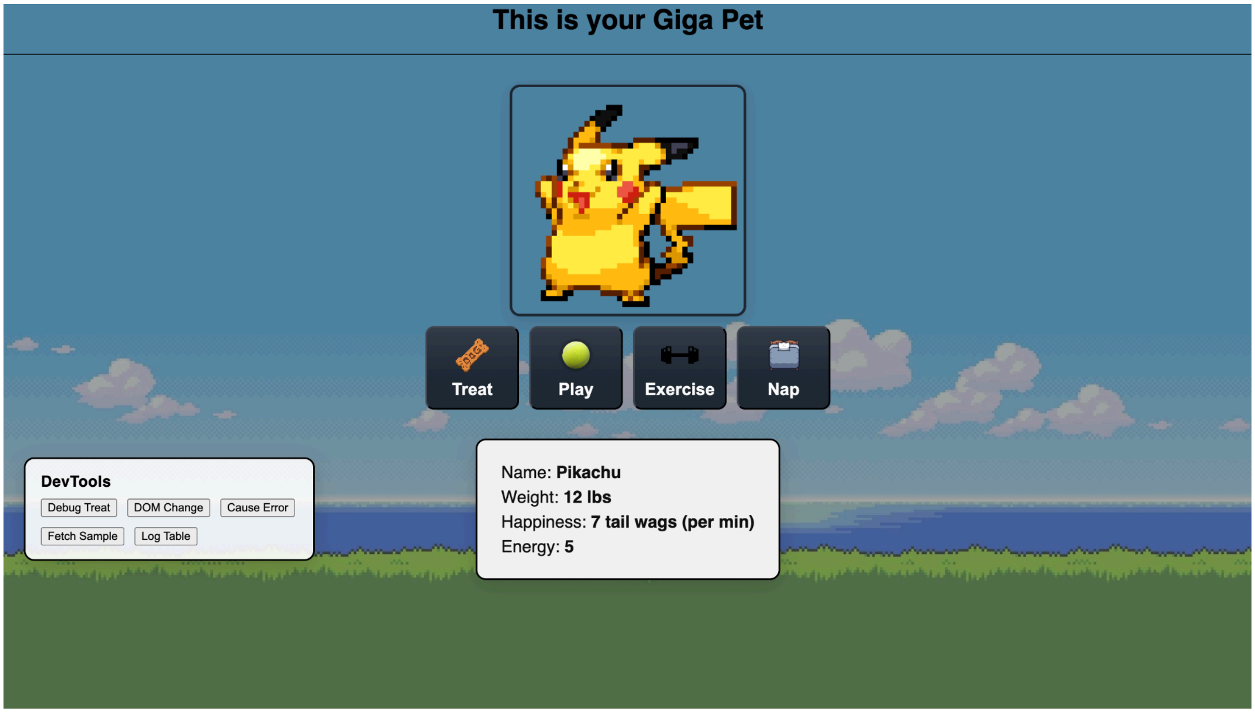
Vincent Watchinda
COMP 484
12/2/2025

COMP484-HW 10

IMPLEMENTATION

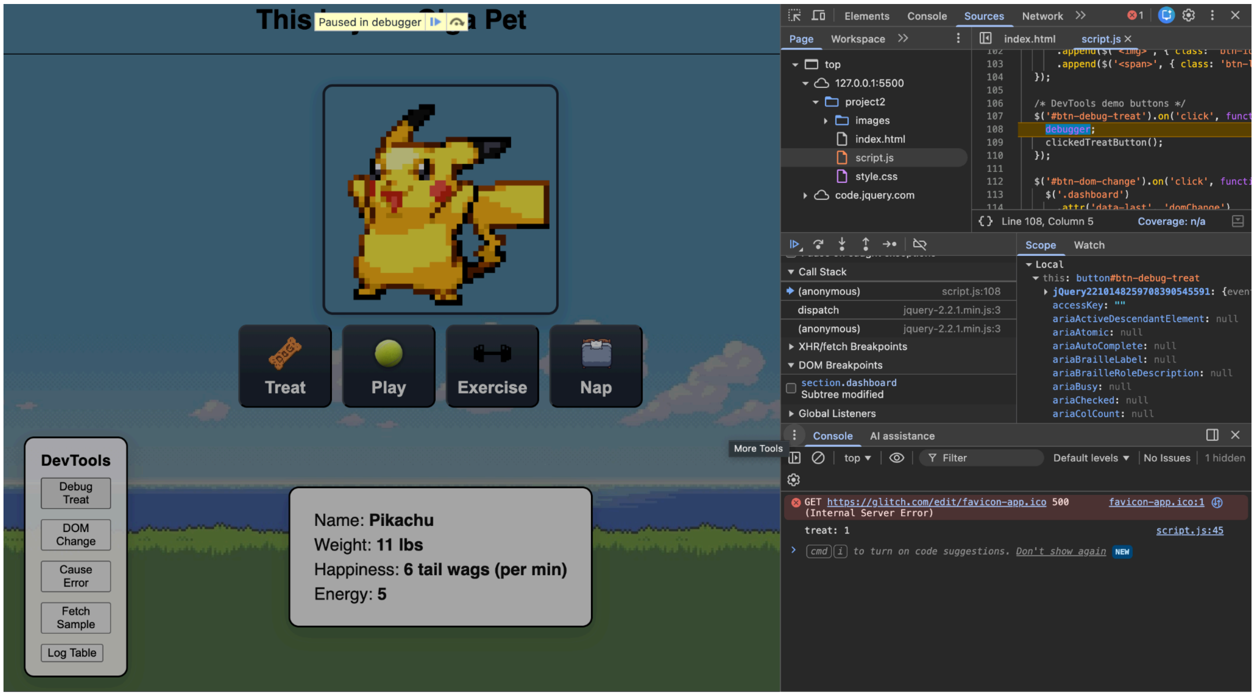
I added the DevTools chart on the bottom left which includes 5 buttons that can see its use cases in the DevTool on Google. In this doc, I will go through what each button uses.



Name: **Pikachu**
Weight: **12 lbs**
Happiness: **7 tail wags (per min)**
Energy: **5**

DEBUG TREAT BUTTON

In sources, you will see that “debugger;” pauses the execution of the code at that point. In this case, it stops before “clickedTreatButtin();” so none of the pet stats are actually updated at this point. Once you resume the execution, you’ll see that treat will impact the pet stats per usual.



Name: **Pikachu**
Weight: **11 lbs**
Happiness: **6 tail wags (per min)**
