Chris Wright

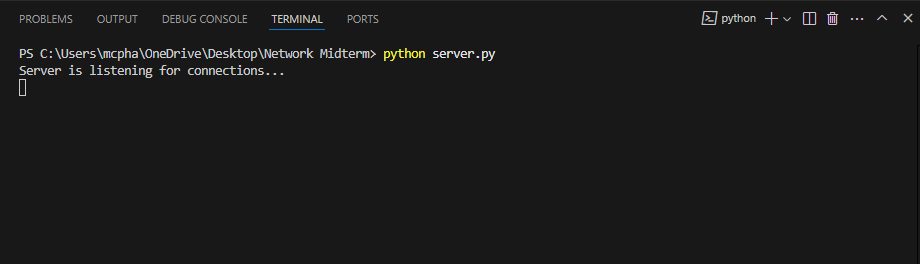
4/21/2025

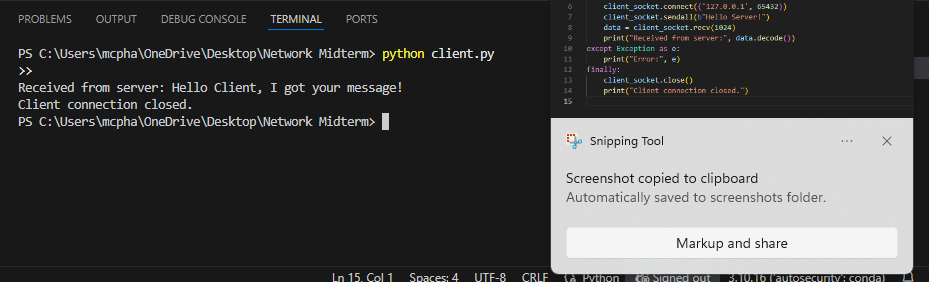
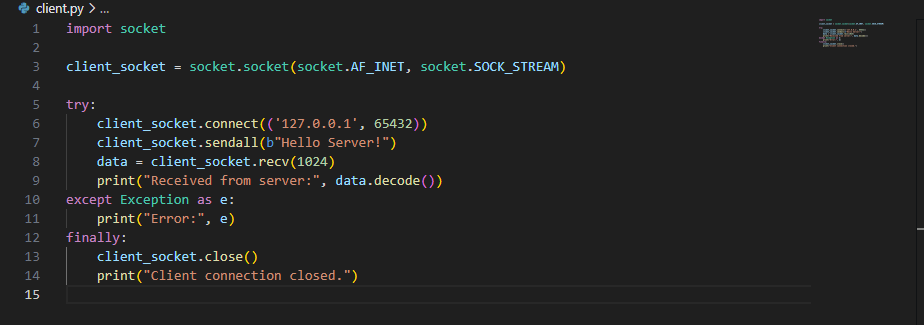
CYB333

