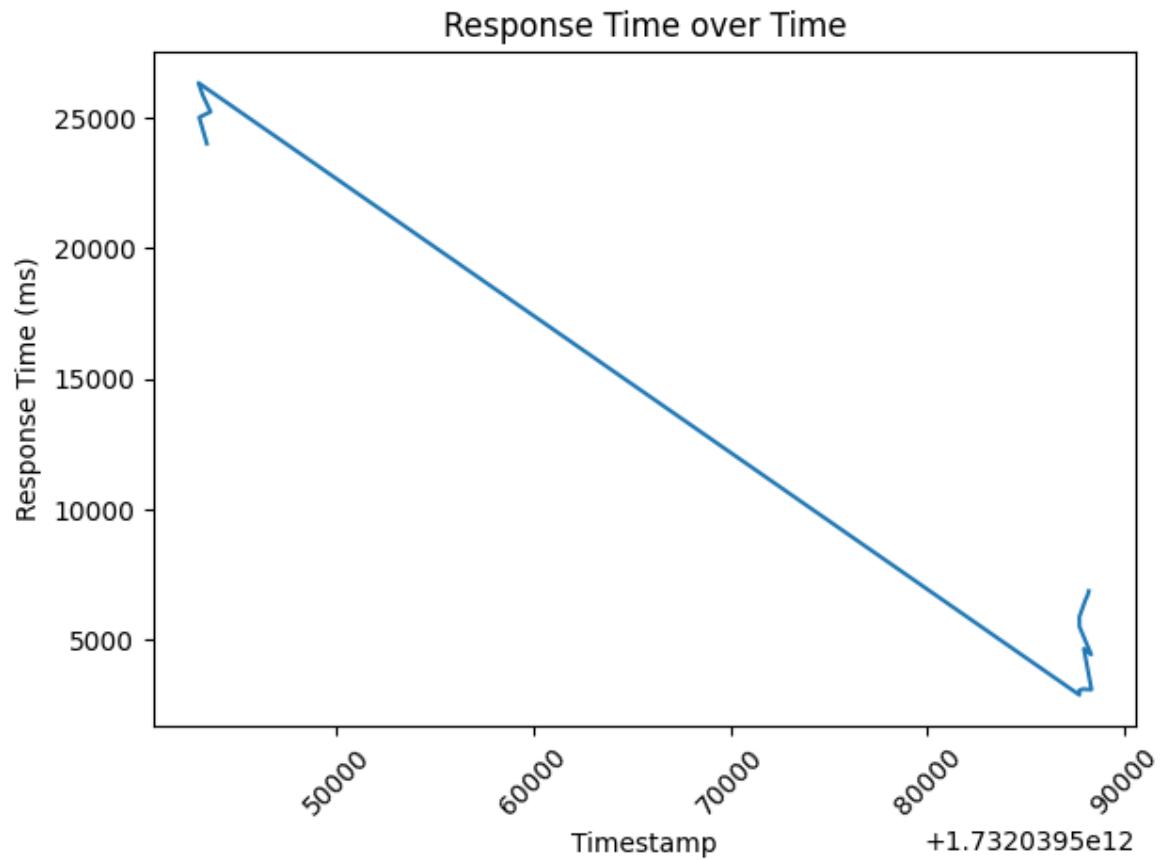


Figure 3.21: Graph displaying performance metrics for Sprint 4 Reviewing NGO for submission review by user

Chapter 4

Usabilities and Accessibilities

4.1 Usability Testing

Usability testing evaluates the user-friendliness and effectiveness of a system, ensuring that end-users can easily interact with the application. It involves assessing how intuitive, efficient, and satisfying the user experience (UX) is across different devices and browsers. In the context of Non-Functional Requirements (NFR) testing, usability plays a critical role in ensuring that a system meets the expectations of its users.

