PES UNIVERSITY
Department of Computer Science and Engineering
UE21CS341A: Software Engineering

Browser Compatibility Test-based Network Connectivity (Library Search System)

5TH SEM, B.Tech (CSE)

Aug – Dec 2023

Final Report

Version 1.0 approved

Prepared by: Software Squad

SUBMITTED BY

S No.	Name	SRN
1.	N Shreelekha	PES1UG21CS353
2.	Meghana N	PES1UG21CS334
3.	Mridvi Sharma	PES1UG21CS343
4.	Mullapudi Jahnavi	PES1UG21CS346

Proposal of the Project

In the dynamically evolving landscape of web applications, ensuring seamless user experience across various browsers and under diverse network conditions is crucial. This project, titled "Browser Compatibility Testing for Network Connectivity," aims to address this challenge by systematically testing web applications under different network scenarios. The primary focus is on identifying and mitigating issues related to rendering, functionality, and overall user experience.

The primary purpose of this project is to conduct rigorous testing of the web application to validate its functionality, responsiveness, and overall behavior under varying network conditions and across different browsers. The goal is to gather insights into how the application behaves in real-world situations. This project conducts a comprehensive assessment of how web applications respond to different network conditions. By emulating real-world scenarios, the project seeks to uncover potential challenges related to browser compatibility and network connectivity.

Overall Goal of the Project:

Browser Compatibility: *Verifying that the web application functions as intended on popular web browsers, including Mozilla Firefox, Google Chrome, Internet Explorer, Safari, etc.*

Network Condition Resilience: *Evaluating the application's performance under different network conditions, such as slow and unreliable connections, to ensure accessibility for users.*

The objective of this project includes:

1. Identify and address rendering inconsistencies across various browsers.

2. Assess the impact of slow network connections on complete rendering and functionality.

3. Ensure critical elements of web applications are not overlooked during rendering on slow connections.

4. Evaluate how ISP speed affects page rendering across different browsers.

5. Validate the responsiveness of web applications under diverse network conditions.

6. Conduct cross-browser testing to ensure compatibility and consistency.

7. Validating the responsiveness of web applications when proxy extensions are enabled.

8. Validating if website has been added to the firewall block.

Methodology:

Our methodology involves implementing a set of test cases, each targeting specific aspects of browser compatibility and network connectivity. These include checks for partial website loading, rendering issues, missing essential elements, and the impact of Internet Service Provider (ISP) speed on overall page rendering. Through user-centric testing, we emphasize maintaining a consistent and user-friendly experience across different browsers. Using Selenium WebDriver, a powerful automation tool, we conduct comprehensive tests across popular web browsers and varying network speeds. To execute these tests, we employ Python scripts incorporating Selenium WebDriver functionalities. These scripts simulate user interactions, navigating through web pages, and capturing screenshots to visually validate test outcomes. . The project's scope encompasses evaluating the web application's performance on browsers such as Mozilla Firefox, Google Chrome, Internet Explorer, Safari, and more. Additionally, we validated its resilience under challenging network conditions, including slow connections.

Use cases:

Use Case 1: Real-world Browser Simulation

Objective: Simulate real-world scenarios by testing the web application on popular browsers, including Mozilla Firefox, Google Chrome, Internet Explorer, safari etc.

Scenario: Users accessing the Library Search System can choose their preferred browser without compromising functionality, ensuring a consistent experience across different platforms.

Use Case 2: Low Bandwidth Simulation

Objective: Evaluate the application's performance under low bandwidth conditions to ensure accessibility for users with slower internet connections.

Scenario: Users in areas with limited internet speed can still utilize the Library Search System effectively without experiencing significant delays or issues in functionality.

Use Case 3: Cross-browser Consistency

Objective: Validate the cross-browser consistency of the web application, emphasizing a uniform user experience.

Scenario: Users switching between browsers should encounter the same features and functionality, irrespective of their chosen browser.

Use Case 4: Missing Element Detection

Objective: Identify and rectify missing or incorrectly rendered elements on the web page under different network conditions.

Scenario: Ensuring that crucial elements like titles, headers, and navigation options are consistently present and correctly displayed.

Use Case 5: ISP Speed Impact

Objective: Assess how variations in Internet Service Provider (ISP) speed affect the rendering of the Library Search System.

Scenario: Users accessing the web application from different ISPs can expect consistent performance, regardless of speed fluctuations.

Use Case 6: User-centric Testing

Objective: Prioritize user experience by conducting tests that focus on responsiveness and functionality under network constraints.

Scenario: Users with varying internet speeds should still find the Library Search System user-friendly and responsive, ensuring a positive interaction.

Test Cases

Test Case 1: Check if the website loads partially under a slow connection*.*

Explanation: This test checks how well the website handles slow network conditions. It's essential to ensure that even with a slow connection, users can still access some parts of the website while waiting for the rest to load.

Input: Slow internet connection (simulated or actual)

URL of the website

Expected Output: The website loads, but some resources take longer to load, resulting in a partial display of the webpage.

Test Case 2: Check if the website is rendered at all on a slow connection*.*

Explanation: This test focuses on extreme network conditions, ensuring that the website eventually loads, even if it takes a considerable amount of time. It's crucial to prevent a complete failure to load.

Input: Extremely slow internet connection (simulated or actual)

URL of the website

Expected Output: The website may take a significantly longer time to load, but it should eventually render completely.

Test Case 3: Check if the website misses important elements while completely rendering on a slow connection.

Explanation: In this scenario, the goal is to identify if crucial elements (like images, stylesheets, or scripts) fail to load under slow network conditions, potentially affecting the usability and appearance of the website.

Input: Slow internet connection (simulated or actual)

URL of the website

Expected Output: The website loads but may miss loading critical elements such as images, stylesheets, or scripts, impacting the overall user experience.

Test Case 4: Check if the ISP speed affects the rendering of the page across the browser.

Explanation: Different Internet Service Providers (ISPs) may have varying speeds and network characteristics. This test ensures that the website renders consistently across different browsers and is not disproportionately affected by the speed of a specific ISP.

Input: Varying internet speeds from different ISPs

Different browsers and URL of the website

Expected Output: The rendering speed and performance may vary across different browsers and ISPs, affecting the user experience.

Test Case 5: Check if the browser responds well to a slow connection.

Explanation: The user experience during slow network conditions is crucial. This test evaluates how well the browser handles slow connections, whether it provides feedback to the user (e.g., loading spinners), and if it optimizes the loading process.

Input: Slow internet connection (simulated or actual) and URL of the website

Browser type and version

Expected Output: The browser should handle slow connections gracefully, providing a user-friendly experience, such as displaying loading indicators or optimizing resource loading.

Test Case 6: Use alternative browsers to access the website.

Explanation: This test ensures cross-browser compatibility. Different browsers may interpret and render web pages differently. It's crucial to verify that the website works uniformly across various popular browsers.

Input: Different browsers (e.g., Chrome, Firefox, Safari, Internet Explorer) and URL of the website

Expected Output: The website should render consistently across different browsers, ensuring compatibility and a uniform user experience.

Test Case 7: Disable proxies or VPN and try to access the website.

Explanation: Some users might have proxies or VPNs enabled for privacy or security reasons. This test checks if the website remains accessible when these features are disabled, ensuring that they don't interfere with the normal functioning of the site.

Input: Browser with a proxy or VPN enabled and URL of the website

Expected Output: Disabling proxies or VPN should not impact the website's accessibility, and it should load as expected.

Test Case 8: Enable proxy extensions and try to access the website.

Explanation: Conversely, this test evaluates whether enabling proxy extensions impacts the website's accessibility. Proxy extensions could affect how the browser interacts with the internet, and it's important to ensure they don't disrupt normal functionality.

Input: Browser with a proxy extension enabled and URL of the website

Expected Output: Enabling proxy extensions should not disrupt the website's functionality, and it should load as intended

Test Case 9: Cross-browser testing allows for ensuring the correct work of the app in different browser configurations.

Explanation: This is a broader test that involves checking the website's functionality across various browsers, including popular ones like Mozilla Firefox, Google Chrome, Internet Explorer, and Safari. It ensures a consistent and reliable user experience.

