Andrew Glick 5/18/24

Networking

Concept overview

The networking unit had two main parts. Part one (Riddles) focused on sending data over the web. In this part, we made manual HTTP requests (both get and post), learned about status codes, and finished a partially built client program for interacting with a server. Part two (Server lab) focused more on the server aspect. In this part, we interacted with a prebuilt server and then built our own.

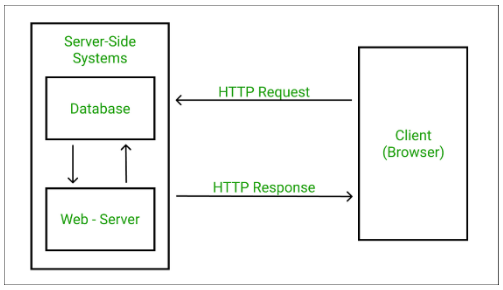

Figure 1 (from geeksforgeeks)

HTTP was a main focus of this unit as it is how clients and servers communicate. The client sends a request to the server which handles it and then send back a response. The two HTTP methods we focused on were get and post. The main difference is that get only requests data while a post both provides and requests data. A diagram is provided at figure 1. Another main focus was a client program. A client program enables us to easily make requests instead of tediously typing them out each time. Some topics that were not covered were the domain name system, TCP/IP and physical networks but they are still important as they are all pieces to the puzzle. This topic is important to learn because it shows students the relevancy of the coding they are doing to websites and applications that they use every day.

How networking could fit into my teaching context

As a prospective math and computer science educator, I do not exactly have a teaching context yet. However, I can still consider a possible teaching context using the New York State Computer Science and Digital Fluency Learning Standards. Standard 7-8.NSD.5 says that students should be able to “summarize how remote data is stored and accessed in a network” (New York State Education Department, 2020). The networking aspects we focused on help to teach students about accessing data from a server.

How I might teach networking

I was very unfamiliar with networking prior to this class, but I learned a lot throughout the process. I might teach networking in a very similar way to how it was taught in this class. I thought it was very smart to have students experiment with HTTP requests through the shell before actually implementing them in their own “front-end” program. This allows them to get a better grip of how they work. From there, they would have more knowledge so they could build their own “back-end” server to deal with the requests. This aspect would also be partly walked-through. I might also use diagrams as an assistant so students could visually what is actually going on behind the scenes. For example, I found the diagram shown above for HTTP requests to be very helpful and would use it on notes.

References

GeeksforGeeks. (2021, September 20). *State the core components of an HTTP response ?*. GeeksforGeeks. https://www.geeksforgeeks.org/self-in-python-class/

New York State Education Department. (2020). *Computer Science and digital fluency learning standards*. https://www.nysed.gov/sites/default/files/programs/curriculum-instruction/computer-science-digital-fluency-standards-k-12.pdf

Refsnes Data. (n.d.) *HTTP Request Methods*. W3Schools. https://www.w3schools.com/tags/ref_httpmethods.asp