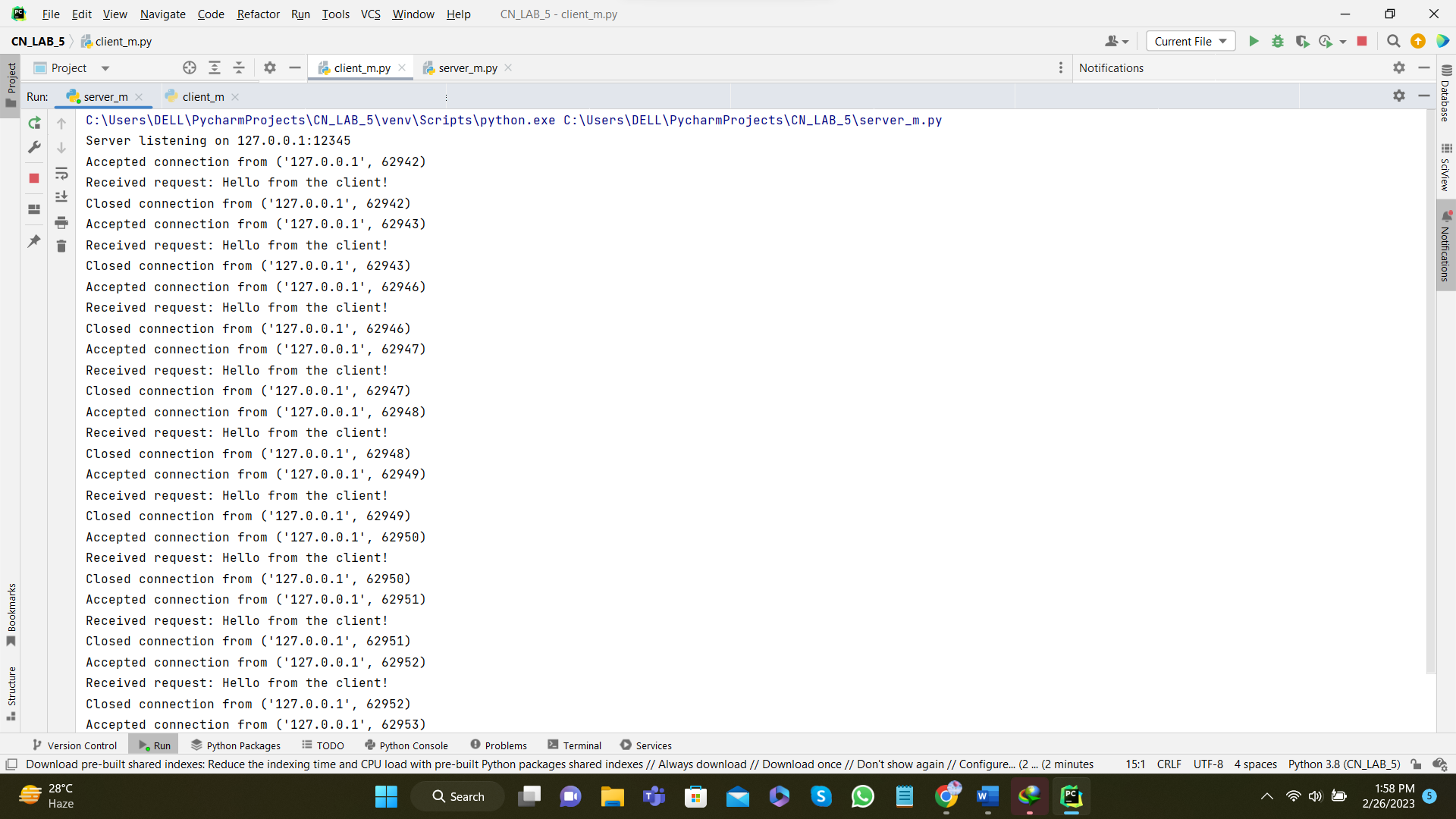
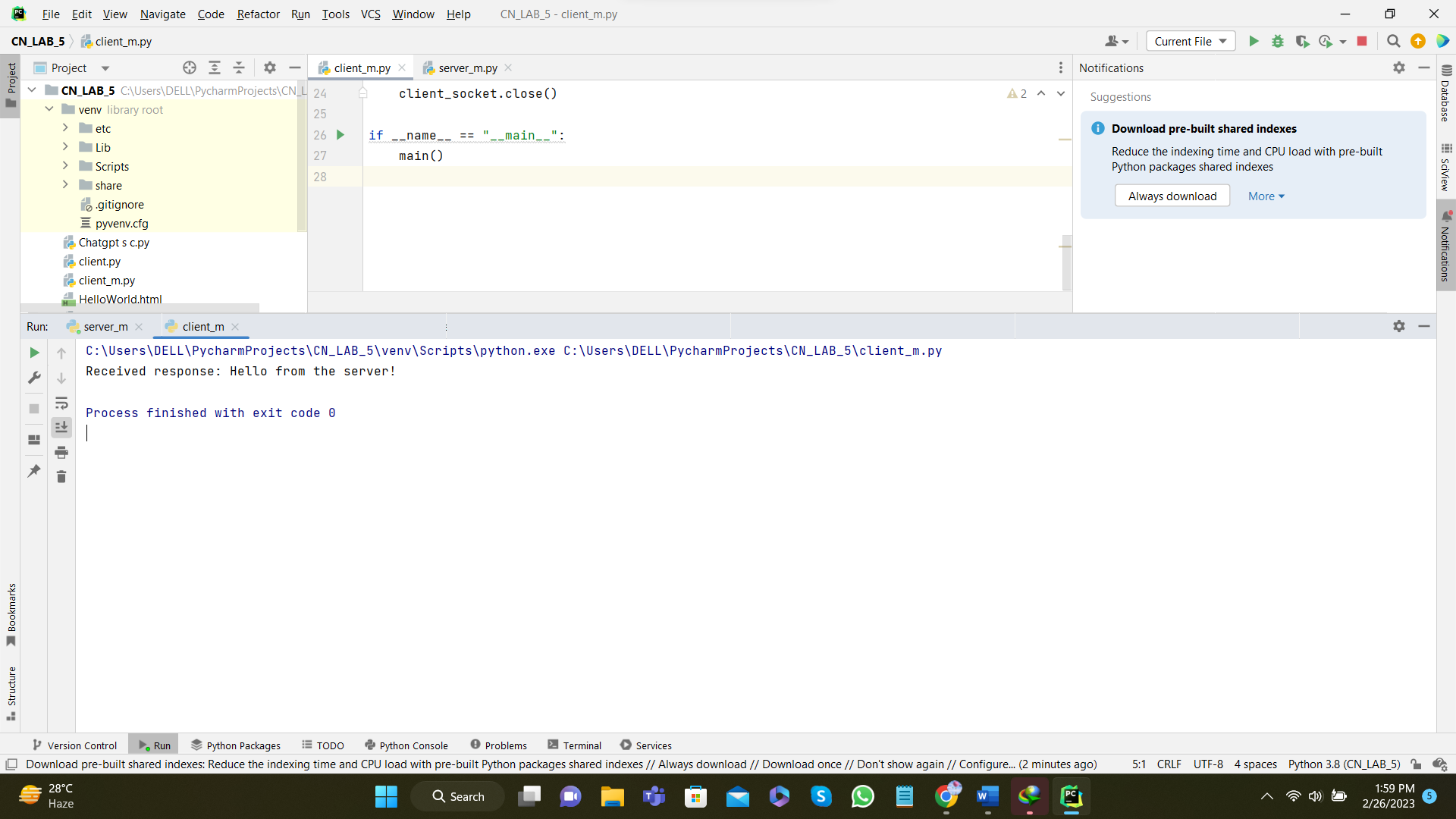


# Lab Exercises

1. Currently, the web server handles only one HTTP request at a time. Implement a multithreaded server that is capable of serving multiple requests simultaneously. Using threading, first create a main thread in which your modified server listens for clients at a fixed port. When it receives a TCP connection request from a client, it will set up the TCP connection through another port and services the client request in a separate thread. There will be a separate TCP connection in a separate thread for each request/response pair.