Socket Programming & Port Scanner

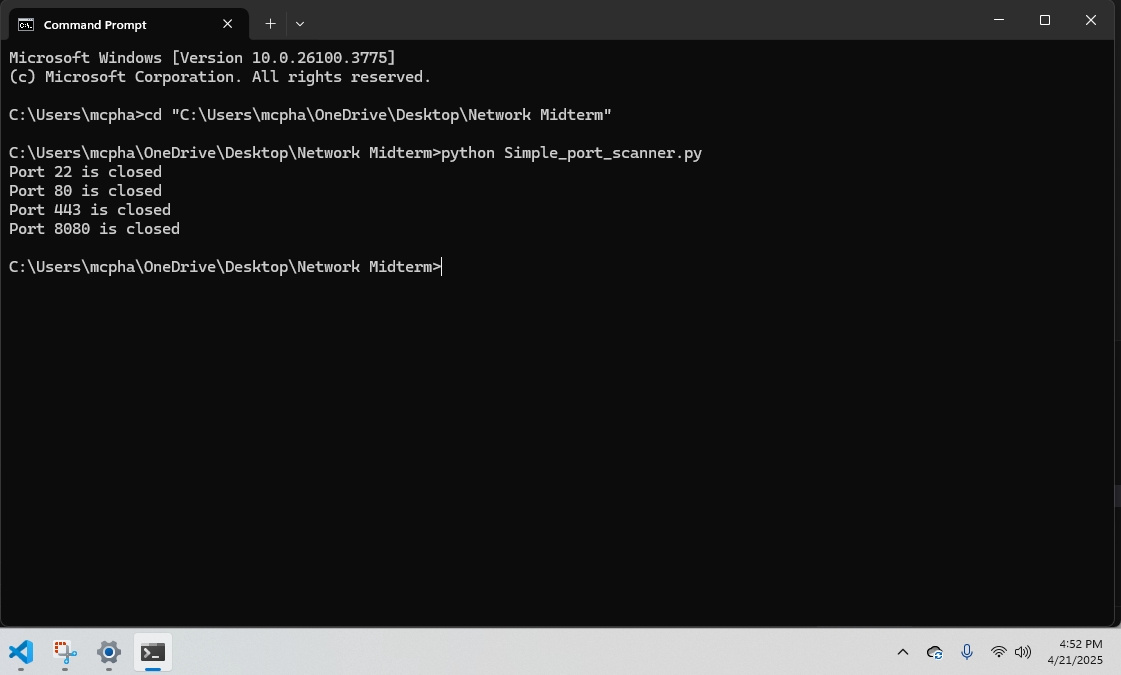
For this assignment, I worked on creating both a Python server and client script, along with a basic port scanner. I started with almost no knowledge of Python socket programming, so I needed help at nearly every step. I used ChatGPT (AI) to guide me through the process, from writing the code to fixing errors and understanding the commands.

Step 1: Setting Up the Server and Client  
  
First, I created two Python files: server.py and client.py. ChatGPT explained that the server would wait for connections, and the client would send a message once connected. I had trouble knowing where to save these files, but with help, I saved them both inside the same folder on my desktop: Network Midterm.



Step 2: Running the Server and Client  
  
After saving the files, I opened the Anaconda Prompt (in my specific environment, autosecurity) and navigated to the folder using the cd command. When I ran the server, it displayed:  
Server is listening for connections...  
When I ran the client, I saw the confirmation that the message had been sent and received. This part was fun because I saw the scripts talking to each other!  


Step 3: Error Handling and Troubleshooting  
  
I encountered a significant issue at one point: my terminal wouldn’t recognize Python. After much trial and error, I realized I needed to reinstall Python because of a PATH variable problem. ChatGPT walked me through checking the PATH settings and reinstalling Python correctly.

Step 4: Writing and Running the Port Scanner  
  
Next, I created a third file called simple\_port\_scanner.py. ChatGPT gave me the code and explained how it worked by looping through port numbers and trying to connect to see if they were open. I saved and wanted to run it, but the terminal couldn’t find the file. The issue was that I accidentally typed the file name wrong. Once I corrected that, the script scanned localhost and scanme.nmap.org, displaying the open and closed ports as expected.  


Extra Scan:  
  
I even tried scanning a custom IP address for more practice. ChatGPT explained how to change the target IP in the script, and I added that to it and ran the scan successfully.

AI Tool Usage:  
  
Throughout this entire project, I used ChatGPT to help me:  
- Write the original server, client, and port scanner code.  
- Troubleshoot errors like 'No such file or directory' and 'Python not found.'  
- Understand basic socket programming and scanning concepts.  
Here’s an example of a prompt I typed:  
I asked:  
"Why do I see the error: can't open file — No such file or directory?"

Personal Insights:  
  
This assignment taught me that even when code looks simple, running it can be challenging if your environment isn’t set up correctly. I learned that PATH settings in Windows matter a lot when using Python. It also showed me the power of asking for help, using AI as a learning tool, and sticking with a problem until it’s solved.  
  
I am confident opening terminals, navigating folders, and running Python scripts. I still don’t fully understand every piece of the code, but I can recognize the flow:  
- The server starts and waits.  
- Client connects and sends a message.  
- Port scanner loops through ports to find what’s open.  
  
I also learned the importance of error messages. Before, I would have just given up when I saw something like “No such file,” but now I know how to troubleshoot using directory checks, correct filenames, and Python environment setups.  
  
In the future, I want to understand better how sockets work under the hood and start experimenting with more advanced features, like handling multiple clients and writing cleaner, more organized code.