Input: Different browsers (Mozilla Firefox, Google Chrome, Internet Explorer, Safari, etc.) and URL of the website

Expected Output: The website should function correctly across various browsers, ensuring a consistent user experience.

Test Case 10: Check if any specific website has been added to the firewall block.

Explanation: Firewall settings can sometimes block access to specific websites. This test ensures that the website is not erroneously blocked by the firewall, leading to a failed connection.

Input: Firewall settings

URL of the website

Expected Output: If the website is blocked by the firewall, the browser should display an error or fail to connect.

Test tools used to test the scenario:

1. Check Partial Loading:

Test Objective: Verify if the website loads fully or partially under slow connection conditions.

Test Tool Used: Selenium with Chrome WebDriver.

Description:

- *The script initializes a headless Chrome browser using Selenium.*
- *Network conditions are set to simulate a slow 3G connection.*
- *The website is loaded, and the script checks if the page is fully loaded using `document.readyState`.*
- *The result is displayed based on the loading status.*

2. Check Browser Response:

Test Objective: Verify if the website is rendered on a slow connection.

Test Tool Used: Selenium with Chrome WebDriver.

Description:

- *A headless Chrome browser is launched.*
- *The website is loaded, and the script waits for 10 seconds for the page to load.*
- *A screenshot is taken to check if the website is rendered.*
- *The result is displayed based on the rendering status.*

3. Check Elements Rendered:

Test Objective: Check if a specific important element is rendered on a slow connection.

Test Tool Used: Selenium with Chrome WebDriver.

Description:

- *A Chrome browser is initialized using Selenium.*
- *Network conditions are set to simulate a slow 3G connection.*
- *The script navigates to the website and waits for the page to load.*
- *It finds an important element by its class name ('header').*

- *The result is displayed based on the visibility of the element.*

4. Check Speed:

Test Objective: Measure the rendering time of the website on a slow connection.

Test Tool Used: Selenium with Chrome WebDriver.

Description:

- *A Chrome browser is initialized using Selenium.*
- *The script records the start time, loads the website, and records the end time.*
- *The rendering time is calculated and displayed as the result.*

5. Check ISP Speed Affects Rendering:

Test Objective: Measure rendering time under different network conditions.

Test Tool Used: Selenium with Chrome WebDriver.

Description:

- *A Chrome browser is initialized using Selenium.*
- *Network conditions are set to simulate a slow connection.*
- *The script measures the rendering time and displays the result.*

6. Check Browser Responsive to Slow Connection:

Test Objective: Check if the browser is responsive under slow connection conditions.

Test Tool Used: Selenium with Chrome WebDriver.

Description:

- *A Chrome browser is initialized using Selenium.*
- *Network conditions are set using Chrome DevTools to simulate a slow connection.*
- *The script interacts with the Google search page and checks for responsiveness.*
- *The result is displayed based on the browser's responsiveness.*

7. Test Alternative Browsers:

Test Objective: Test website access with different browsers (Chrome, Edge, Firefox).

Test Tool Used: Selenium with Chrome, Edge, and Firefox Web Drivers.

Description:

- *The script iterates through different browsers, initializes each browser using Selenium, and attempts to access the website.*
- *The result is displayed for each browser, indicating success or failure.*

8. Access With Proxies Extensions:

Test Objective: Check if the website is accessible using a proxy extension.

Test Tool Used: Selenium with Chrome WebDriver.

Description:

- *A Chrome browser is initialized with a proxy extension using Selenium.*
- *The script attempts to access the website, and the result is displayed.*

9. Check Firewall Block:

Test Objective: Check if the website is accessible despite firewall restrictions.

Test Tool Used: Requests library for HTTP requests.

Description:

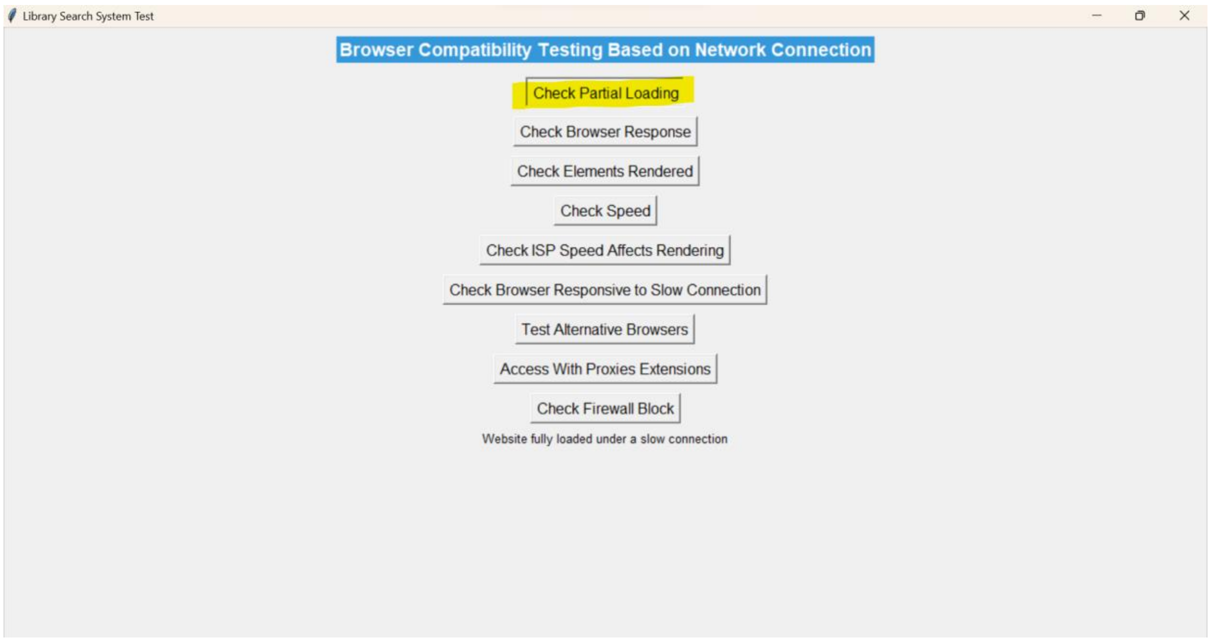
- *The script uses the `requests` library to send an HTTP GET request to the website.*
- *The response status code is checked to determine if the website is accessible.*
- *The result is displayed based on the response status.*

The primary testing tool used is Selenium, which allows browser automation and interaction. Additionally, the Requests library is employed for testing firewall block scenarios.

Output for Test Cases

Test Case 1: Check Partial Loading This test case simulates a slow 3G connection and checks if a website is fully loaded.

Expected Behaviour: The script opens a headless Chrome browser, sets network conditions to simulate slow 3G, and checks if the website is fully loaded. It should display whether the website is fully loaded or partially loaded.

