An Automated Website Tester

WebEvaluator

Team

Hack_Feast

Theme

Open-Innovation

Team Members

Jai Gupta

Aman Dwivedi

Harsh Mishra

Yash Vardhan

Abstract

The idea is to create a tool that can take a website URL and check for its compatibility on various parameters such that all the information must be accessible to those persons with disabilities (following ADA compliances), checking the security of the website to make sure that it has headers to protect it from client-side attacks, SSL checks for making sure that users sensitive data that is being sent between two systems is private and secure, checking cookies set by the websites and their allies that track the visitors. Here, make sure that the website asks for the user's consent before placing cookies on the browser and stop tracking if the user denies the permission.

Problem Statement

Creating a tool that can crawl the sites on some given parameters to identify any adherence/compliance issues that might exist and report back with suggestions to fix them.

Building an advanced web crawling that will not only discover the active URLs within the website but also provide information about SSL certificate compliance, Cookie checker and ADA compliance and details about the security headers.

- **SSL Certificate Compliance:** Checking all the links within the website for URL validation of SSL, all URLs should direct to HTTPS and verify the validity of the SSL certificate.
- **ADA Compliance:** Checking the website for Alt text in all the images, Color contrast for all the sites as per w3.org guidelines, accessibility issues to check the site markup for null tab index, and check for all HTML, CSS, JavaScript-based errors.
- **Cookies Checkers:** Verifying cookies being used by the website, according to General Data Protection Regulation. The cookie checker will scan for cookies on the website and cookies consent verification links**.**
- **Security Headers:** Checking the headers present on the web applications to analyze security defenses in web browsers and prevent client-side attacks.

Solution/Features proposed in the Tool

Some of the main objectives of the tool are to crawl the complete website, verify SSL certificate compliance, check for the cookies available and their properties, check for ADA compliance under various guidelines, and export the entire result on the website & as pdf:

- **Crawler:** A super fast crawler is created in golang using the colly framework. The crawler is configured with many advanced features like - request delays, maximum concurrency, parallel scraping, Robots.txt support, specifying custom parameters for different results including depth.
  The complete list of URLs found would be subdivided based on -
  - Active
  - Inactive/Broken URLs
  - Domain-wise classification
  - HTTP and HTTPS
- **SSL Certificate Compliance:** For collecting the SSL and TLS information from the host we are using a script to get the information form the host and return the information of the identity and signature in JSON format. It also checks for an encrypted connection.
  Some of the features of the script are -
  - Checks the validity of the certificate
  - Checks the issuer of the certificate
  - Analyze the certificate for security issue
  - Checks for some well-known vulnerabilities
- **Cookie Analysis:** According to general data protection regulation, cookie compliances include receiving user's consent before you use any cookie except for necessary ones, providing accurate data each cookie tracks, document and store consent received from users, make it easy to withdraw the consent and others.
  Our approach to this is -
  - Create a list of all cookies along with all its details
  - Categorize the cookies into different categories
  - Searching for all the URLs if website has any cookie disclaimer
  - Cookie consent checked by clearing cookies and reloading page

- **Ada Compliance:** ADA compliance means that all electronic information and technology must be accessible to those with disabilities. It clearly issues guidelines for accessibility including alternatives, presentations, user-control, understandability and predictability of the website to disabled ones.
  Our approach to checking the Ada compliance -
  - A microservices is created in node js which receives a list of URLs from the main golang backend and runs compliance scans.
  - It checks the HTML code by checking its structural and logical integrity by looking at possible syntax and security concerns
  - The CSS errors like color contrast, blindness, blurred vision look of the website is achieved using Checka11y framework
  - Javascript errors would be reported as errors and warnings on the console, syntactically invalid code and other runtime and logical errors.
  - Implying on the standard WCAG levels viz. [A], [AA], [AAA] compliant errors and exporting them using Tenon API.
  - Using HTML Sniffer to report miscellaneous compliant warnings

- **Rendering and Exporting of Results:** After the entire crawling and intermediate results are fetched, we give options to export the complete report to the user on screen with tables and modals and also generate the report as a pdf file. We have used EJS and javascript to achieve this.
  Brief details around this functionality are -
  - As it is not viable to render all errors on the site itself, we give options to users to export all the data as pdf and review better.
  - We export the detailed analysis of each and every sub-part of the tool as tables wherever necessary
  - All the errors are also classified as warnings, errors and notices so that user can review them on priority basis.

Tech-Stacks to be used

The project is using a microstructure architecture where we have the main server in golang which is communicating with all other agents/scripts such as Python, Node.js

Agent name/ Feature	Tech Stack	Description
Crawler	Golang colly framework	Using colly framework of Golang as it is one of the fastest available crawler
SSL certificate compliance	Golang	Golang script that searches for the SSL information
Cookie checker	Node.js and JavaScript	Using Puppeteer for automated cookie consent verification
ADA compliance	Node.js and JavaScript	Using a variety of node libraries for getting complete ADA compliance information
Security Headers	Python	Built a Flask API that checks for the headers in the HTTP request

Future Scopes and Improvements

We can always improve the existing project in terms of display, functionalities and performance -

- There is a scope of adding extensive animation to the user interface, highlighting each issue with separate colors to distinguish among.
- Although, almost all of the functionality and script is implemented from scratch, we can minimize the use of all APIs, even the basic ones like the SSL API from the official SSL Labs, by writing custom scripts for them
- Making the project more reusable and avoiding stiff coding
- Creating a chrome extension using the project and publishing it to the web store
- Extending the tool to cover functionalities like SEO and security scans for a website as well.