Energy: **5**

```
      .append($('<span>', { class: 'btn-
  });

  /* DevTools demo buttons */
  $('#btn-debug-treat').on('click', funct
    debugger;
    clickedTreatButton();
  });

  $('#btn-dom-change').on('click', functi
    $('.dashboard')
```

Line 108, Column 5   Coverage: n/a

```
(anonymous)    script.js:108
dispatch       jquery-2.2.1.min.js:3
(anonymous)    jquery-2.2.1.min.js:3
```

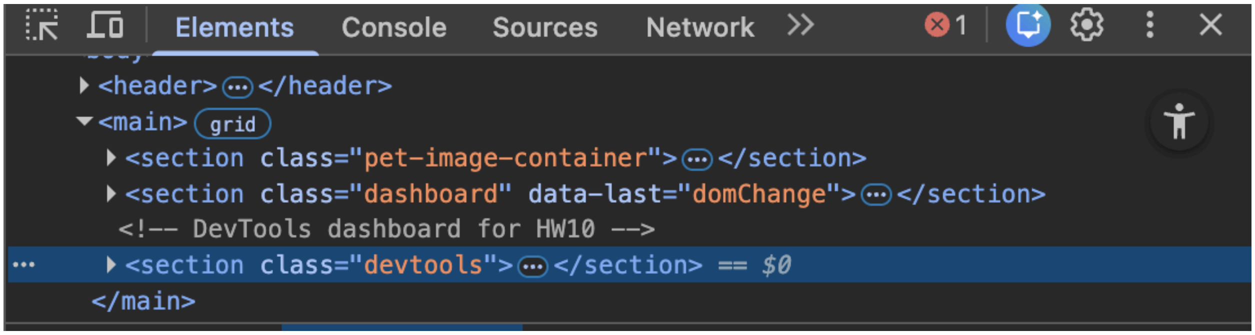
section.dashboard
Subtree modified

```
Local
  this: button#btn-debug-treat
    jQuery22101482597083905455591: {even
    accessKey: ""
    ariaActiveDescendantElement: null
    ariaAtomic: null
    ariaAutoComplete: null
    ariaBrailleLabel: null
    ariaBrailleRoleDescription: null
    ariaBusy: null
    ariaChecked: null
    ariaColCount: null
```