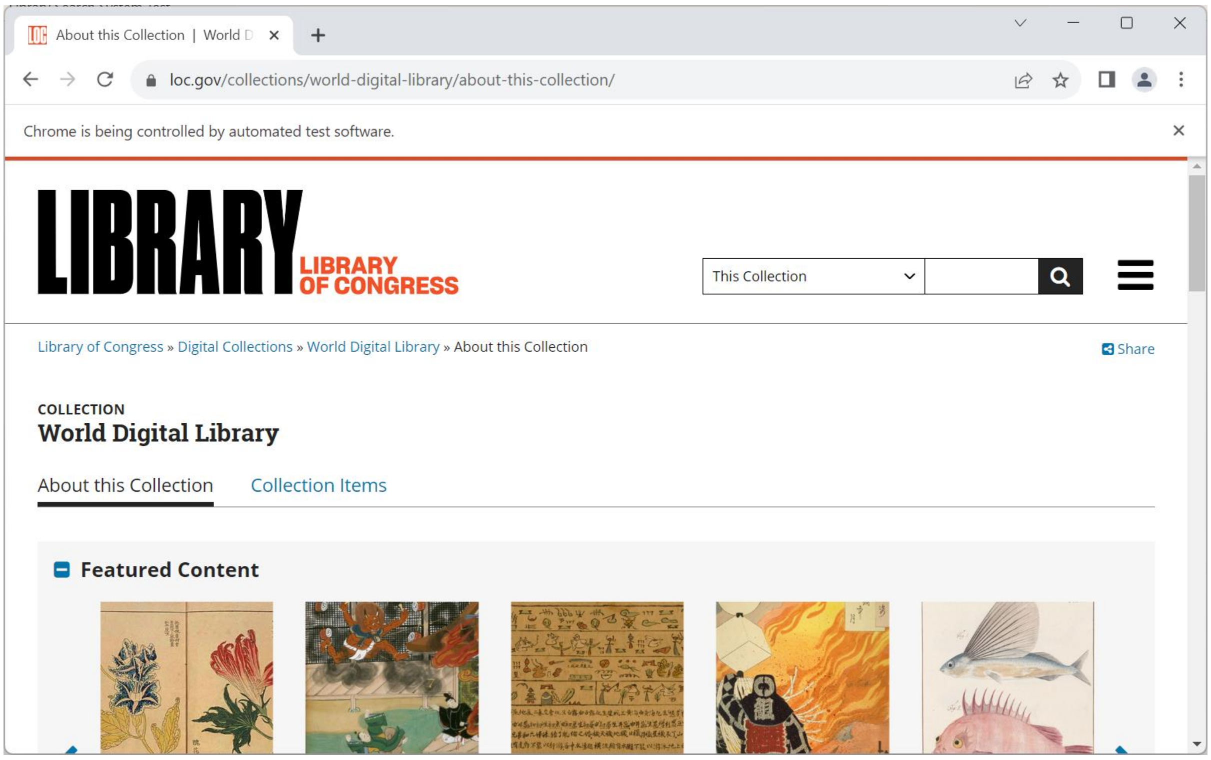

Library Search System Test
Browser Compatibility Testing Based on Network Connection
Check Partial Loading
Check Browser Response
Check Elements Rendered
Check Speed
Check ISP Speed Affects Rendering
Check Browser Responsive to Slow Connection
Test Alternative Browsers
Access With Proxies Extensions
Check Firewall Block

Website fully loaded under a slow connection

Test Case 2: Check Browser Response

This test case opens a headless browser, captures a screenshot, and checks if the website is rendered.

Expected Behavior: The script should open a browser, capture a screenshot, and display whether the website is rendered or not.


About this Collection | World D

loc.gov/collections/world-digital-library/about-this-collection/

Chrome is being controlled by automated test software.

LIBRARY LIBRARY OF CONGRESS

This Collection

Library of Congress » Digital Collections » World Digital Library » About this Collection

