**Testing Automation Tool V1**

**Overview:**

Create a tool that helps us in automation the following process.

* Ability to extract all the URLs from a sitemap page and add them to a stack or database.
* Ability to perform a check on the extracted URLs for the availability of correct DTM Library.
* Ability to perform a user journey (automating web browsers) and catch the network calls request and response.
* Ability to run a javascript during the web automation scenario.

**Technical Requirements:**

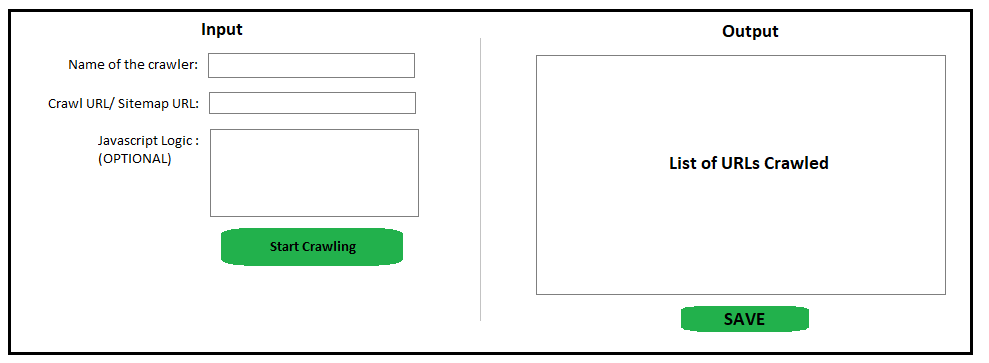
* Selenium – For automating user journey.
* Java – For writing backend logic of storage, access, and data delivery.
* MySQL – For storing the test results.
* Javascript – For writing frontend logic and DOM manipulation.
* REST APIs – For communication between backend and frontend ensuring secured data delivery.
* HTML and CSS – For writing frontend architecture and design.
* Charting library (Google Charts/ChartJS/D3JS etc) – For creating visualization on frontend.

**Major Pages:**

* Crawler – For extracting the URLs from given page.
* Scanner – For performing a check on the provided URLs
* Automator (Scenario) – For writing and steps to be performed by the browser.
* Tests – For declaring the conditions and checks to be performed on Scanner and Automator.
* Dashboard – For displaying the visualization and performance of the tests.

**Process:**

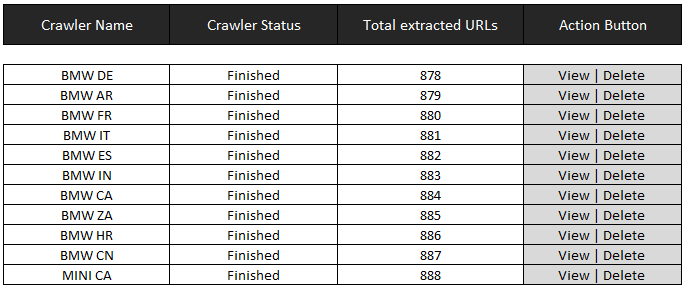
Crawler Main View:



1. Name the crawler eg. BMW DE Sitemap URLs.
2. Provide the Sitemap URL for BMW DE. Eg. [*https://www.bmw.de/de/footer/sitemap.html*](https://www.bmw.de/de/footer/sitemap.html)
3. Write a javascript login if required on page to extract the URLs using unique classname or identifier.
4. Expect the outputs on the right side text box in next line format.
5. Facility to save all the crawled URLs.

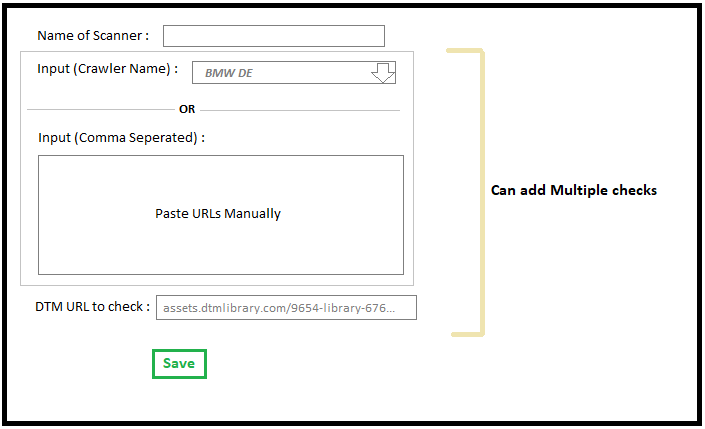
Crawler Index Page:

This page will store the list of all the crawlers initiated by the user.



* **Crawler Name**: Unique name for every crawler must be provided to be used in scanner.
* **Crawler Status**: Finished stating the URLs crawl complete. Pending stating the crawler did not ran.
* **Total Extracted URLs**: Totals URLs extracted by the crawler.
* **Action Buttons**: View to go to crawler home view and delete to delete the crawler.

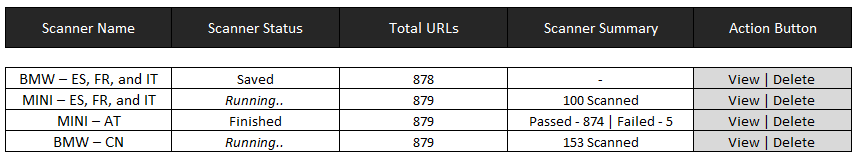
Scanner Main View:



1. Enter a scanner Name. Eg. (BMW – ES, FR, and IT)
2. Either provide the crawler name which crawler for the URL of either market or paste the URLs manually.
3. Provide a DTM Script to be checked on URLs provided.
4. Repeat step 2 and 3 for all 3 markets.
5. Save the scanner and run for the check.
6. Additionally, browser option can be added.

Scanner Index View:

This page stores all the scanner which are either running, finished or saved.



* **Scanner Name** – Unique name of the scanner.
* **Status** – Saved is for configuration settings saved. Running is the scanner is running. Finished is when the scan results are available.
* **Total URLs** – Total URLs provided by the user.
* **Scanner Summary** – Live status of the scanner.
* **Action Buttons** – View the scanner or delete the scanner.

Scanner Result View:

