**Learning Outcome HTMX vs JavaScript –**

**Christian Rock**

**Software Engineering II**

**02/04/2025**

**Project Used:** Server-Side Web Final Project – CS:GO Skins API File

**Important Note**: HTMX is designed to handle HTML responses from API not Json.

**Using HTMX to fetch and display API data dynamically, we achieved the following outcomes**:

**Simplified Frontend**

Eliminated the need for JavaScript's fetch() or XMLHttpRequest() to handle API calls.

Reduces the complexity of handling asynchronous API requests manually.

**API Requests**

Using hx-get="http://localhost:80/skins/64", we sent an HTTP GET request directly from an HTML element.

The response was received from the API and displayed on the web page without the use of any JS.

No need to write JavaScript event listeners or AJAX logic. This saves a ton of time.

**Seamless DOM Updates**

The hx-target="#weapon-details" attribute ensured that the response was injected into the correct part of the page. In this case it was injected into a div with the ID = weapon-details

hx-swap="innerHTML" replaced the content dynamically, avoiding the need to manually manipulate the DOM. This avoids having to reload as well.

**Better Performance and Maintainability**

HTMX reduces the amount of JavaScript code, leading to faster page loads and easier debugging.

Code remains clean and readable, as behavior is controlled via attributes in HTML rather than plain JavaScript.

**Simplified Enhancement**

HTMX enhances functionality while maintaining simplicity and maintainability.

This HTMX library allows developers to quickly integrate API-driven interactions with less effort, making it a great choice for our project where some members may not be familiar with front-end frameworks such as React or Vue. Below is an example of how much HTMX shortens API calls.

**Agile Principle #10 Simplicity – The motivating factor behind learning HTMX.**

**JavaScript AJAX Snippet:**

In JavaScript File:

document.getElementById("loadWeapon").addEventListener("click", function() {

fetch("http://localhost:80/skins/64")

.then(response => response.json())

.then(data => {

document.getElementById("weapon-details").innerHTML = data;

})

.catch(error => console.error("Error fetching data:", error));

});

In HTML File:

<div id="weapon-details" class="weapon-container">

<!—API GET Response will be injected here -->

</div>

**HTMX Snippet:**

ONLY HTML FILE:

<button hx-get="http://localhost:80/skins/" hx-target="#weapon-details" hx-swap="innerHTML">

Load Weapon Details

</button>

<div id="weapon-details" class="weapon-container">

<!—API GET Response will be injected here -->

</div>

**SOURCES**

[**https://htmx.org/docs/**](https://htmx.org/docs/)

**GitHub Example**:

**https://github.com/rockcm/HTMX\_Example**