Key areas of usability testing include:

- **Ease of Use:** Measures how easy it is for users to navigate the website or application and perform common tasks, such as registering, making donations, or accessing key features.
- **Consistency:** Ensures that the design and interface elements are uniform across different pages and platforms, providing a seamless experience.
- **User Satisfaction:** Collects feedback on the overall experience, addressing whether users are satisfied with the system's design, responsiveness, and accessibility.
- **Accessibility:** Tests how well the system adheres to accessibility standards such as WCAG (Web Content Accessibility Guidelines), ensuring that it is usable by people with disabilities (e.g., screen reader compatibility, keyboard navigation).
- **Performance under Load:** Assesses the system's performance when handling a large number of users or transactions, ensuring that it remains responsive and functional during peak usage times.

By incorporating usability testing into NFR evaluations, organizations can identify pain points, improve user experience, and ensure that the platform is accessible to a wider audience, including those with special needs.

Results:

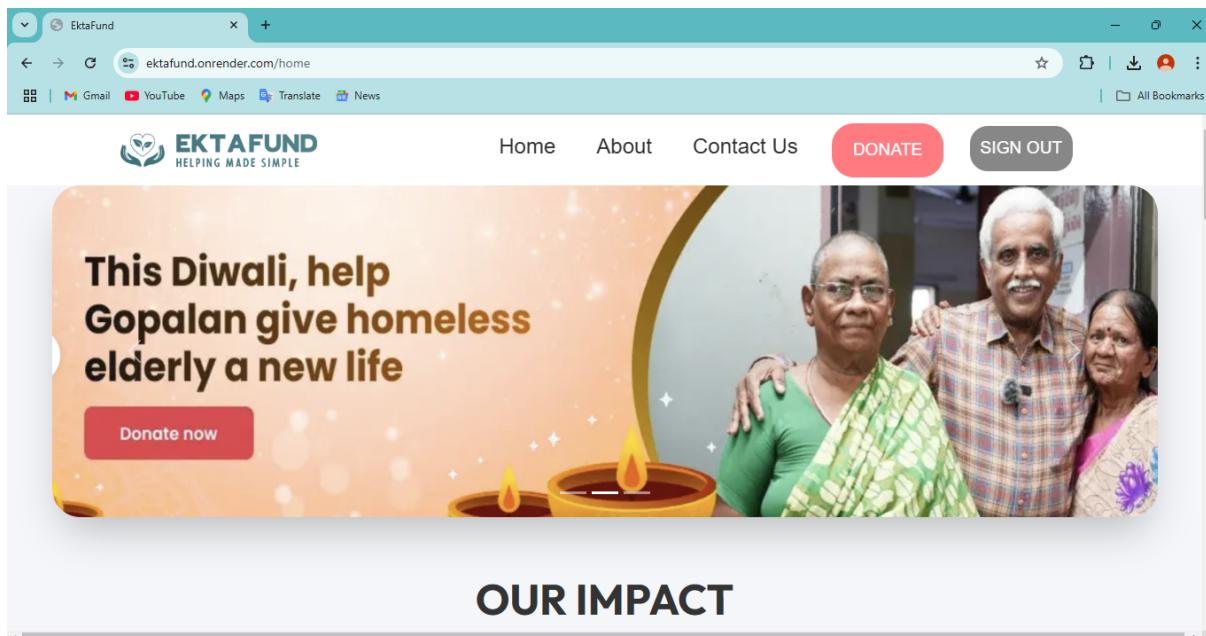


Figure 4.1: Google browser usablities

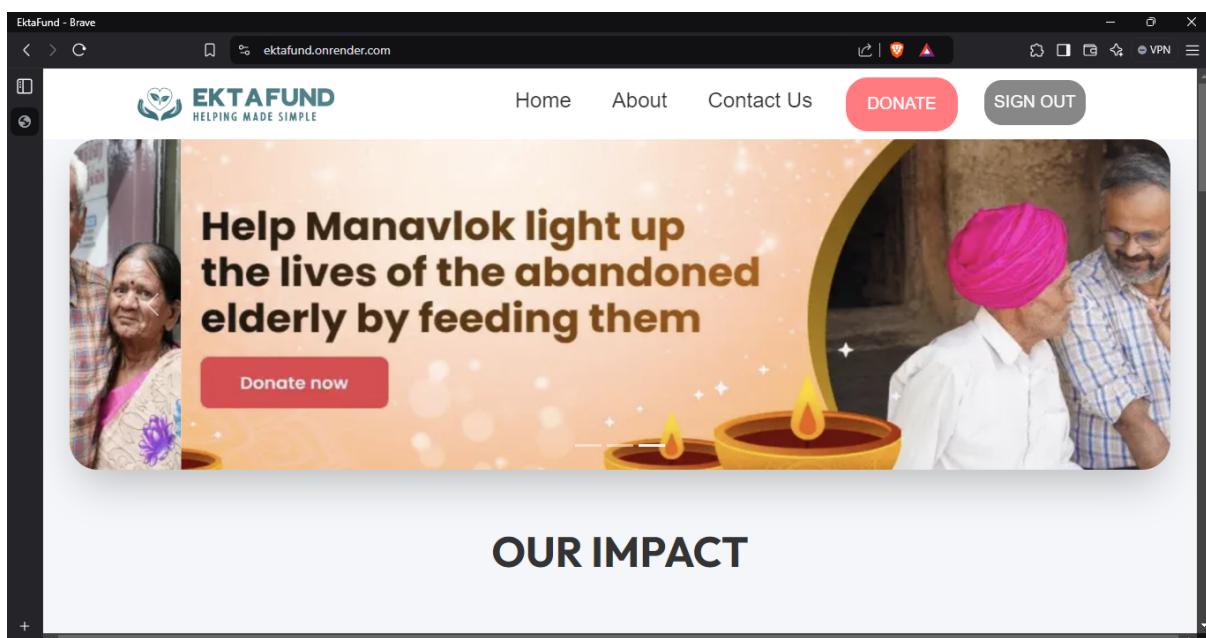


Figure 4.2: Brave browser usablities

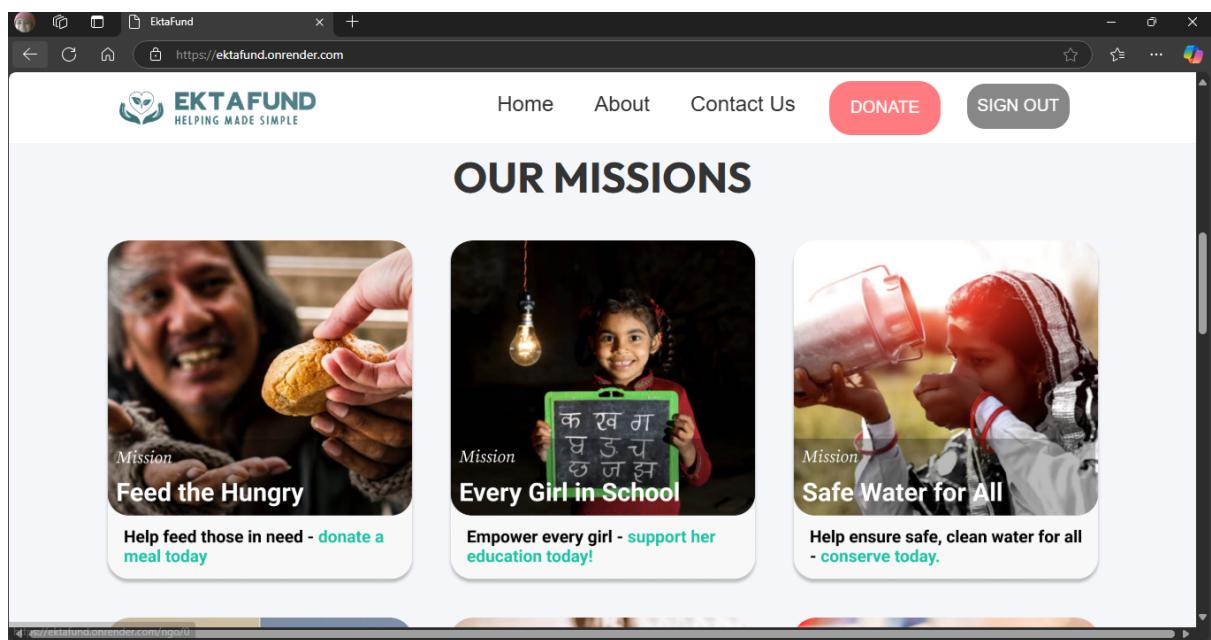


Figure 4.3: Edge browser usablities

Chapter 5

Security Testing using Pentest-tool

5.1 Overview of Pentest-Tools