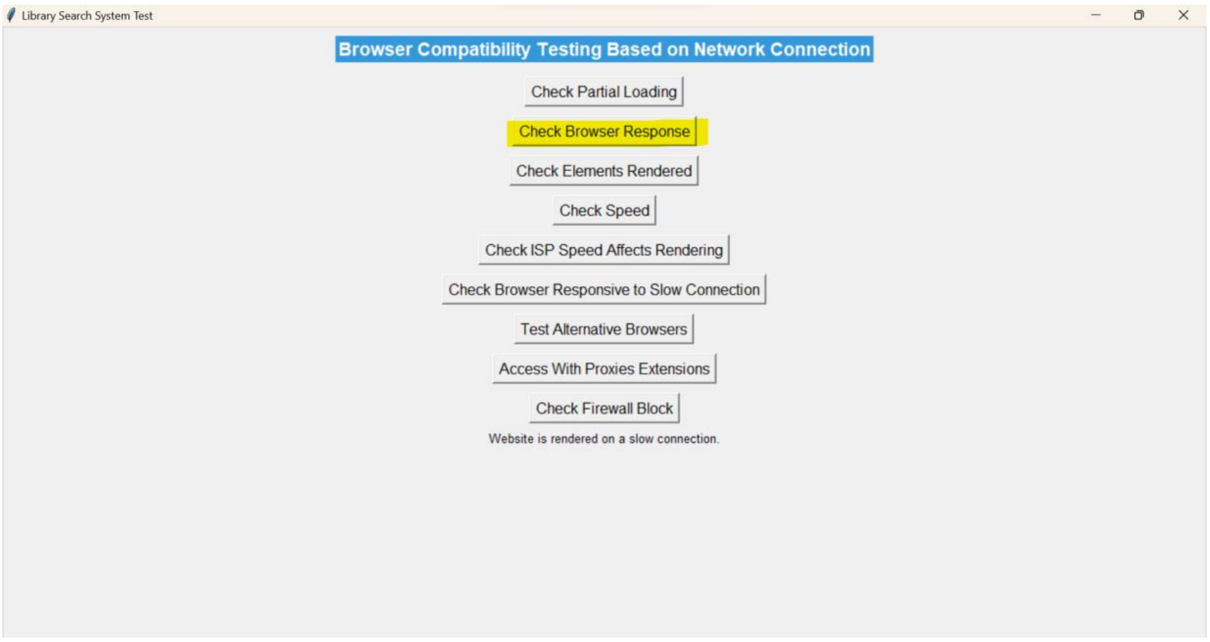
Share

COLLECTION

World Digital Library

About this Collection | Collection Items

Featured Content


Library Search System Test

Browser Compatibility Testing Based on Network Connection

Check Partial Loading

Check Browser Response

Check Elements Rendered

Check Speed

Check ISP Speed Affects Rendering

Check Browser Responsive to Slow Connection

Test Alternative Browsers

Access With Proxies Extensions

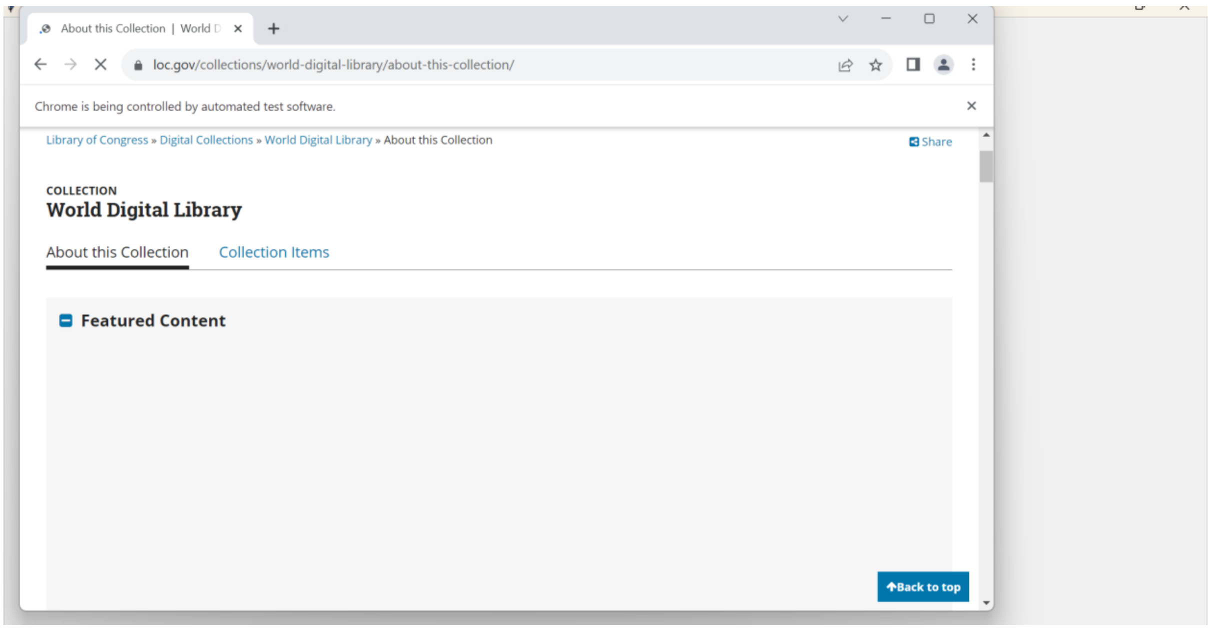
Check Firewall Block

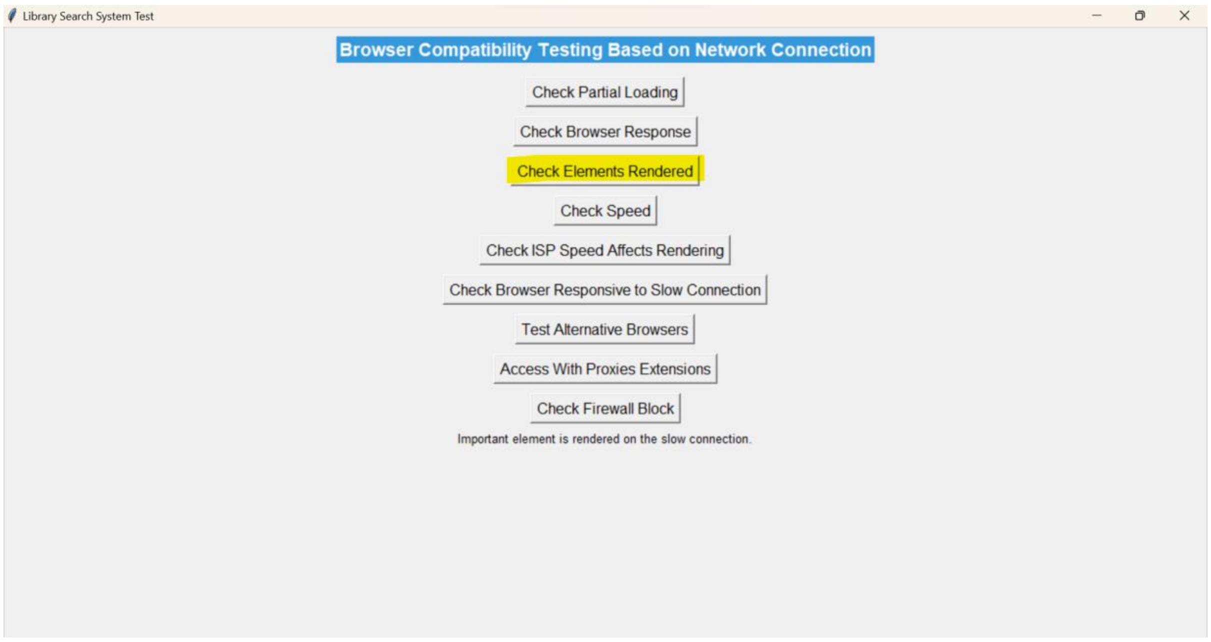
Website is rendered on a slow connection.

Test Case 3: Check Elements Rendered

This test case checks if a specific element (identified by class name 'header') is rendered on a slow connection.

Expected Behaviour: The script should open a browser, set network conditions, navigate to the website, wait for the page to load, and check if the specified element is rendered.


About this Collection | World D
loc.gov/collections/world-digital-library/about-this-collection/
Chrome is being controlled by automated test software.
Library of Congress » Digital Collections » World Digital Library » About this Collection
Share
COLLECTION
World Digital Library
About this Collection
Collection Items
Featured Content
Back to top

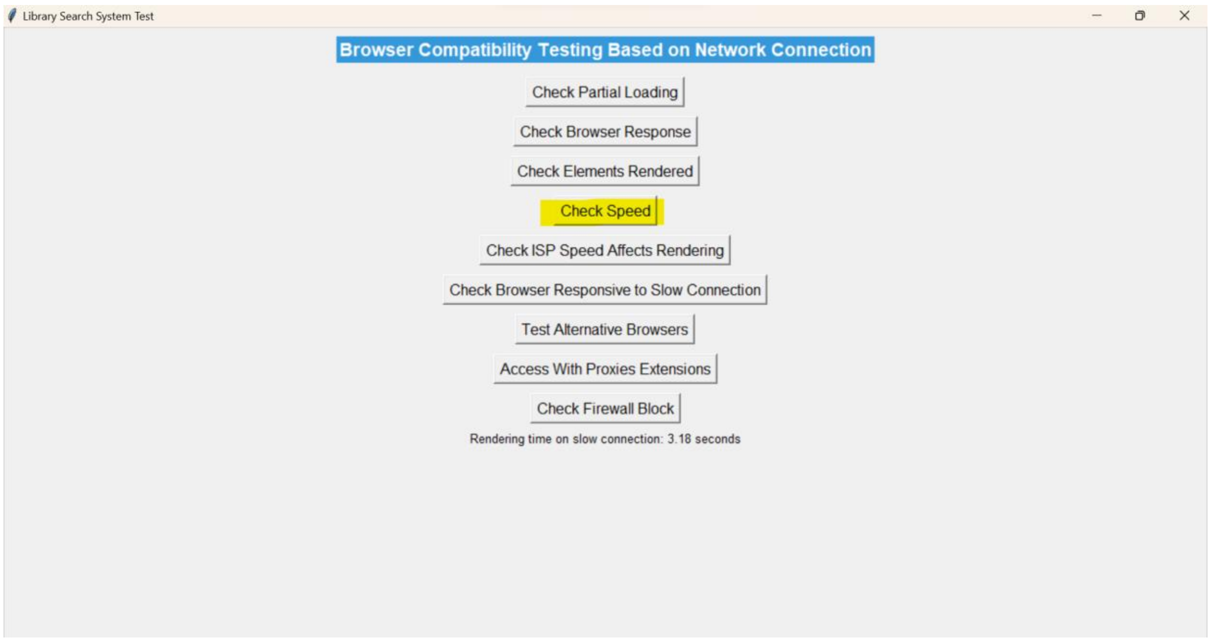

Library Search System Test
Browser Compatibility Testing Based on Network Connection
Check Partial Loading
Check Browser Response
Check Elements Rendered
Check Speed
Check ISP Speed Affects Rendering
Check Browser Responsive to Slow Connection
Test Alternative Browsers
Access With Proxies Extensions
Check Firewall Block

```
Important element is rendered on the slow connection.
```

Test Case 4: Check Speed

This test case measures the rendering time of a website on a slow connection.

Expected Behaviour: The script should open a browser, measure the rendering time, and display the time taken to render the website on a slow connection.


Library Search System Test

Browser Compatibility Testing Based on Network Connection

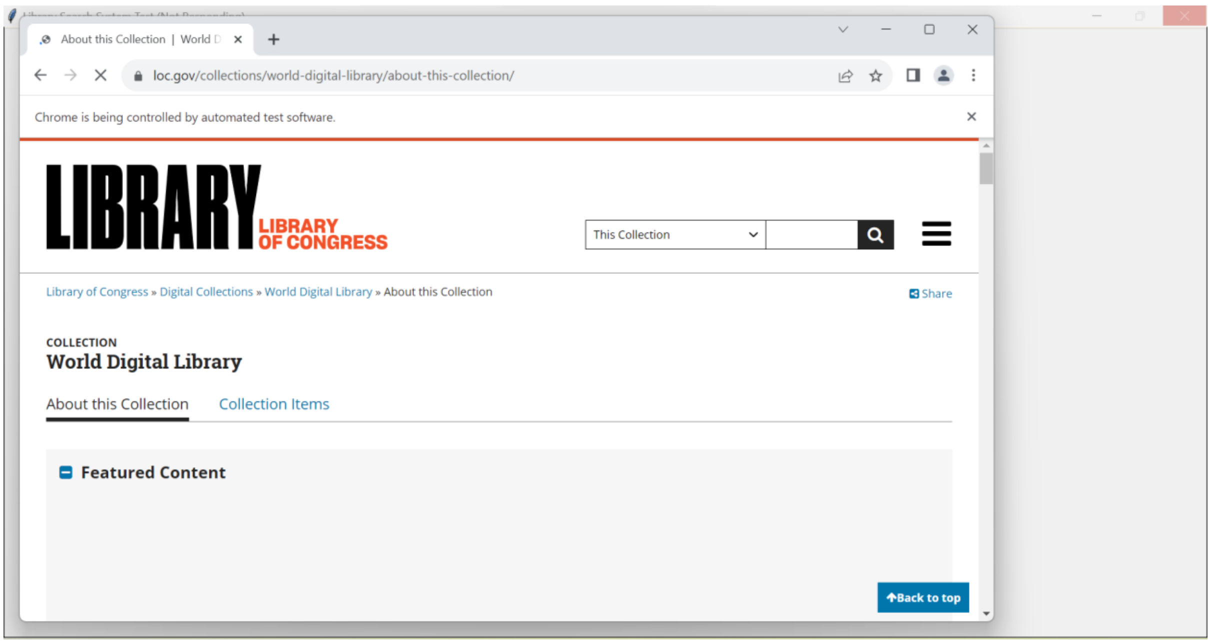
Check Partial Loading
Check Browser Response
Check Elements Rendered
Check Speed
Check ISP Speed Affects Rendering
Check Browser Responsive to Slow Connection
Test Alternative Browsers
Access With Proxies Extensions
Check Firewall Block

Rendering time on slow connection: 3.18 seconds

Test Case 5: Check ISP Speed Affects Rendering

This test case simulates a slow connection and measures the rendering time.