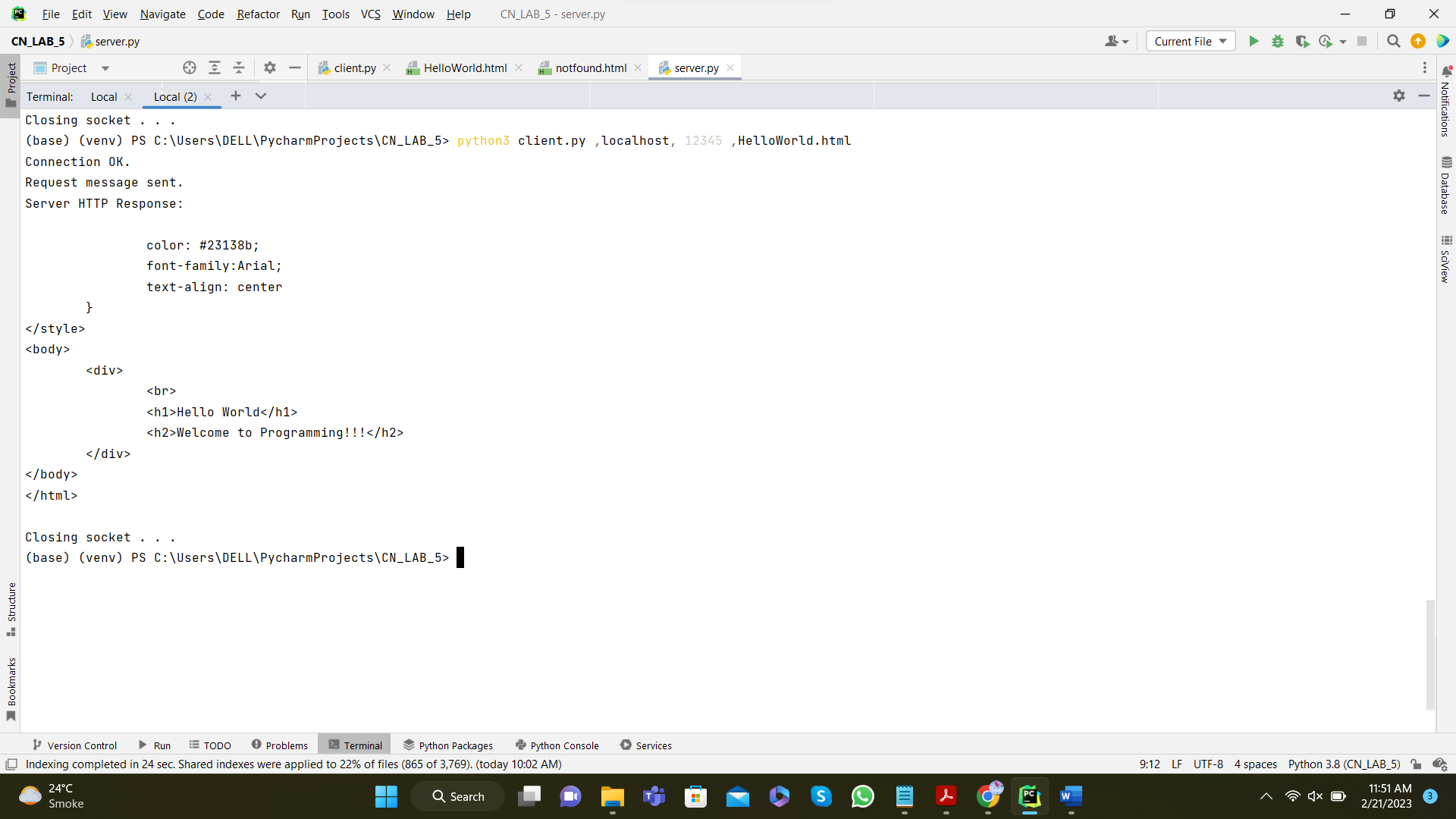


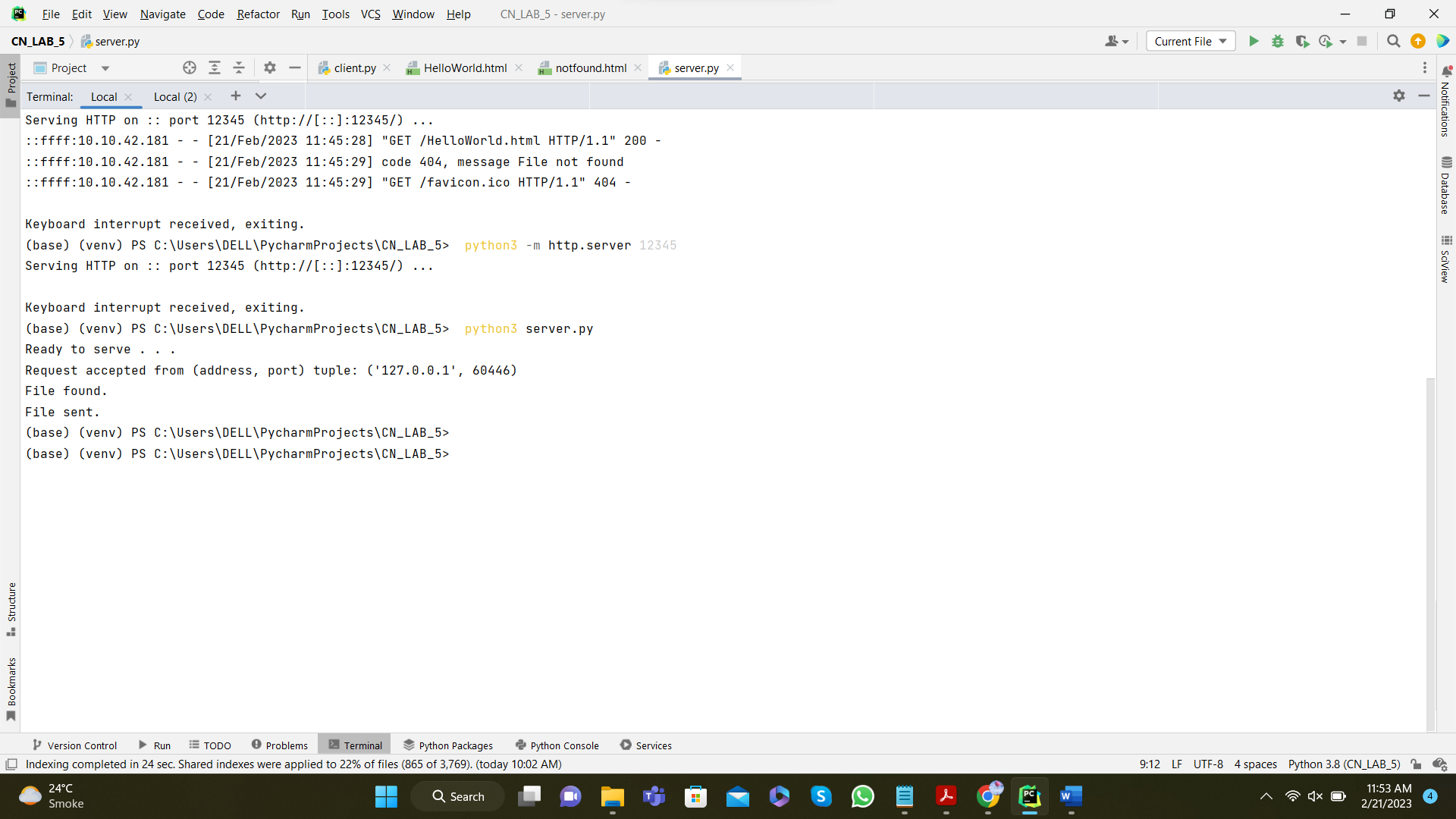
1. Instead of using a browser, write your own HTTP client to test your server. Your client will connect to the server using a TCP connection, send an HTTP request to the server, and display the server response as an output. You can assume that the HTTP request sent is a GET method.

