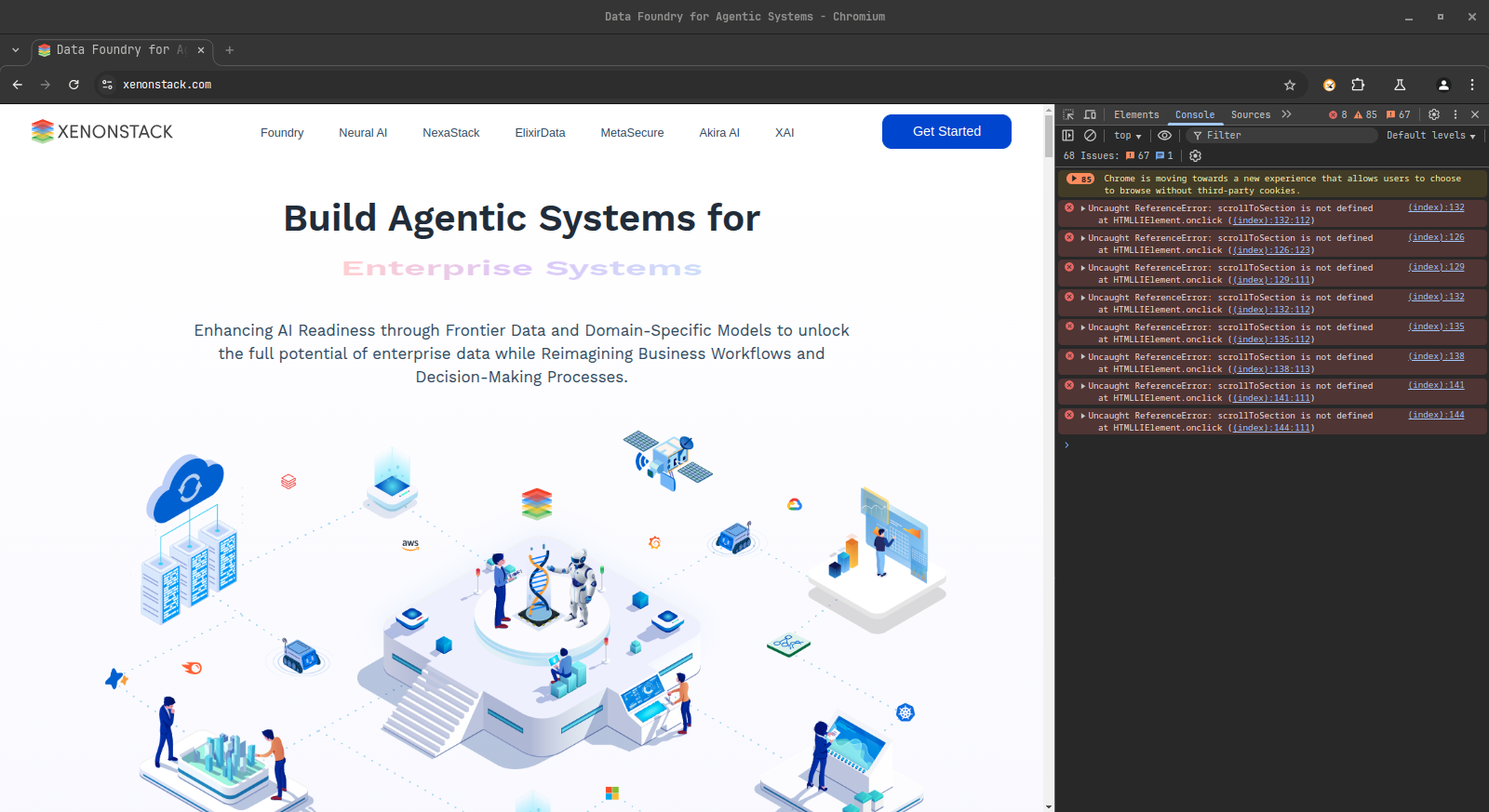
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case Report | | | | | | | | | | |
| **ID** | **Title** | **Description** | **Test Case Included** | | | | | | | |
| **TC\_ID** | **TC\_Title** | **TC\_Description** | **TC\_Steps** | **Expected Result** | **Actual Result** | **Status** | **Priority** |
| **TS001** | A user wants to check the links in the navigation bar. | A User wants to check any of the links in the navigation bar pointing towards a product of the company. | **FTC001** | Check Navigation Bar | We need to verify if the links in the navigation bar work or not | 1. Visit xenonstack.com 2. Click on all the links in the navigation bar | The links in the navigation bar should redirect us to the desired page | Most of the navigation bar links do not work, which is not what we were expecting. | Fail ❌ | High |
| TS002 | A user wants to fill the `Get Started` Form | A User wanted to `Get Started` with the company so they decided to fill in the form and put in a request. | **FTC002** | Check `Get Started` Form for Valid Input | We need to verify if the form passes the valid inputs | 1. Visit xenonstack.com 2. Click on `**Get Started**` 3. Fill the Form with valid data 4. Click Submit 5. Click on various options the next page of the form has 6. Click Submit | The form should be submitted properly with no errors mentioned. Also, we should be provided with an acknowledgement for the operation. | Successfully submitted the form. | Pass ✅ | High |
