PROJECT: WEB ACCESSIBILITY TESTING
AUTOMATED WEB ACCESSIBILITY TEST PLAN

Web accessibility: simply put, it is making all available web content usable to persons with disabilities and making sure newly created content adheres to standardized, internationally accepted guidelines. Current guidelines are outlined in the Web Content Accessibility Guidelines (WCAG 2.0) and Section 508 of U.S. federal government legislation.

Disabilities range from inability to use a keyboard, mouse (due to injury, advanced age) to impaired vision, colorblindness, dyslexia, epilepsy and others. Web accessibility to disabled users is enabled with the use of assistive technologies (e.g. screen readers, Braille keyboards, etc.), properly designed and created websites, the combination of the two and testing.

ACCESSIBILITY TESTING: testing of websites' content so that it is accessible and usable by anyone, including and most importantly people with disabilities. Websites must be designed and created in a manner that all content within can be used, understood, accessed and disseminated by anyone.
Popular accessibility problems include but are not limited to: websites not navigable with the sole use of a keyboard. Images without properly descriptive content in alt="image" attribute within HTML, videos without captions, text that is too small, unfavorable color use and combinations, etc.

CURRENT SITUATION: There is currently no testing or web accessibility tool being used by Mozilla. There are a variety of manual testing methods and automated testing tools for general use online that target different criteria with varying degrees of accuracy.

FOCUS & GOAL: Automated accessibility testing that will provide the most comprehensive results, covering most wide ranging criteria with the use of available open source tools and frameworks that will ultimately run within Jenkins CI and minimizes false positives. Produce an automated testing tool engineered with the use of open source APIs and deployed for use.

TESTING TOOLS:

> AATT (Automated Accessibility Testing Tool) - tests within guidelines in WCAG 1.0, WCAG 2.0 and Section 508. Test single or multiple pages and restricted/password protected pages.
>
> AXE - tests HTML based apps using Javascript. Also checks single, multiple and pages restricted/password protected. Tests within WCAG 2.0 and Section 508. Claims to produce zero false positives.
>
> Pa11y/CI - tests within WCAG 2.0 and Section 508 guidelines, runs HTML CodeSniffer and can run tests agains multiple pages and report issues.
>
> CynthiaSays - tests within WCAG 2.0, Section 508 and U.S. federal procurement standards. Checks single web pages and supports CSS, HTML and images. Available as free software.

ENGINEER/CODE: write code that employs one of the testing tools mentioned above or possibly if time allows use all as a way to test for best results. Axe would be a good starting point

since it claims to have zero false positives which is one of the main objectives of the project. Axe seems to have all the features provided by the other tools and it is designed to work with whatever tools, frameworks, libraries and environments currently being used.

API will be integrated within Jenkins and deployed.

"TEST EARLY, TEST OFTEN" and analyze results to determine if they're comprehensive enough for our purposes and the accessibility standard chosen.