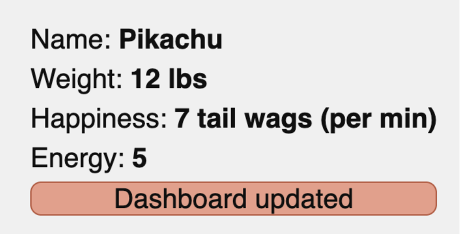
```
GET https://glitch.com/edit/favicon-app.ico 500        favicon-app.ico:1
(Internal Server Error)
treat: 1                                               script.js:45
> cmd i to turn on code suggestions. Don't show again NEW
```

DOM CHANGE BUTTON

In Elements, you can watch the page structure update in real time. When you click this, a small "Dashboard updated" note is added to the dashboard and a data attribute changes. If you right-click the dashboard in Elements and choose "Break on → Subtree modifications," DevTools will pause exactly where that DOM change happens. This shows how JavaScript can change the live page without touching the original HTML file.



```
  <header>…</header>
  <main> grid
    <section class="pet-image-container">…</section>
    <section class="dashboard" data-last="domChange">…</section>
    <!-- DevTools dashboard for HW10 -->
    <section class="devtools">…</section> == $0
  </main>
```

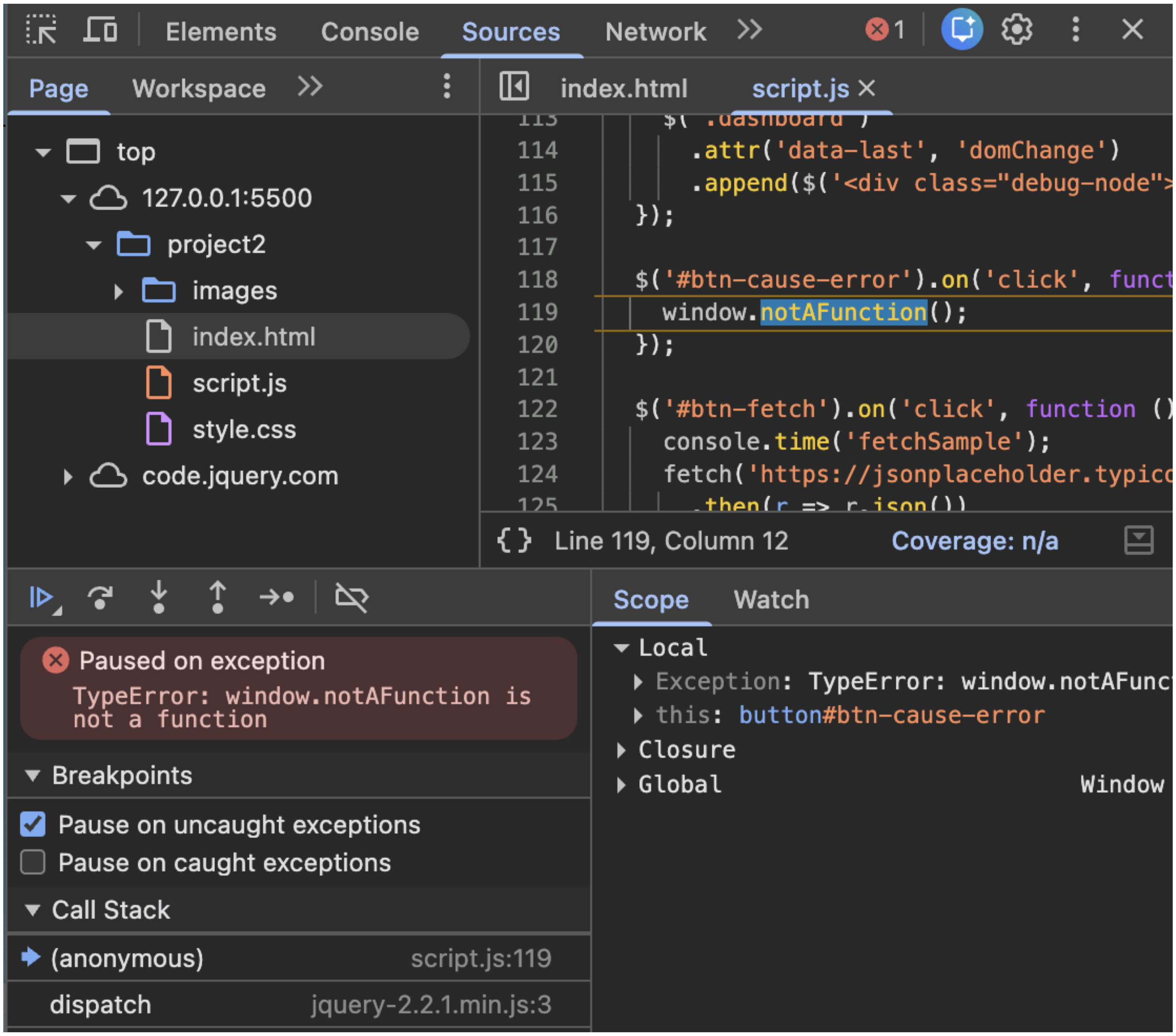


Name: **Pikachu**
Weight: **12 lbs**
Happiness: **7 tail wags (per min)**
Energy: **5**
Dashboard updated

The DOM is changed here with the addition of class="devtools" but the html itself was not changed.

CAUSE ERROR BUTTON

This purposely throws an error. If you turn on “Pause on uncaught exceptions” in Sources, clicking this will freeze the code on the exact line that failed. You can see the error message in the Console. This is helpful to learn how to track down bugs and speeds up the process overall. The console output is below.



```
      $('.dashboard')
        .attr('data-last', 'domChange')
        .append($('<div class="debug-node">
  });

  $('#btn-cause-error').on('click', funct
    window.notAFunction();
  });

  $('#btn-fetch').on('click', function ()
    console.time('fetchSample');
    fetch('https://jsonplaceholder.typico
      .then(r => r.json())
```

Paused on exception
TypeError: window.notAFunction is not a function

Breakpoints

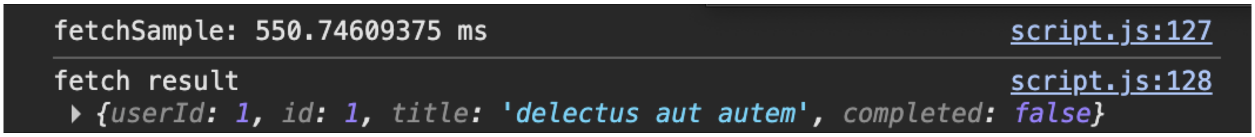
☑ Pause on uncaught exceptions

☐ Pause on caught exceptions

Call Stack

```
(anonymous)        script.js:119
dispatch           jquery-2.2.1.min.js:3
```

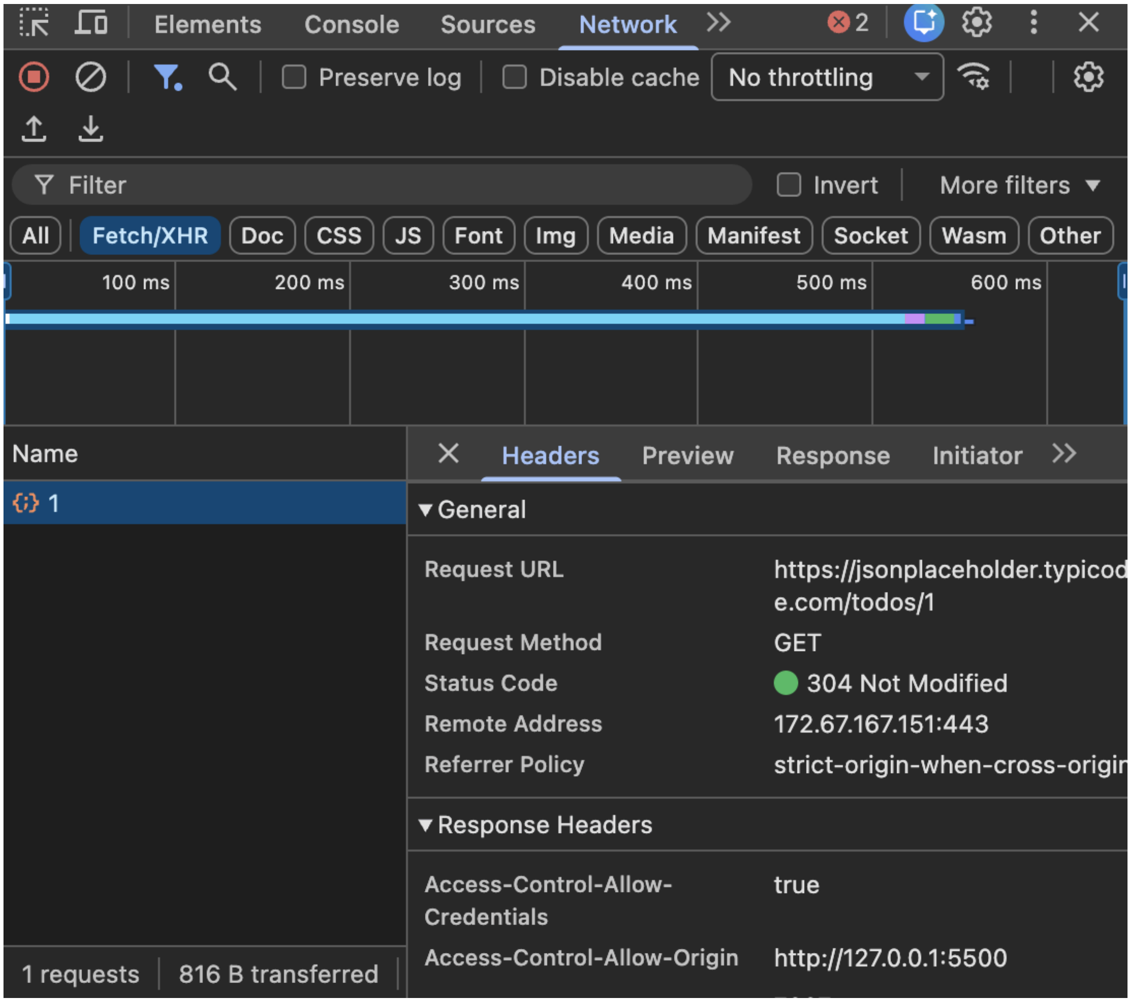
```
▼ Local
  ▶ Exception: TypeError: window.notAFunc
  ▶ this: button#btn-cause-error
▶ Closure
▶ Global                          Window
```



```
fetchSample: 550.74609375 ms                          script.js:127
fetch result                                          script.js:128
 ▶ {userId: 1, id: 1, title: 'delectus aut autem', completed: false}
```

FETCH SAMPLE BUTTON

This makes a small network request. In Network, a request appears “todos/1”. You can click it to see headers and the JSON preview. In the Console, you’ll also see timing “fetchSample:” and the fetched result printed out. A short message pops up on the page so you can tell it finished. Below is the console output.



{;} 1

▼General

Request URL	https://jsonplaceholder.typicode.com/todos/1
Request Method	GET
Status Code	● 304 Not Modified
Remote Address	172.67.167.151:443
Referrer Policy	strict-origin-when-cross-origi

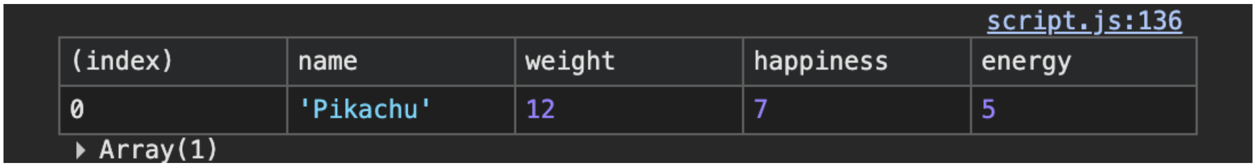
▼Response Headers

Access-Control-Allow-Credentials	true
Access-Control-Allow-Origin	http://127.0.0.1:5500

1 requests | 816 B transferred

LOG TABLE BUTTON

This prints the pet's stats in a table format in the Console. It's easier to read than a plain object. You'll see columns for name, weight, happiness, and energy in one neat row, so you can confirm the values quickly.



script.js:136

(index)	name	weight	happiness	energy
0	'Pikachu'	12	7	5

▸ Array(1)