| **FTC003** | Check `Get Started` Form with no data | We need to verify if the form works with empty data. | 1. Visit xenonstack.com 2. Click on `**Get Started**` 3. Click on Submit | The form should not allow us to go to the next page with empty data fields. | The form provides error message for any field that has not been field properly. | Pass ✅ | High |
| **FTC004** | Check `Get Started` Form with invalid data | We need to verify if the form accepts invalid credentials. | 1. Visit xenonstack.com 2. Click on `**Get Started**` 3. Fill the form with invalid data. 4. Click on Submit | The form should not allow any invalid data and should prompt the user to enter valid data | The form provides error messages for any field that does not have valid data. | Pass ✅ | High |
| TS003 | A malicious user wants access to the website and performs injection attacks | A hacker wants to gain privilege access to the systems. So, they enter some malicious code into the input field of the form. | **FTC005** | Check `Get Started` Form with SQL Injection | We need to verify if the form is safe from SQL injection attacks | 1. Visit xenonstack.com 2. Click on `**Get Started**` 3. Fill the form with the data: **' OR '1'='1'; --** 4. Click on Submit | The form should hold its ground and should display error messages due to the data passes being invalid | The form successfully displays error messages for any field where the injection code is entered. | Pass ✅ | High |
| **FTC006** | Check `Get Started` Form with Cross Site Scripting (XSS) | We need to verify if the form is safe from XSS attacks | 1. Visit xenonstack.com 2. Click on `**Get Started**` 3. Fill the form with the data: **<script>alert('Vulnerable to XSS')</script>** 4. Click on Submit | The form should hold its ground and should display error messages due to the data passes being invalid and should not parse the javascript code that has been given to the input fields | The form successfully displays error messages for any field where the malicious code is entered. | Pass ✅ | High |
| TS004 | A user wants to visit some links in the footer of the website. | Check if the links in the footer are all valid or not. | **FTC007** | Check the footer of the page. | We need to verify if all the footer links in the page work properly or not | 1. Visit xenonstack.com 2. Go to the end of the page. 3. Click all the links in the footer. | The links should be opened successfully without any discrepancies. | Some links are not available. | Fail ❌ | High |
| TS005 | A user wants to quickly visit the website's pages. | Check if all the links on the website have good performance for better user experience. | **FTC008** | Check the load speed of each page | We need to verify if all the pages are responsive enough and are loading under 4s which is the standard for fast and responsive websites | 1. Visit xenonstack.com 2. Click on all the links available in the website | All the subpages listed in xenonstack.com should load under 4s. | Some pages take more than 4s to load. | Fail ❌ | Medium |