Expected Behaviour: The script should open a browser, set network conditions to simulate a slow connection, measure the rendering time, and display the time taken to render the website.


About this Collection | World D
loc.gov/collections/world-digital-library/about-this-collection/

Chrome is being controlled by automated test software.

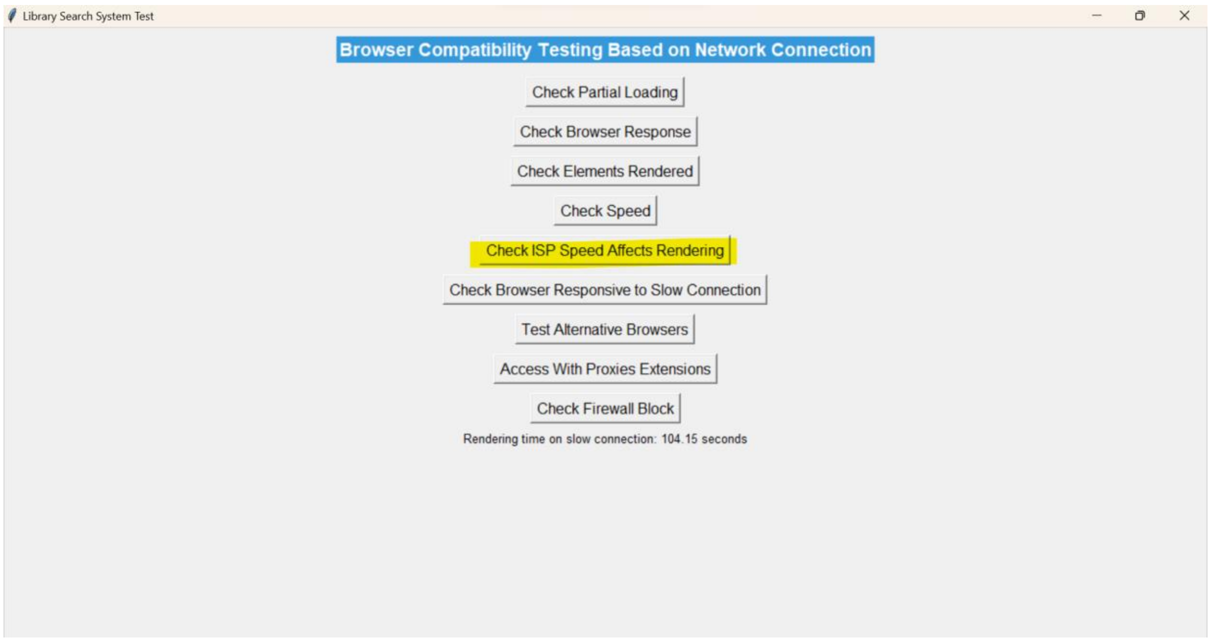
LIBRARY OF CONGRESS

Library of Congress » Digital Collections » World Digital Library » About this Collection

COLLECTION
World Digital Library

About this Collection | Collection Items

Featured Content

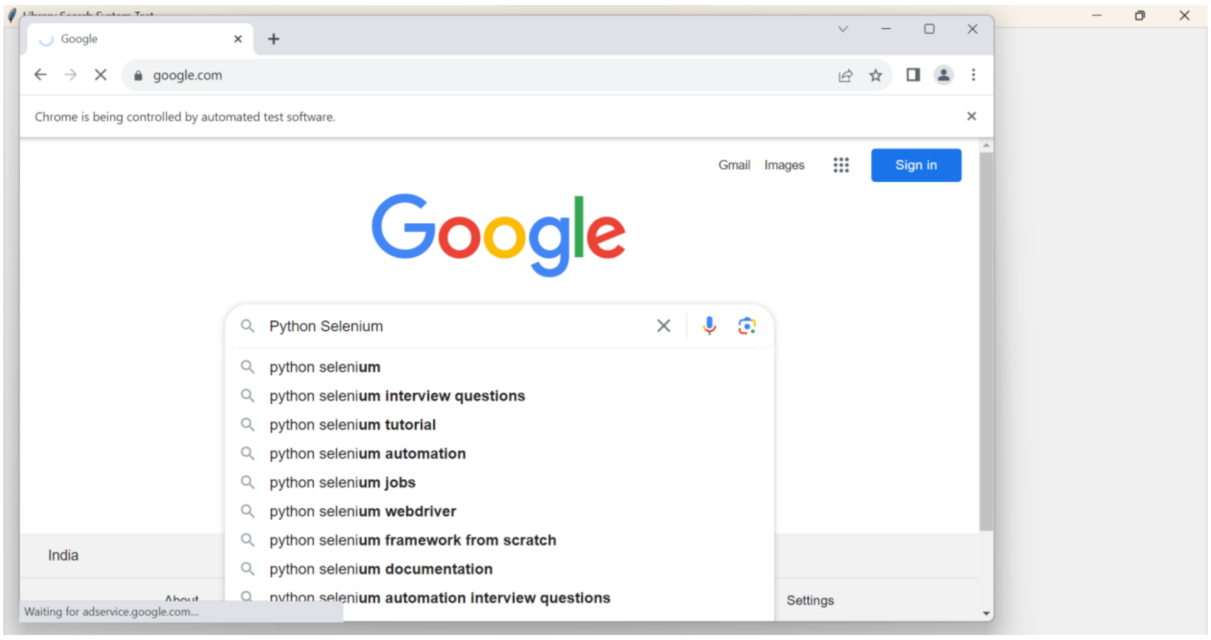

Library Search System Test
Browser Compatibility Testing Based on Network Connection
Check Partial Loading
Check Browser Response
Check Elements Rendered
Check Speed
Check ISP Speed Affects Rendering
Check Browser Responsive to Slow Connection
Test Alternative Browsers
Access With Proxies Extensions
Check Firewall Block

Rendering time on slow connection: 104.15 seconds

Test Case 6: Check Browser Responsive to Slow Connection

This test case checks if a browser is responsive to a slow connection by interacting with a search box on Google.

Expected Behaviour: The script should open a browser, set network conditions, interact with the search box, and check if the browser is responsive to a slow connection.