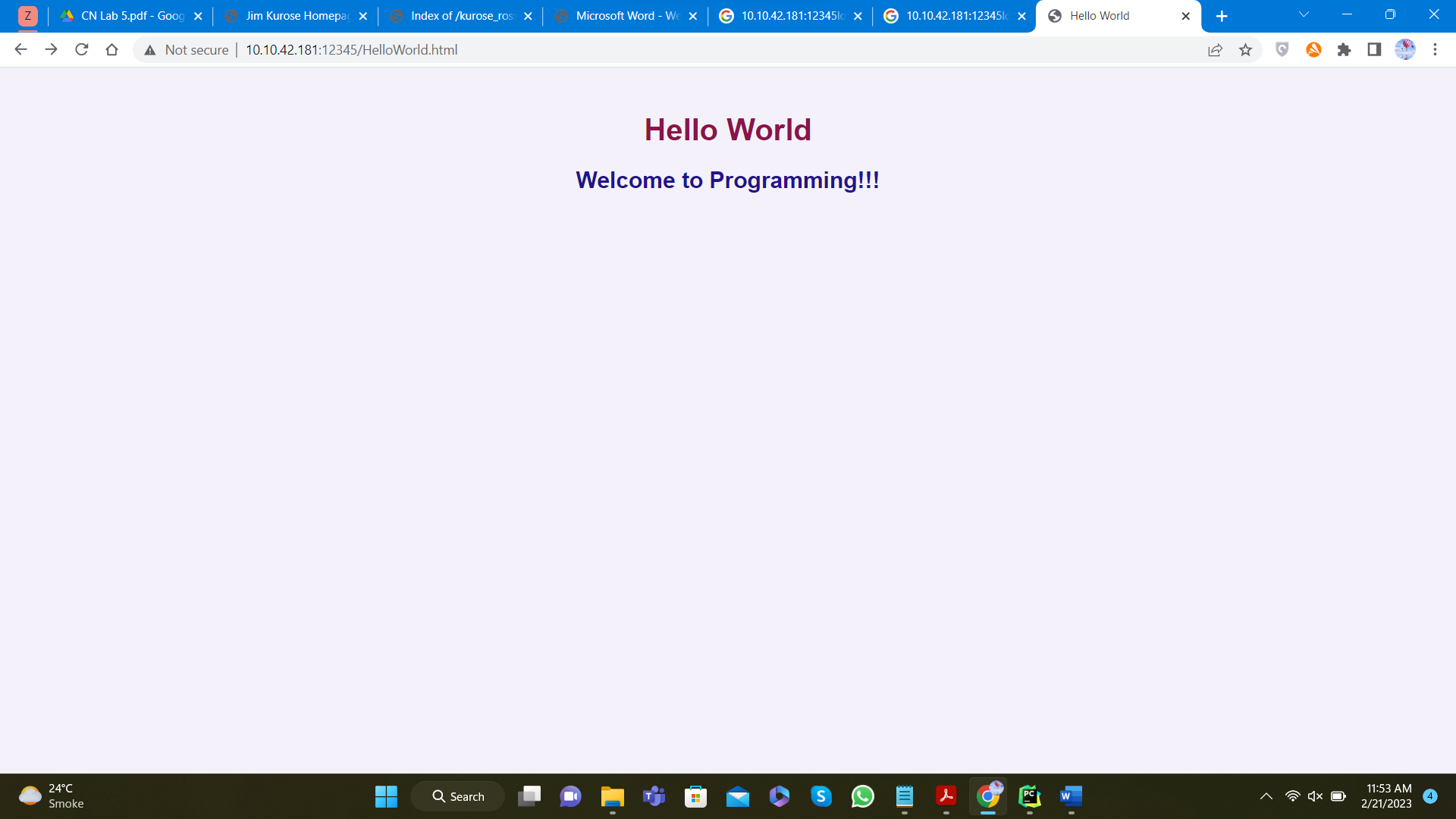
The client should take command line arguments specifying the server IP address or host name, the port at which the server is listening, and the path at which the requested object is stored at the server. The following is an input command format to run the client.