| TS006 | A user by mistake has visited an unrecognized page or invalid page. | Check if the page properly displays the mistake that the user made. | **FTC009** | Check invalid pages | We need to verify if the website properly displays a error 404 message if the user tries to visit a non exisiting page | 1. Visit xenonstack.com/non-existing | Should display an error 404 message and a message stating that the page does not exist. | Displays error 404 and a message stating that the page does not exist. | Pass ✅ | Low |
| TS007 | The company’s website is under heavy load every day. | A large group of people constantly keep visiting the website to read the companies blogs and articles. | **NFTC010** | Check if the website can take a heavy load of users properly | We need to perform a load test to check if the platform can handle heavy traffic. | NIL | Should have minimal or no downtime for heavy loads with response time less than 10 sec | Only 78% of the test maintain a response time of 10sec. | Fail ❌ | High |
| TS008 | Sudden Spike in userbase | The companies blogs or products are shared in an online forum where users wanted to see that website on their own. This led to a sudden spike in network traffic. Check if the website can handle this sudden increase in demand. | **NFTC011** | Check if the website can take a sudden surge of users | We need to check if the website can take a sudden spike in traffic | NIL | Should maintain internet traffic with ease and have response time less than 10sec | Only 96% of the users get successfull connection and only 29% get a response from the website under 10sec | Fail ❌ | High |