Pentest-Tools is an online platform designed to facilitate penetration testing, which evaluates the security of systems, networks, and web applications. It provides a suite of automated tools to identify vulnerabilities, assess configurations, and ensure compliance with security standards.

Key features of Pentest-Tools include vulnerability scanning, exploit detection, web application security testing, and network diagnostics. By simulating cyberattacks, it helps in proactively identifying potential threats and ensuring the system remains robust against unauthorized access or data breaches.

In the context of Non-Functional Requirements (NFR) testing, Pentest-Tools is primarily used for:

- **Security Testing:** Ensures that the application adheres to security best practices by identifying vulnerabilities such as SQL injection, XSS, and misconfigurations.
- **Reliability Testing:** Assesses the system's resistance to malicious attacks that could compromise availability or functionality.
- **Data Privacy:** Evaluates the security mechanisms that protect sensitive user and donor data.

By integrating Pentest-Tools in NFR testing, organizations can strengthen their systems against potential threats while ensuring the reliability and safety of the platform for end users.

Website Vulnerability Scanner Report

✓ <https://demo-frontend-hli3.onrender.com/>

! The Light Website Scanner didn't check for critical issues like SQLi, XSS, Command Injection, XXE, etc. [Upgrade to run Deep scans with 40+ tests and detect more vulnerabilities.](#)

Summary

Overall risk level:
Low

Risk ratings:
High: 0
Medium: 0
Low: 4
Info: 15

Scan information:
Start time: Nov 21, 2024 / 11:10:19 UTC+0530
Finish time: Nov 21, 2024 / 11:10:47 UTC+0530