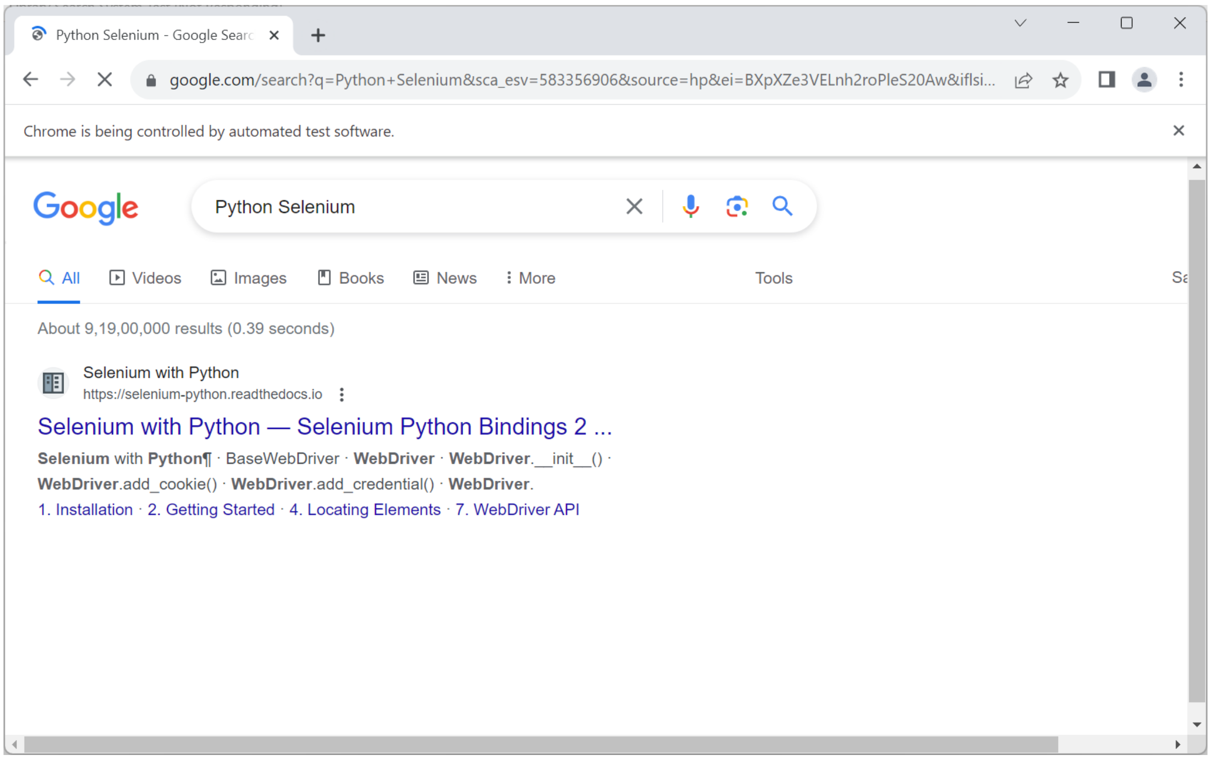


Chrome is being controlled by automated test software.

Python Selenium

python selenium
python selenium interview questions
python selenium tutorial
python selenium automation
python selenium jobs
python selenium webdriver
python selenium framework from scratch
python selenium documentation
python selenium automation interview questions

Waiting for adservice.google.com...


Python Selenium - Google Searc

google.com/search?q=Python+Selenium&sca_esv=583356906&source=hp&ei=BXpXZe3VELnh2roPleS20Aw&iflsi...

Chrome is being controlled by automated test software.

Python Selenium

All · Videos · Images · Books · News · More · Tools

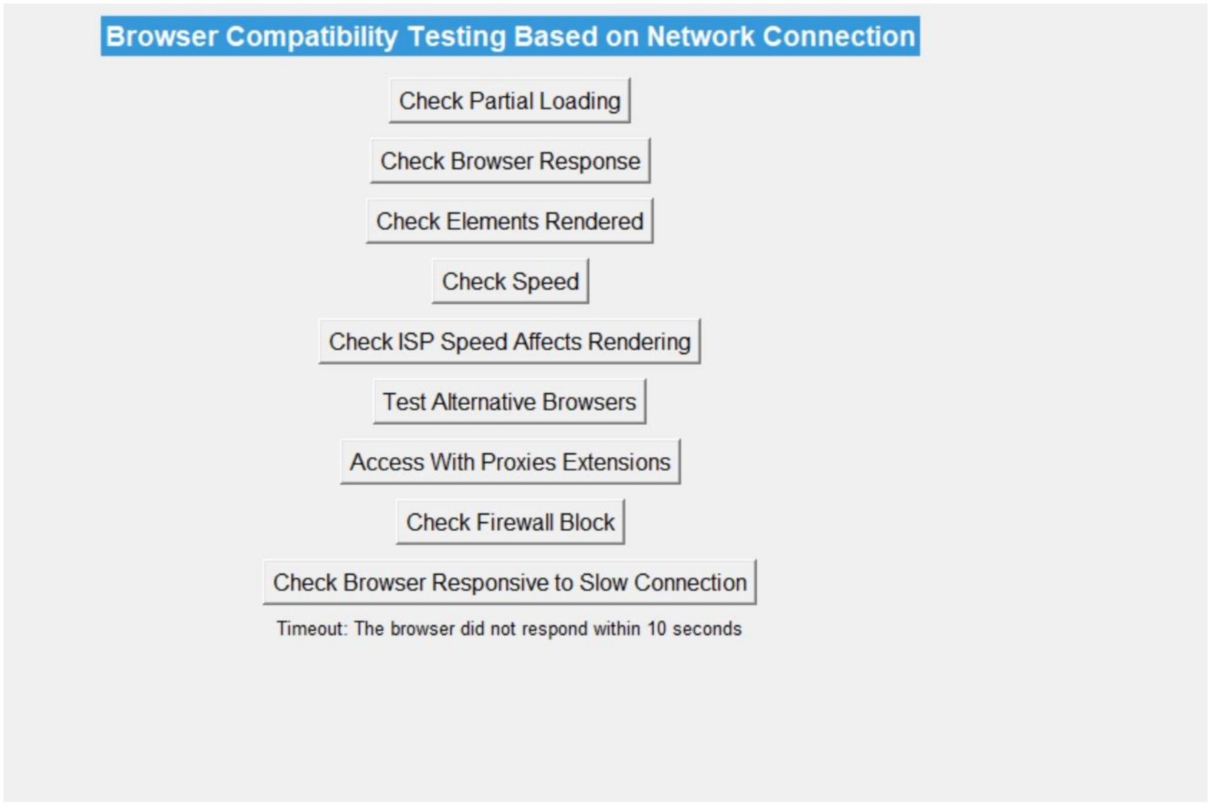
About 9,19,00,000 results (0.39 seconds)

Selenium with Python
https://selenium-python.readthedocs.io

Selenium with Python — Selenium Python Bindings 2 ...

Selenium with **Python**¶ · BaseWebDriver · **WebDriver** · **WebDriver**.__init__() · **WebDriver**.add_cookie() · **WebDriver**.add_credential() · **WebDriver**.

1. Installation · 2. Getting Started · 4. Locating Elements · 7. WebDriver API


Browser Compatibility Testing Based on Network Connection

Check Partial Loading
Check Browser Response
Check Elements Rendered
Check Speed
Check ISP Speed Affects Rendering
Test Alternative Browsers
Access With Proxies Extensions
Check Firewall Block
Check Browser Responsive to Slow Connection

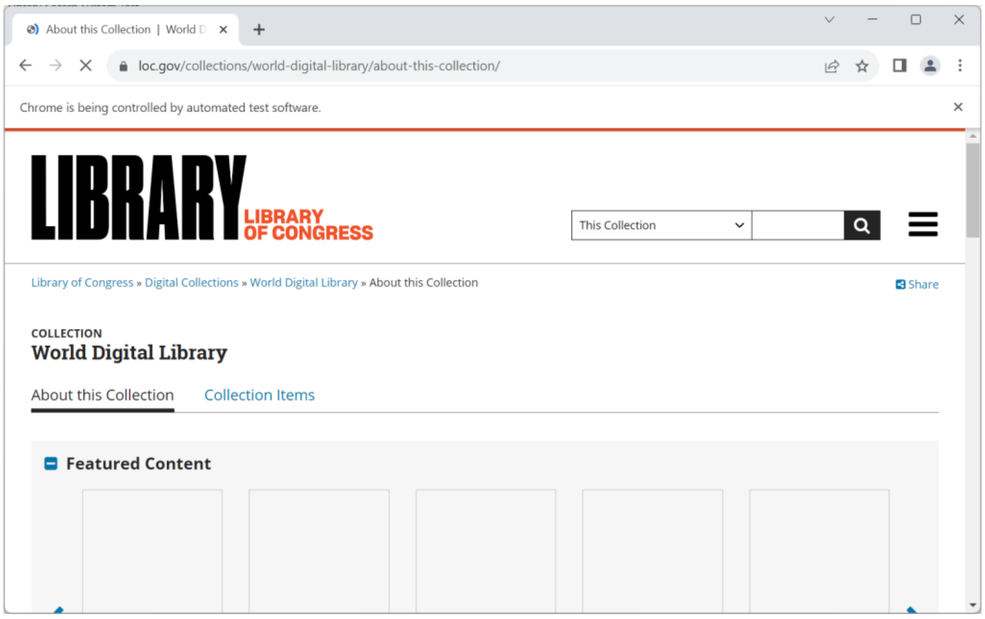
Timeout: The browser did not respond within 10 seconds