| TS009 | Increasing demand and traffic of the company’s user base. | With more advancement and more publicity, the company's website seems to be getting popular day by day. Check if it can handle all this increasing traffic properly. | **NFTC012** | Check if the website can take an increasing amount of network traffic over the time | We need to check if the website is ready to be used by a growing community and check if it can hold to a increasing daily user base. | NIL | The website should successfully meet its user base requirements meaning it should have minimal downtime even if the user count is high with the maximum response time being 10sec | Not a single connection can be maintained. | Fail ❌ | High |
| TS010 | Increasing demand for the company’s products leads to higher `Get Started` form filling. | Check if the form api can handle all the stress and mange the data properly and check if duplicate data is passed and is processed or not. | **NFTC013** | Check if the form can handle large amount of data going in and out. | Check the HubSpot Api for the form that is being used to send data from the form to the server. | NIL | The form should avoid duplicate data as much as possible but most importantly should process each user's data properly. | Only 47% successfully submitted. | Fail ❌ | High |

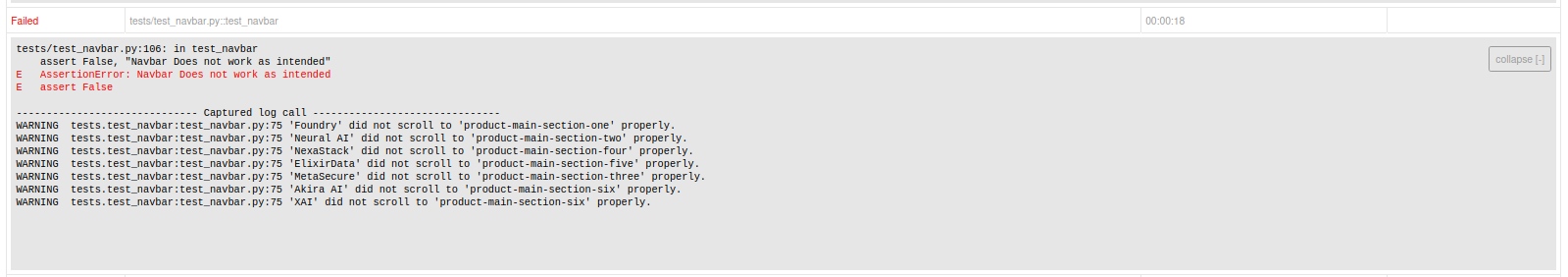
|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bug Report** | | | | | | | | | | | |
| **Bug ID** | **Test Case ID** | **Title** | **Description** | **Steps to Reproduce** | **Expected Result** | **Actual Result** | **Severity** | **Priority** | **Environment** | **Attachments** | **Bug Status** |