client.py server\_host server\_port filename

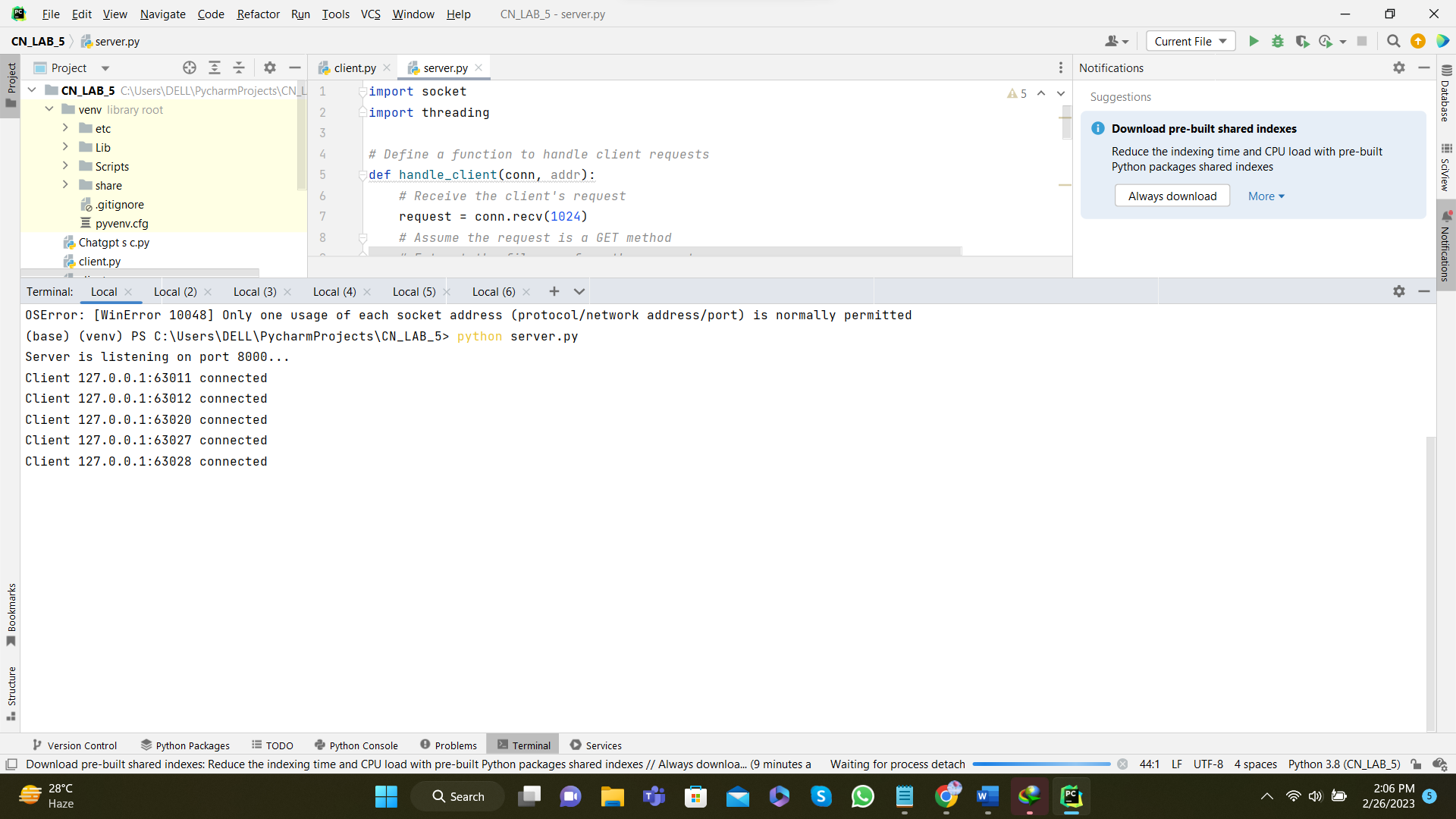
**For one client and one server:**

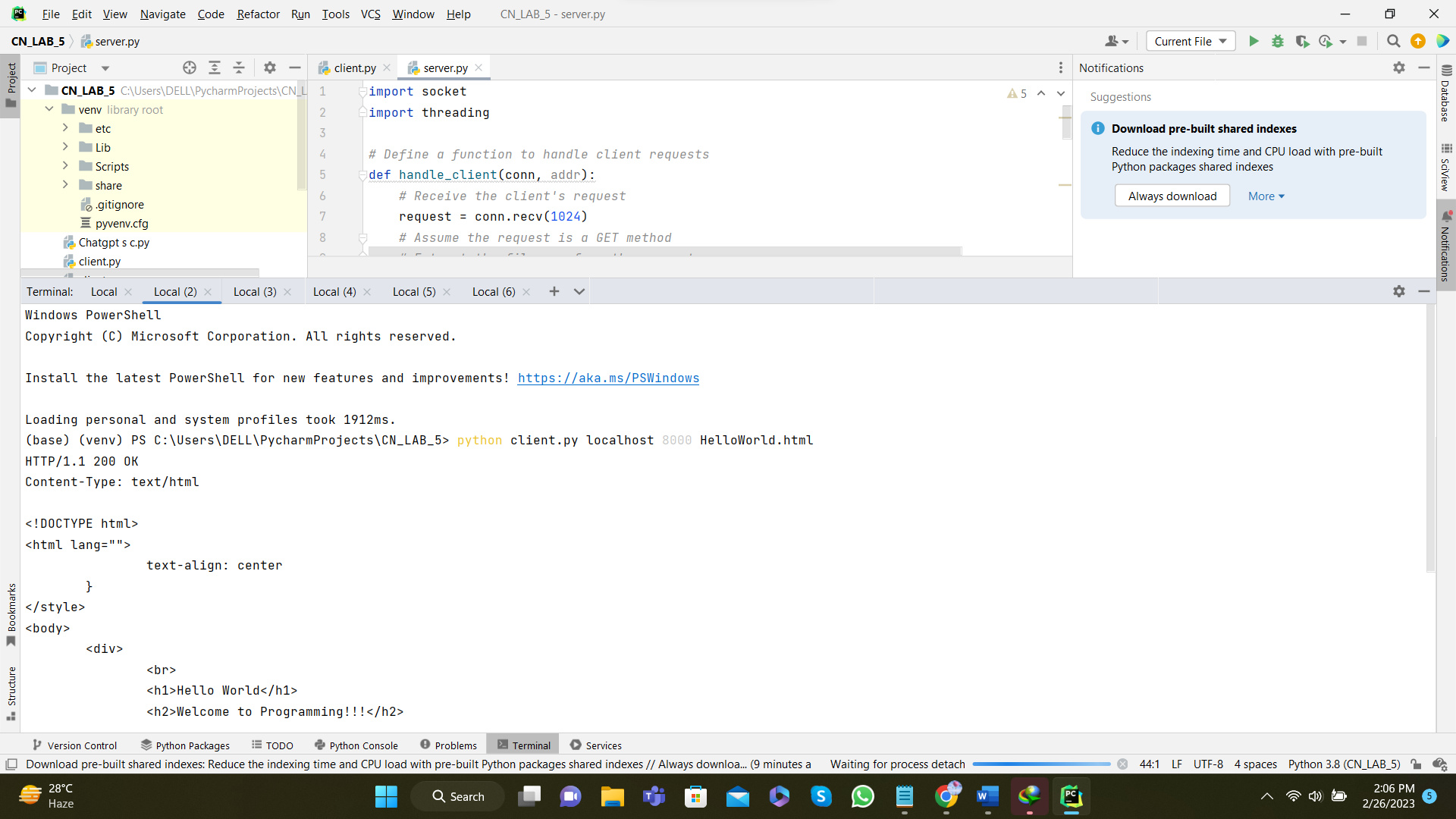


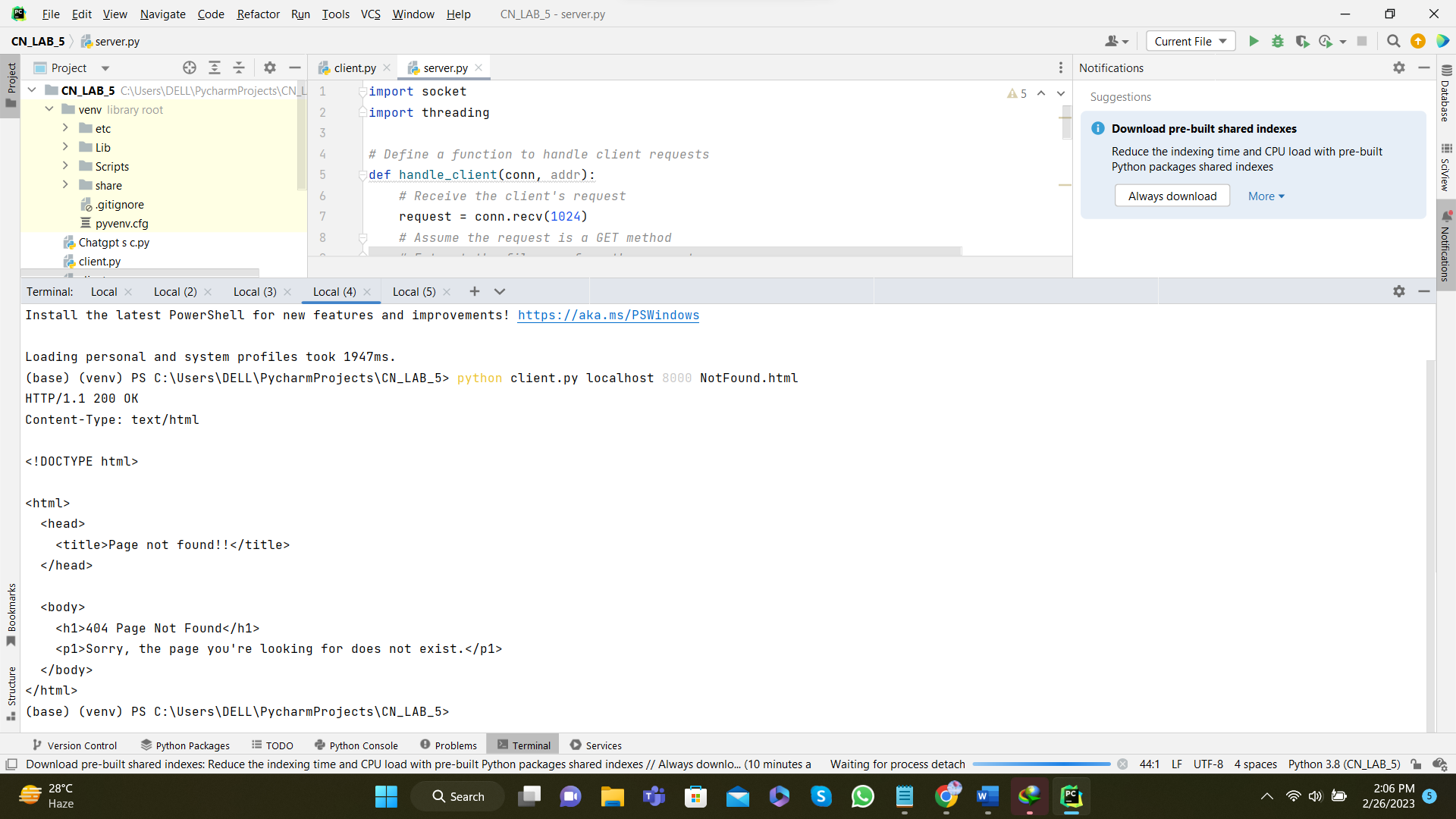




For one server and multiple clients:







**Submission Instructions:**

* All tasks are mandatory to be completed in lab.
* You will submit the complete server code along with the screen shots of your client browser, verifying that you actually receive the contents of the HTML file from the server, compiled in a document, named CNlab5\_registration\_number.docx/pdf on the Google classroom by Sunday, 26th February, 2023 11:59 P.M.
* Cheating or plagiarism will result in 0.