Scan duration: 28 sec
Tests performed: 19/19
Scan status: **Finished**

Findings

FLAG Missing security header: Content-Security-Policy

CONFIRMED

URL	Evidence
https://demo-frontend-hli3.onrender.com/	Response does not include the HTTP Content-Security-Policy security header or meta tag Request / Response

▼ Details

Risk description:

The risk is that if the target application is vulnerable to XSS, lack of this header makes it easily exploitable by attackers.

Recommendation:

Configure the Content-Security-Header to be sent with each HTTP response in order to apply the specific policies needed by the application.

References:

https://cheatsheetseries.owasp.org/cheatsheets/Content_Security_Policy_Cheat_Sheet.html
<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Security-Policy>

Classification:

CWE : [CWE-693](#)

OWASP Top 10 - 2017 : [A6 - Security Misconfiguration](#)

OWASP Top 10 - 2021 : [A5 - Security Misconfiguration](#)

FLAG Missing security header: Referrer-Policy

CONFIRMED

URL	Evidence
https://demo-frontend-hli3.onrender.com/	Response headers do not include the Referrer-Policy HTTP security header as well as the <meta> tag with name 'referrer' is not present in the response. Request / Response

▼ Details

Risk description:

The risk is that if a user visits a web page (e.g. "http://example.com/pricing/") and clicks on a link from that page going to e.g. "https://www.google.com", the browser will send to Google the full originating URL in the `Referer` header, assuming the Referrer-Policy

header is not set. The originating URL could be considered sensitive information and it could be used for user tracking.