| **BUG001** | FTC001 | Navbar Items Not Scrolling to Correct Sections | Several navbar items do not scroll to their intended sections when clicked. | 1. Visit xenonstack.com 2. Click on each navbar item (Foundry, Neural AI, NexaStack, etc.). 3. Observe if the page correctly scrolls to the respective section. | Clicking a navbar item should scroll the page to the correct section. | Navbar items failed to scroll to their respective sections (Foundry, Neural AI, NexaStack, etc.). | Medium | Medium | Browser: Chromium  OS: Linux-6.8.0-54-generic-x86\_64-with-glibc2.39 | {[SCREENSHOT](#_BUG001)}  {[LOGs](#_BUG001_LOGS)} | Open |
| **BUG002** | FTC007 | Missing/Non-functional Footer Links | Several links in the footer do not have href attributes, making them non-functional. Additionally, the 'Data Protection' link did not navigate correctly. | 1. Visit xenonstack.com 2. Scroll to the footer. 3. Check if all footer links have href attributes. 4. Click on each footer link to verify navigation. | All footer links should contain valid href attributes and navigate correctly. | Several links are missing href attributes, and 'Data Protection' does not navigate properly. | High | Medium | Browser: Chromium  OS: Linux-6.8.0-54-generic-x86\_64-with-glibc2.39 | {[SCREENSHOT](#_BUG002)}  {[LOGs](#_BUG002_LOGS)} | Open |
| **BUG003** | FTC008 | Page Load Time Exceeds 4 Seconds | Some pages take too long to load (>4 seconds), impacting performance. | 1. Open xenonstack.com 2. Click on internal links such as <https://www.xenonstack.com/readiness-assessment/data-protection/> and <https://www.xenonstack.com/generative-ai> 3. Measure page load times. | Pages should load in less