Mini-Project 1 - web proxy server (part 3)

**a) What is different in request handling in a proxy server and a web server that hosts your files?**

The main difference between a proxy server and a web server is that the requests go to the origin server in a web server while the proxy server acts as both the client and server and most requests will go to the cache - depending on the hit rate.

We try to answer requests locally as much as possible with a proxy server so that we don’t have to involve the origin server, which is much less efficient.

The cache in the web proxy is a server for the original requesting client and the cache is a client to the origin server.

By the RFC 4.2.3, we can use define request methods to be cacheable - then it indicates that the responses to them can be stored for future reuse.

**Detailed specifications for minimal proxy server only using the knowledge from module 2 slides 29-34 (and consulting RFCs)**

User sets browser: Web accesses via cache

Browser sends all H T T P requests to cache

Object in cache: Cache returns object

Otherwise cache requests object from origin server, then returns object to client

Cache acts as both client and server

**b)** To test the functionality of the proxy server, we can verify that the requests are indeed being cached and contact with the origin server is being avoided. To accomplish this, we can make an initial request, check that it performs normally - with a status code of 200.

Next, we can perform the exact same request, but instead of receiving a 200 status code, we should expect to see 304 Not Modified. That’s because that would indicate our cache is working since the stored object is returned. (Command 1 + 2)

Additionally, we can check that the proxy server recognizes invalid requests and provides the appropriate response/status code. (Command 3)

Command 1

curl -x http://127.0.0.1:8888 -v http://example.com/

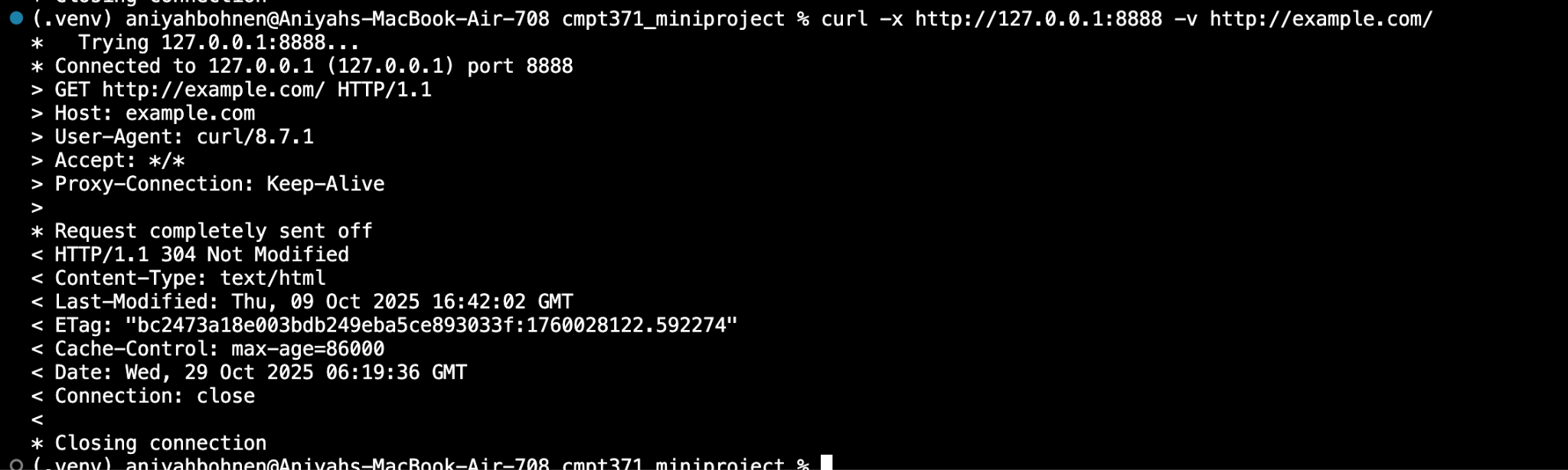
Output 1



Command 2

curl -x http://127.0.0.1:8888 -v http://example.com/

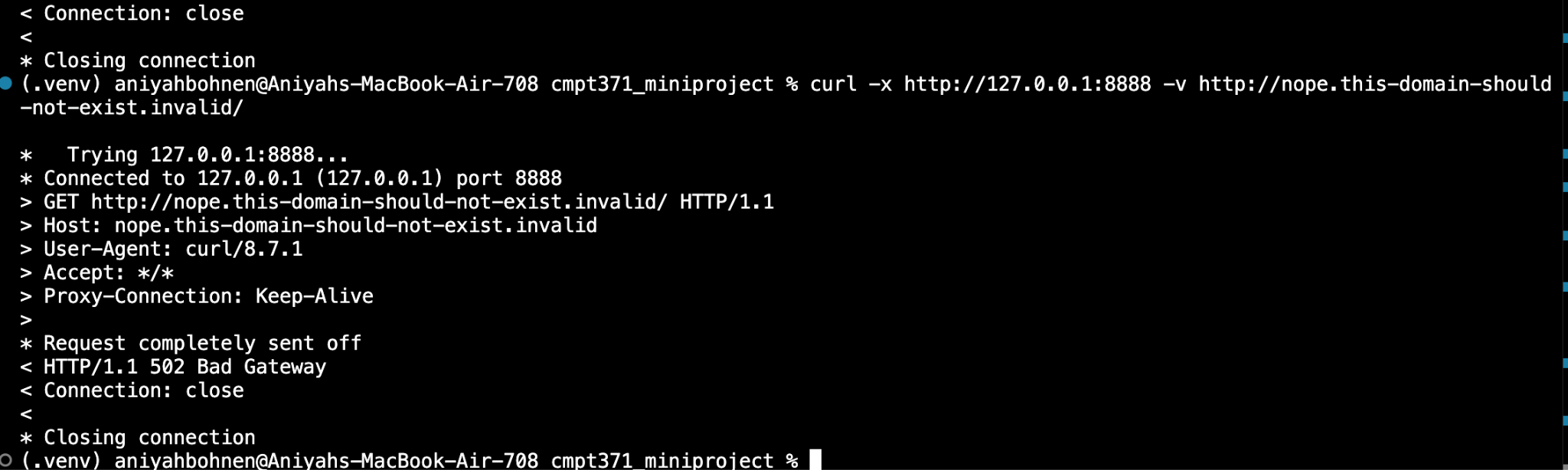
Output 2



Command 3

curl -x http://127.0.0.1:8888 -v http://nope.this-domain-should-not-exist.invalid/

Output 3



**c) Explain why your web server is multi-threaded and how it impacts performance.**

The web server is multi threaded because for each connection to the socket, there is a new thread spawned. This helps with efficiency since we do not need to create a new [..]