Recommendation:

The Referrer-Policy header should be configured on the server side to avoid user tracking and inadvertent information leakage. The value `no-referrer` of this header instructs the browser to omit the Referer header entirely.

References:

https://developer.mozilla.org/en-US/docs/Web/Security/Referer_header:_privacy_and_security_concerns

Classification:

CWE : [CWE-693](#)

OWASP Top 10 - 2017 : [A6 - Security Misconfiguration](#)

OWASP Top 10 - 2021 : [A5 - Security Misconfiguration](#)

FLAG Robots.txt file found

CONFIRMED

URL

<https://demo-frontend-hli3.onrender.com/robots.txt>

▼ Details

Risk description:

There is no particular security risk in having a robots.txt file. However, it's important to note that adding endpoints in it should not be considered a security measure, as this file can be directly accessed and read by anyone.

Recommendation:

We recommend you to manually review the entries from robots.txt and remove the ones which lead to sensitive locations in the website (ex. administration panels, configuration files, etc).

References:

<https://www.theregister.co.uk/2015/05/19/robotstxt/>

Classification:

OWASP Top 10 - 2017 : [A6 - Security Misconfiguration](#)

OWASP Top 10 - 2021 : [A5 - Security Misconfiguration](#)

FLAG Server software and technology found

UNCONFIRMED ⓘ

Software / Version	Category
 Bunny	CDN
 Bootstrap 5.3.0	UI frameworks
 Google Font API	Font scripts
 HTTP/3	Miscellaneous
 React	JavaScript frameworks
 Cloudflare	CDN
 jsDelivr	CDN
 HSTS	Security

▼ Details

Risk description:

The risk is that an attacker could use this information to mount specific attacks against the identified software type and version.

Recommendation:

We recommend you to eliminate the information which permits the identification of software platform, technology, server and operating system: HTTP server headers, HTML meta information, etc.

References:

https://owasp.org/www-project-web-security-testing-guide/stable/4-Web_Application_Security_Testing/01-Information_Gathering/02-Fingerprint_Web_Server.html

Classification:

OWASP Top 10 - 2017 : A6 - Security Misconfiguration

OWASP Top 10 - 2021 : A5 - Security Misconfiguration

 **Security.txt file is missing**CONFIRMED**URL**Missing: <https://demo-frontend-hii3.onrender.com/.well-known/security.txt> **Details****Risk description:**

There is no particular risk in not having a security.txt file for your server. However, this file is important because it offers a designated channel for reporting vulnerabilities and security issues.

Recommendation:

We recommend you to implement the security.txt file according to the standard, in order to allow researchers or users report any security issues they find, improving the defensive mechanisms of your server.

References:<https://securitytxt.org/>**Classification:**

OWASP Top 10 - 2017 : A6 - Security Misconfiguration

OWASP Top 10 - 2021 : A5 - Security Misconfiguration

 **Website is accessible.** **Nothing was found for vulnerabilities of server-side software.** **Nothing was found for client access policies.** **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 missing HTTP header - X-Content-Type-Options.** **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.

└ Nothing was found for unsafe HTTP header Content Security Policy.

Scan coverage information

List of tests performed (19/19)

- ✓ Starting the scan...
- ✓ Checking for missing HTTP header - Content Security Policy...
- ✓ Checking for missing HTTP header - Referrer...
- ✓ Checking for website technologies...
- ✓ 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 missing HTTP header - X-Content-Type-Options...
- ✓ Checking for domain too loose set for cookies...
- ✓ Checking for HttpOnly flag of cookie...
- ✓ Checking for Secure flag of cookie...
- ✓ Checking for unsafe HTTP header Content Security Policy...

Scan parameters

Target: https://demo-frontend-hli3.onrender.com/
Scan type: Light
Authentication: False

Scan stats

Unique Injection Points Detected:	7
URLs spidered:	7
Total number of HTTP requests:	16
Average time until a response was received:	248ms