than 3 seconds for optimal performance. | Some internal pages took **more than 4 seconds** to load. | Medium | High | Browser: Chromium  OS: Linux-6.8.0-54-generic-x86\_64-with-glibc2.39 | {[SCREENSHOT](#_BUG003_NaN)}  {[LOGs](#_BUG003_LOGS)} | Open |
| **BUG004** | NFTC013 | API Requests Failing at a High Rate (80.59%) | During load testing, 80.59% of API requests failed due to timeouts when submitting data to https://api.hsforms.com/submissions/v3/integration/submit/8161231/22ff0c52-d126-484e-a693-64f0a2082746. | 1. Run a load test using k6 run api\_load\_test.js 2. Simulate 80 concurrent virtual users sending API requests 3. Observe that 80.59% of requests fail due to timeouts. | API should handle concurrent requests efficiently with minimal failures. | 2130 out of 2643 requests failed due to timeouts. | Critical | High | k6 API Load Test, Linux | {[LOGS](#_BUG004_LOGS)} | Open |
| **BUG005** | NFTC012 | 100% API Request Failure Under High Load | All 5,852 API requests failed under stress conditions with 100 concurrent virtual users. The server did not respond successfully even once. | 1. Run k6 run stress\_test.js with 100 virtual users. 2. Observe the response status for each request. 3. Notice that every request fails. | The API should handle at least 50% of the requests successfully under stress conditions. | 0% success rate, 100% failure across all 5,852 requests. | Critical | High | k6 Stress Test, Linux | {[LOGS](#_BUG005_LOGS)} | Open |