Test Case 7: Test Alternative Browsers

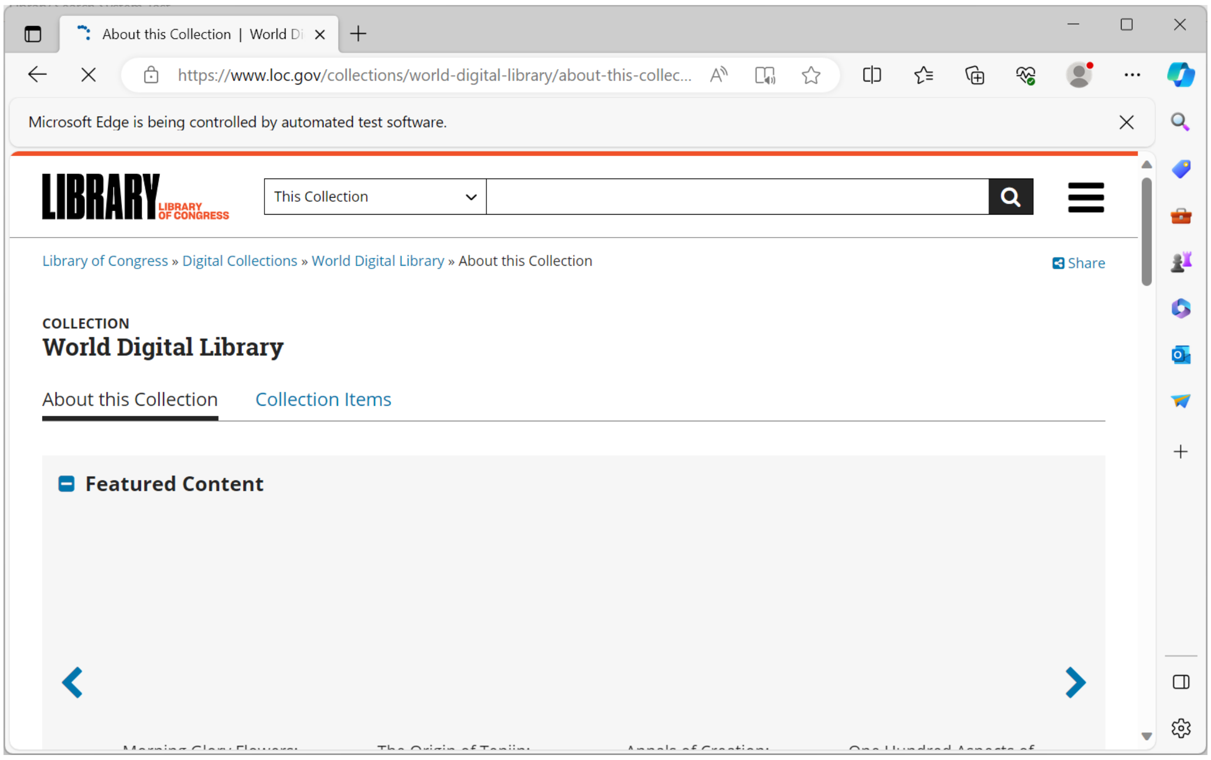
This test case tests three alternative browsers (Chrome, Edge, Firefox) to see if they can access the specified website.

Expected Behaviour: The script should attempt to open the website using each browser and display whether each browser successfully accessed the website or encountered an error.

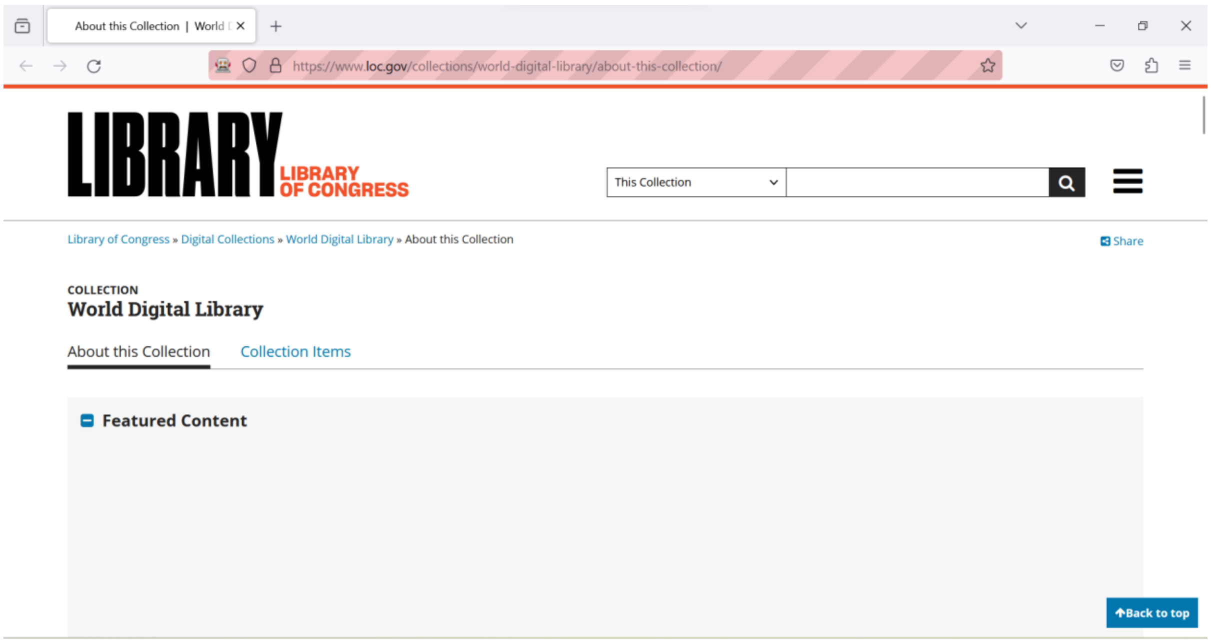
Chrome

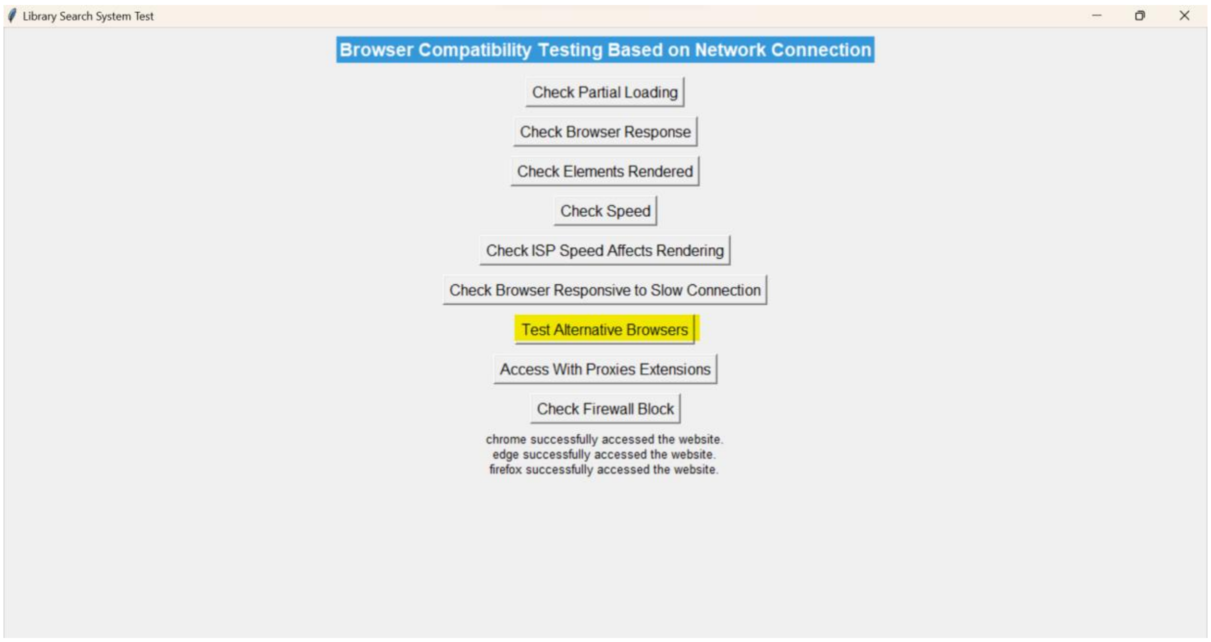

About this Collection | World D
loc.gov/collections/world-digital-library/about-this-collection/
Chrome is being controlled by automated test software.
LIBRARY
LIBRARY OF CONGRESS
This Collection
Library of Congress » Digital Collections » World Digital Library » About this Collection
Share
COLLECTION
World Digital Library
About this Collection
Collection Items
Featured Content

