# SEO Technical Tool READ\_ME:

## Objective:

This tool will take in one or multiple Macy’s URLs, Audit the page for technical SEO Factors, and score the page to give the users a pass/fail grade. The target audience would be Product Managers as well as Development teams to ensure every new project would be SEO friendly; requiring a minimum score in order to reach the production site.

(Optional Feature): after running, the tool would send a report back to the SEO team.

## Prerequisites:

The user has a LIVE Macys.com page to test. This page is going to be indexed on Google and is Customer Facing.

## SEO Scoring Factors:

|  |  |
| --- | --- |
| **Feature** | **Score** |
| Client Side vs. Server side | Pass/Fail |
| HTTPS Status = 200 or 301 | Pass/Fail |
| No Index/No Follow | Pass/Fail |
| Canonical Tag +Rel Alt Tag | 25% |
| Meta Data: Title & Description | 25% |
| IMG. Alternate Text | 10% |
| URL Parameters | 25% |
| Responsive | 15% |

* **Client side Vs Server Side:** The SEO team would like to see all pages be more server side rendering than Client Side. This tool will scan the page’s code to find elements off each, the weigh the comparison. To PASS the page must have more Server Side elements than Client Side.
* **HTTPS Status**: The SEO team would like to see all pages HTTPS Response code as either a 200 OK or a 301 Redirect (This should throw a flag). The test will FAIL if a code other than the 2 previously stated.
* **NoIndex/NoFollow:** The SEO team wants all production pages to be index by Google (Assuming the Prerequisites are met. \*\*Excludes special cases) therefore we should NOT see the NoIndex/NoFollow Tag in the HTML source code of the page.
* **Canonical Tag + Rel. Alternate Tag:** The SEO Team would like to see all pages have a Canonical & Alternate Tag for each page.
* **Meta Data:** The SEO Team would like to see each page have corresponding Title Tags & Meta Descriptions.
* Image Alternate Tags: The SEO team would like to see each image having its own alternate text.
* **URL Parameters:** The SEO team would like to see each page contain the following parameters in the URL:
  + SSL Tag (HTTPS://)
  + www.
  + Macys.com Domain
  + (Optional): IMP Parameter /social
  + (Optional): CE Parameter /ce
* **Responsive:** In future iterations of Macys.com, the site will be moving to a responsive design. To help accommodate for this change the SEO Team would like this tool to check for responsive design elements.

#### Scoring:

The tool will first test the page for the factors above starting with the first 3 PASS/FAIL tests. If any of those tests FAIL the page will be automatically graded with an F and failed.

If the page passes the PASS/FAIL Requirements, the tool will test for the remaining SEO factors. The page must get at least a 76% (C) in order to be passed.

#### Grading Scale:

**Percentage Grade**

100 - 91% A

90% - 81% B

81% - 76% C

75% - 0% F

## Current Code:

**Current Modules:**

from lxml import html

import requests

import csv

from bs4 import BeautifulSoup

**Definitions:**

def clientside():

This function will take in the URL; search the HTML for two snippets of code which are rendered on Server side & Client side. The resulting score is applied for pass or fail.

def httpStatus():

This function checks for the HTTP Status Code. \*\* Have not found a way to determine if 200 or 301 to score.

def noIndex():

This function checks for the NoIndex/NoFollow Tag

def finalScore():

This function applies the Pass/Fail score & tells the test whether to continue or fail the page.

def urlSyntax():

This function takes the inputted URL to scans for Macys.com parameters.

def tags():

This function parses through the page’s HTML to find the Canonical and Alternate Tags.

def metaDesc():

This function will parse through the HTML to find & scrape the Meta Description.

def metaTitle():

This function will parse through the HTML to find & scrape the Title Tag.

\*\*Having trouble grabbing just the Title Tag text without the HTML tags. <title> </title>

def responsive():

This function will parse through the page HTML to find the viewport responsive code.

\*\*Having an issue with which code is accepted, as there are multiple cases.

## Future Enhancements:

* Having the ability for the tool to send the SEO Team a report when the tool has been ran.
* A User friendly GUI.
* Adding more SEO feature tests.
* Mobile support.
* Ability to analyze and score localhost projects/code without a valid URL

## Roadblocks:

* Scoring system is no working properly/Needs to be updated to include the percentage tests.
* Finding the correct http response/redirect chain.
* Scrapping the Title Tag for text only.
* Finding a true source of truth for client side vs server side. (Thought about doing a ratio of Javascript that is on page).
* Hosting the tool on heroku, still needs a GUI.

Links:

GitHub Repository: <https://github.com/SeanCarey91/Project_ST-ST>

Heroku: <https://dashboard.heroku.com/apps/project-st-st/deploy/github> - (May need my sign in credentials.)

Multiple URL Input Python Code:

c = csv.writer(open("”Name your file here”.csv", "wb"))

with open('slp\_url.csv', 'rb') as f:

reader = csv.reader(f)

for row in reader:

c.writerow([ “Enter Variables Here” ])