| **BUG006** | NFTC011 | Connection Failures Under Sudden Traffic Surge | During a sudden spike of 300 concurrent users, multiple requests failed due to unexpected EOF errors and HTTP/2 connection failures, indicating instability when handling rapid traffic increases. | 1. Run k6 run spike\_test.js with 300 virtual users. 2. Observe API responses. 3. Notice 13 failed requests due to connection failures. | The server should gracefully handle sudden traffic spikes without connection issues. | Connection failures occurred intermittently, leading to dropped requests. | High | High | k6 Spike Test, Linux | {[LOGS](#_BUG006_LOGS)} | Open |
| **BUG007** | NFTC010 | API Response Time Exceeded 10s for 79 Requests | During the load test with 100 virtual users, 78% of requests were successful within 10s, but 79 requests exceeded 10 seconds, with the slowest taking 34.12 seconds. | 1. Run k6 run load\_test.js with 100 virtual users. 2. Monitor response times. 3. Observe that some requests exceed 10s, causing potential timeouts. | The API should handle high load efficiently, keeping response times below 10s. | Some requests took up to 34.12 seconds, leading to performance degradation. | High | High | k6 Load Test, Linux | {[LOGS](#_BUG007_LOGS)} | Open |

Bug Images:

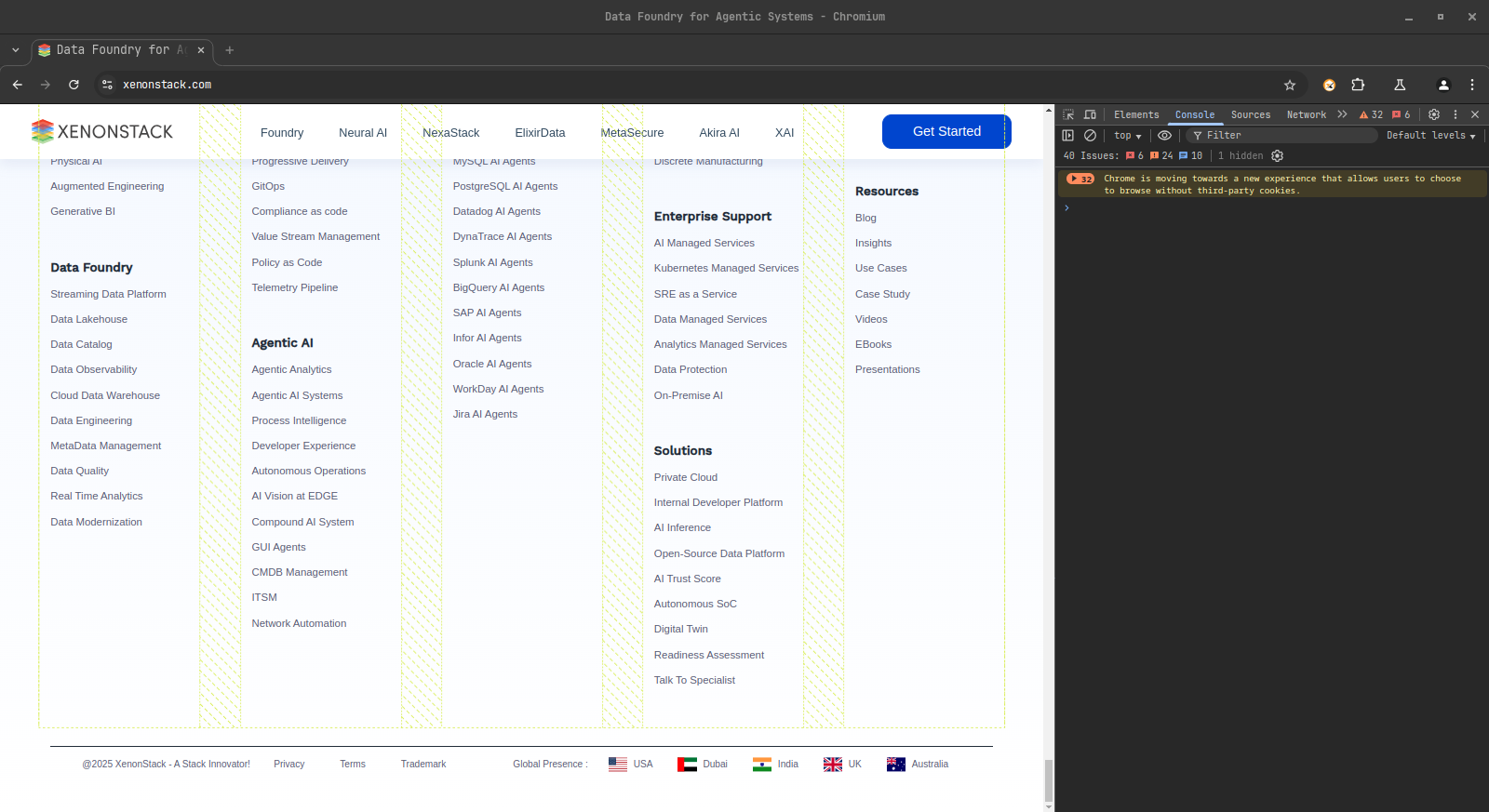
# BUG001



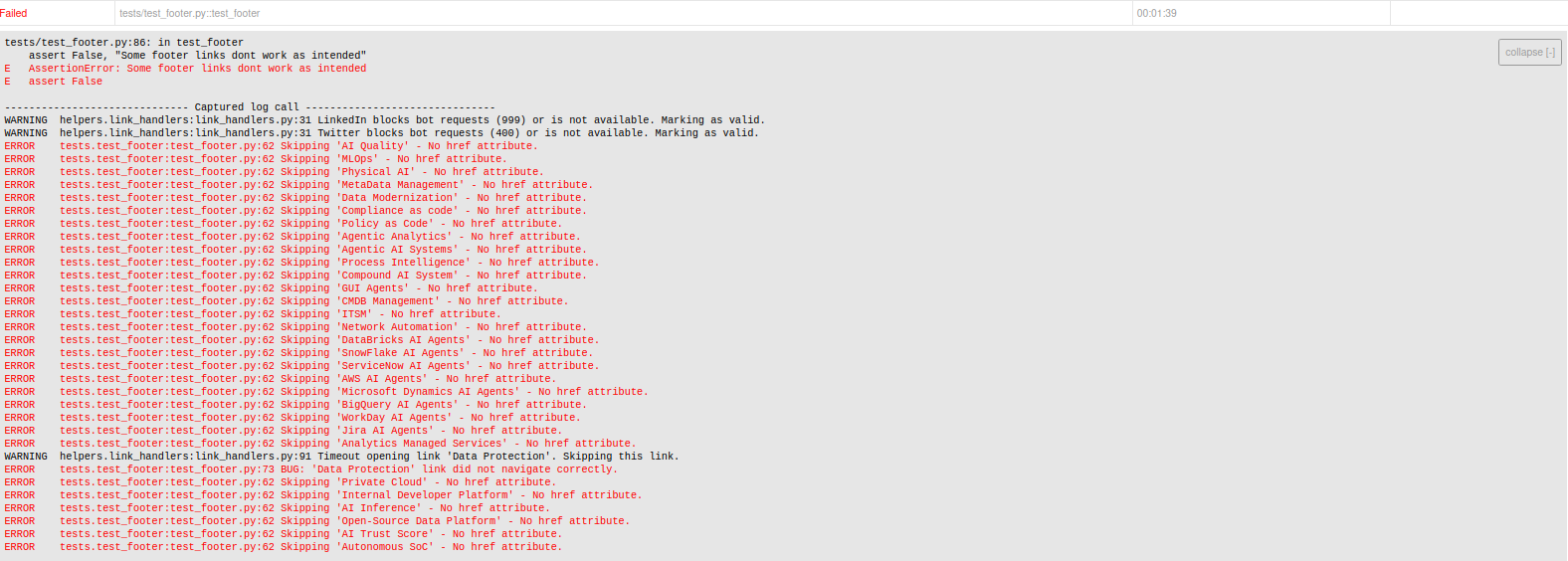
# BUG001\_LOGS



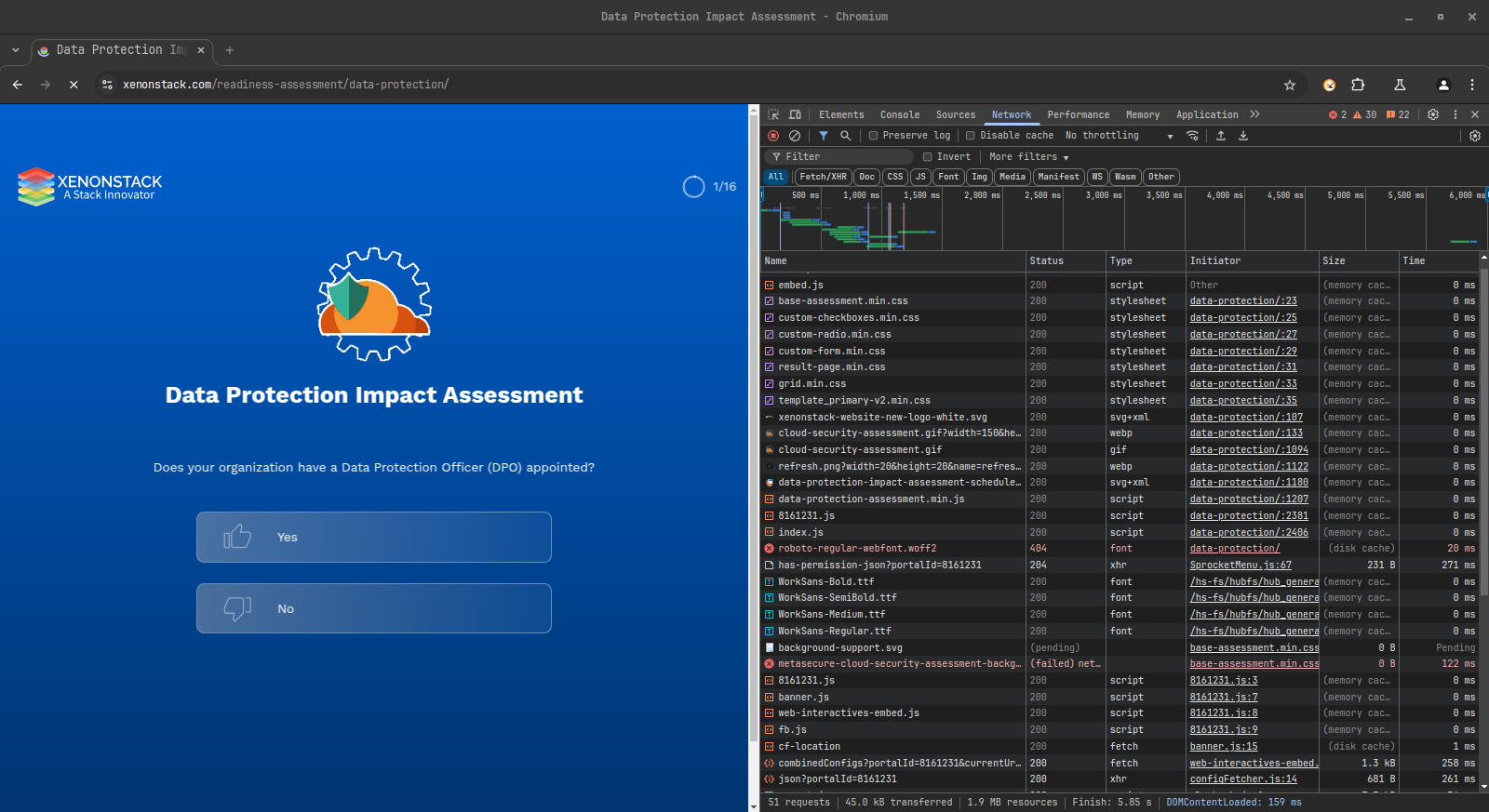
# BUG002



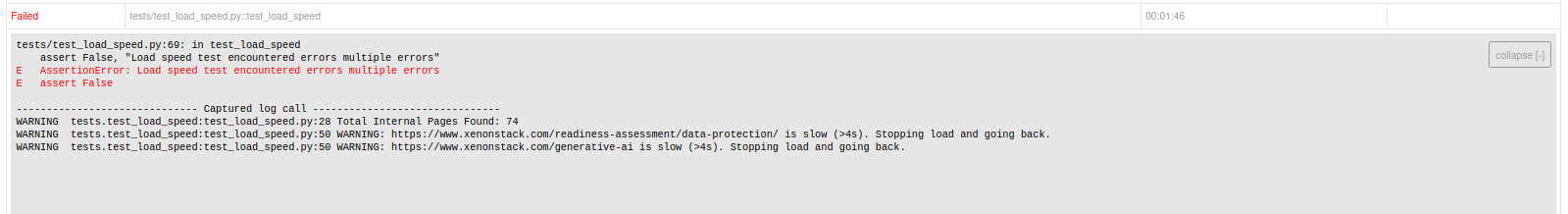
# BUG002\_LOGS



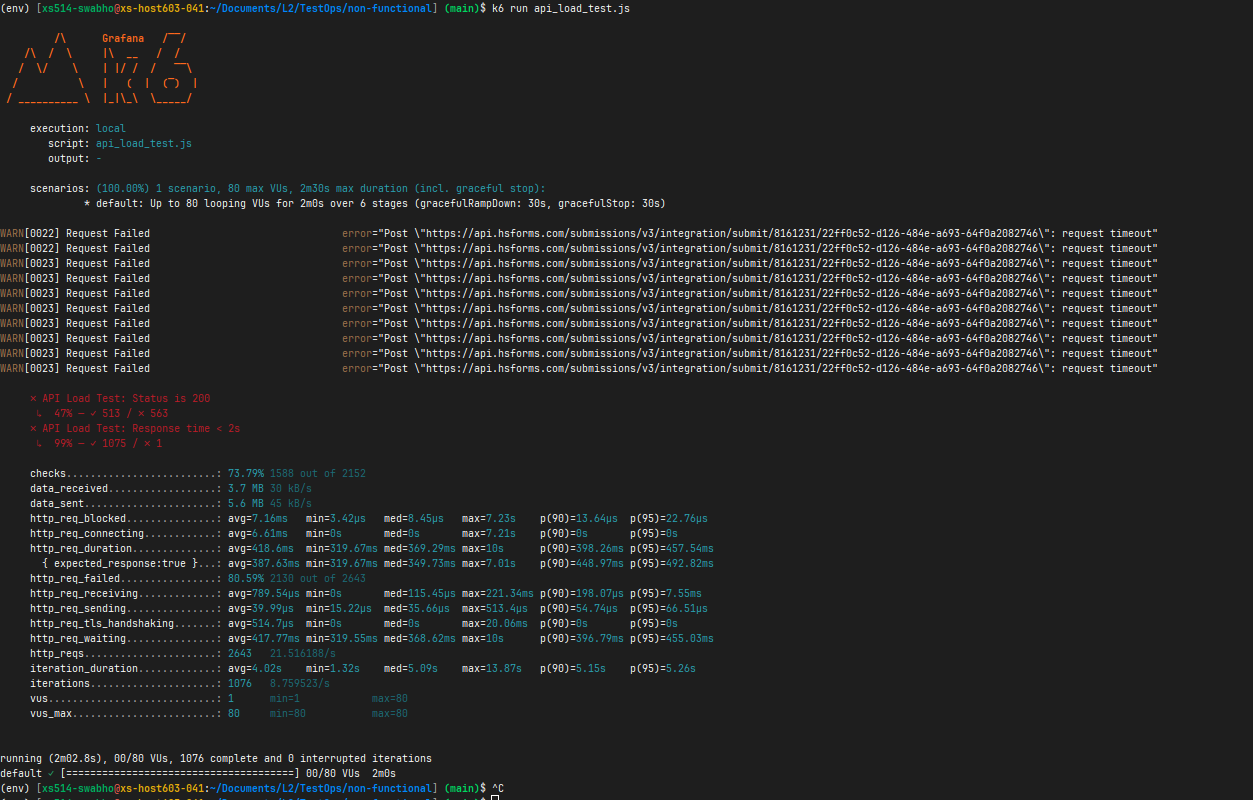
# BUG003



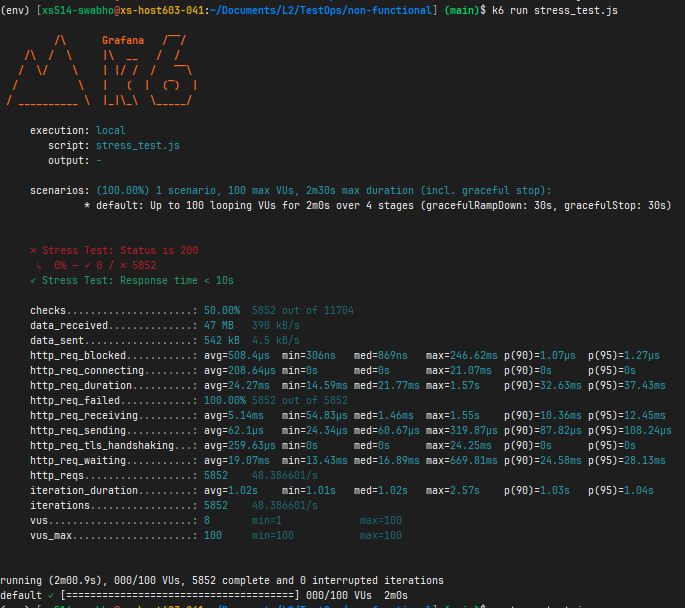
# BUG003\_LOGS



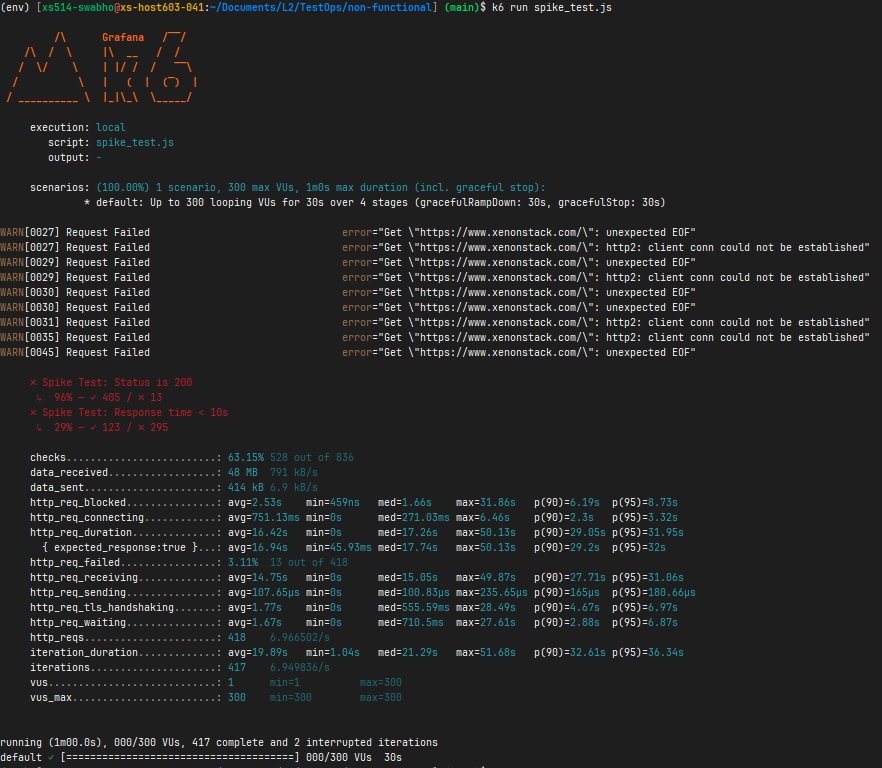
# BUG004\_LOGS



# BUG005\_LOGS



# BUG006\_LOGS



# BUG007\_LOGS

