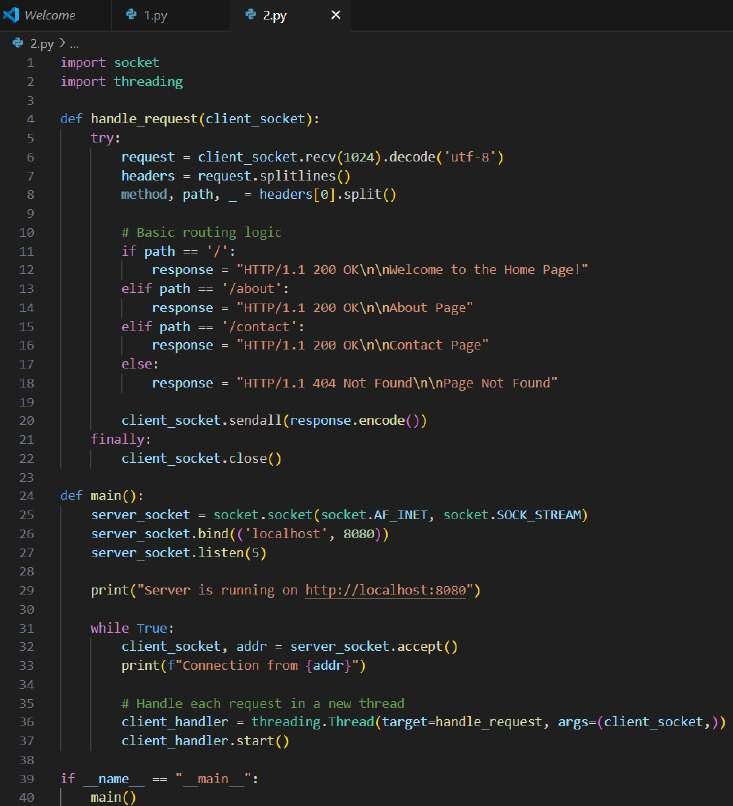
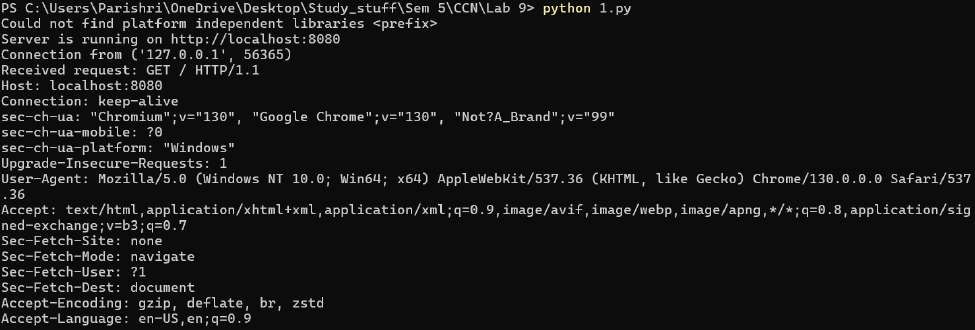
# Lab 9

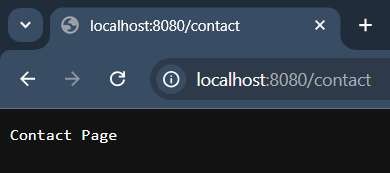
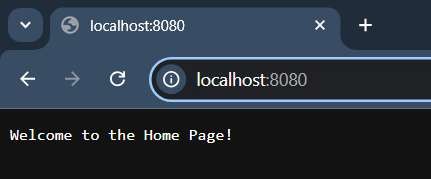
* 1. Use sockets to handle http requests

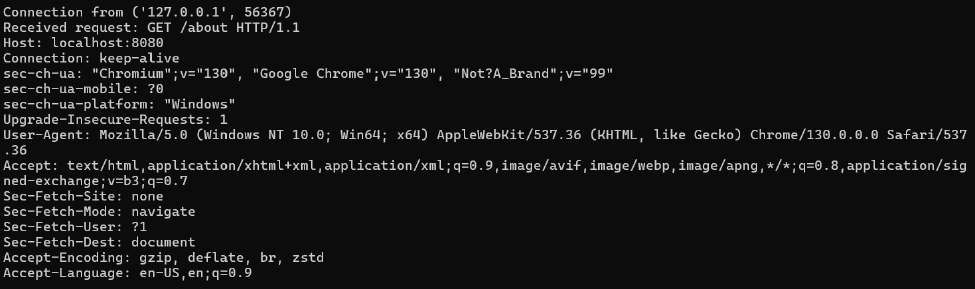


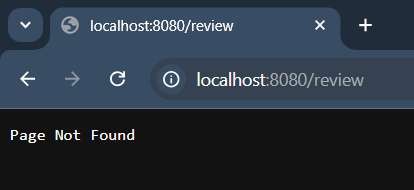
* 1. Implement basic routing for different urls











* 1. Explore how web servers handle multiple requests

**Concurrency**: Web servers like Nginx or Apache handle multiple requests through threading, multiprocessing, or asynchronous event loops (like in asyncio).

* **Threading**: Similar to the above example, threading allows the server to handle multiple connections at once.
* **Asynchronous I/O**: Non-blocking I/O (e.g., using Python’s asyncio library) can handle multiple connections more efficiently by not waiting for I/O operations to complete before moving on to the next request.

**Load Balancing**: In more advanced scenarios, web servers distribute the load across multiple threads, processes, or even multiple machines (in case of distributed systems), ensuring each request is handled promptly.

**Connection Pooling**: Servers often use connection pools to reuse open connections and handle more requests efficiently without creating new ones for each request.