|  |  |
| --- | --- |
|  | Web Scraper |
|  | Minneapolis College |
|  | Dominic Guye  ITEC 1150  4/22/25 |

Website that scrapes other websites. It will programmatically scan HTML documents for particular elements and generate out in accordance with the results. Useful for passively monitoring websites for updates. Great for testing to compare a browser’s display of an unfamiliar webpage to what the scraper actually found.

User story statements

* As a security engineer, I want to use an external server to scrape websites for particular content so that I can see whether the website is actually safe or not without actually exposing my computer to risk by directly visiting the website.
* As a search engine optimizer, I want to be able to scan websites for keywords and HTML elements from an independent source so that I can determine whether the edits I made to my website are functioning on a basic level, as I debug why optimizations I made are not yielding the results I expected.
* As a browser developer, I want to check a website for a set of given webpage elements so that I can know which web elements are the cause of unexplained performance issues in my browser
* As a web developer, I want to have a non-GUI means of viewing webpages so that I can quickly know what HTML elements are working without dealing with the performance overhead of a full webpage viewer or browser.
* [I don’t really have a sixth, sorry]

Requirements:

1. I intend on using the PyCharm IDE to create this program, and I intend on fully annotating my code with comments.
2. I will use variables. Strings will be used extensively as the data type for raw HTML content.
3. (I don’t know yet)
4. If-Elif statements are used to check for null content, which is an unacceptable response that will registered as an error and handled accordingly.
5. (I don’t know yet, but probably) Will use For loops to perform operations on the same types of HTML elements
6. I will use lists as the output when I search a document for more than one instance of the same type of element.
7. The commands to send an HTTP request will be its own function, and the HTML parsing will be the action of its own function

Milestones:

* Successfully make an HTTP request
* Save the HTML into a variable that can be parsed
* Request a particular HTML element with Beautiful Soup

# References

**There are no sources in the current document.**