Edge


About this Collection | World D
https://www.loc.gov/collections/world-digital-library/about-this-collec...
Microsoft Edge is being controlled by automated test software.
LIBRARY
LIBRARY OF CONGRESS
This Collection
Library of Congress » Digital Collections » World Digital Library » About this Collection
Share
COLLECTION
World Digital Library
About this Collection
Collection Items
Featured Content

Firefox


About this Collection | World
https://www.loc.gov/collections/world-digital-library/about-this-collection/
LIBRARY
LIBRARY OF CONGRESS
This Collection
Library of Congress » Digital Collections » World Digital Library » About this Collection
Share
COLLECTION
World Digital Library
About this Collection
Collection Items
Featured Content
Back to top

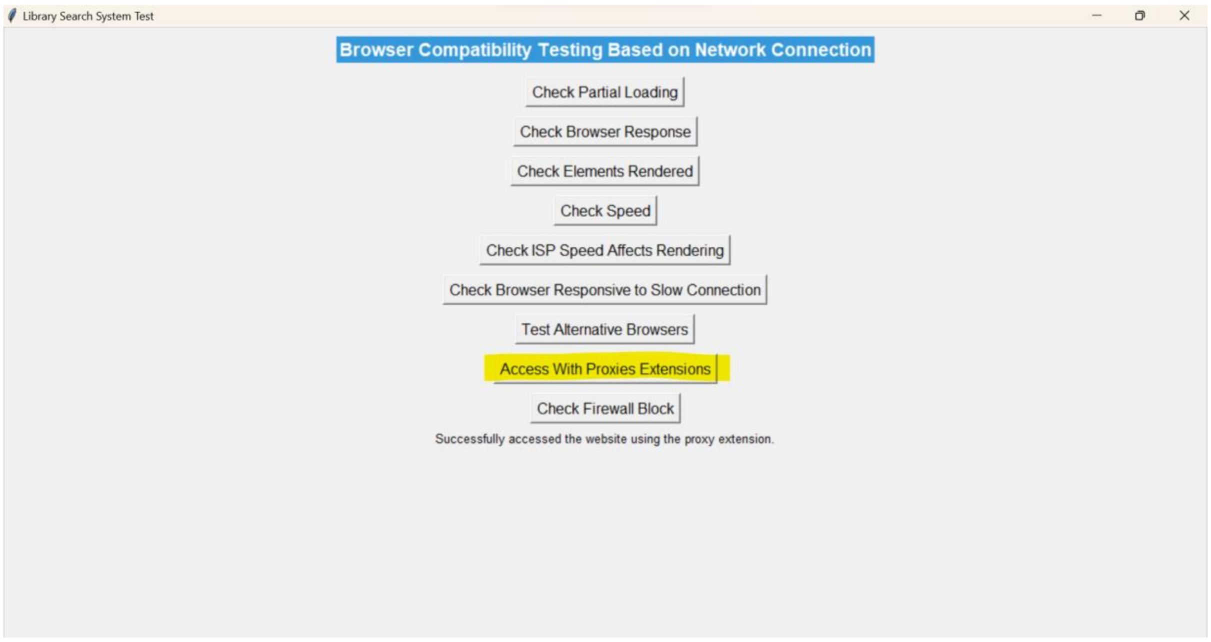

Library Search System Test
Browser Compatibility Testing Based on Network Connection
Check Partial Loading
Check Browser Response
Check Elements Rendered
Check Speed
Check ISP Speed Affects Rendering
Check Browser Responsive to Slow Connection
Test Alternative Browsers
Access With Proxies Extensions
Check Firewall Block

```
chrome successfully accessed the website.
edge successfully accessed the website.
firefox successfully accessed the website.
```

Test Case 8: Access With Proxies Extensions

This test case attempts to access the website using a Chrome extension (proxy.crx).

Expected Behaviour: The script should open a browser with the proxy extension and display whether the website is successfully accessed or if an error occurs.

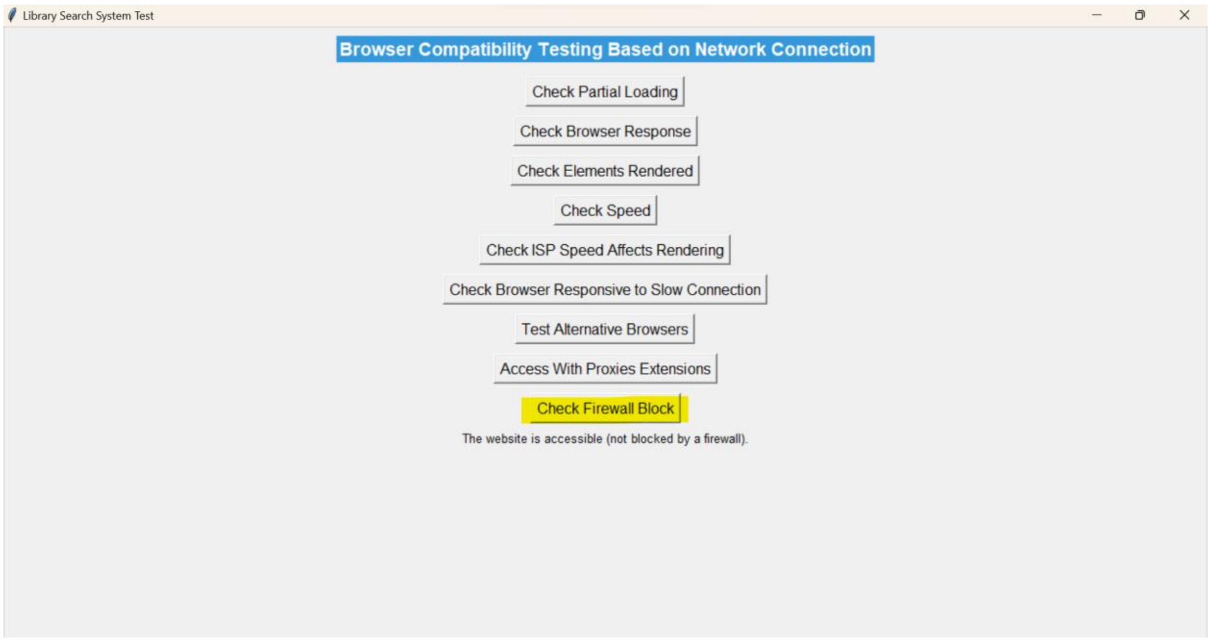

Library Search System Test
Browser Compatibility Testing Based on Network Connection
Check Partial Loading
Check Browser Response
Check Elements Rendered
Check Speed
Check ISP Speed Affects Rendering
Check Browser Responsive to Slow Connection
Test Alternative Browsers
Access With Proxies Extensions
Check Firewall Block

Successfully accessed the website using the proxy extension.

Test Case 9: Check Firewall Block

This test case checks if the website is accessible or blocked by a firewall.

Expected Behaviour: The script should send an HTTP request to the website and display whether the website is accessible or blocked.


Library Search System Test
Browser Compatibility Testing Based on Network Connection
Check Partial Loading
Check Browser Response
Check Elements Rendered
Check Speed
Check ISP Speed Affects Rendering
Check Browser Responsive to Slow Connection
Test Alternative Browsers
Access With Proxies Extensions
Check Firewall Block

The website is accessible (not blocked by a firewall).

References:

- *Selenium Documentation:*

  (https://www.selenium.dev/documentation/en/)

- *Chrome DevTools Protocol Documentation:*

  (https://chromedevtools.github.io/devtools-protocol/)

  When setting network conditions programmatically, understanding the Chrome DevTools Protocol is essential.

- *Requests Library Documentation:*

  (https://docs.python-requests.org/en/latest/)

  For making HTTP requests in the check_firewall_block test case.

- GitHub repository of the project https://github.com/Shreelekha-142/Browser-Compatibility-Testing

Peer Evaluation

Name	**Evaluation**
Meghana N	5
Mridvi Sharma	5
Mullapudi Jahnavi	5
N Shreelekha	5