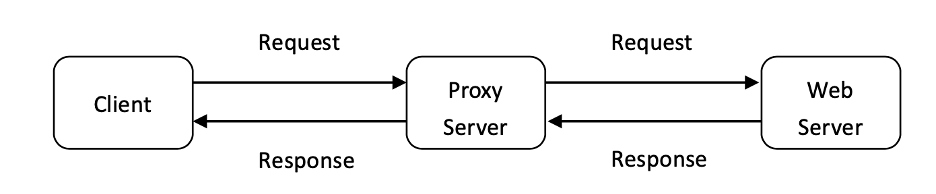
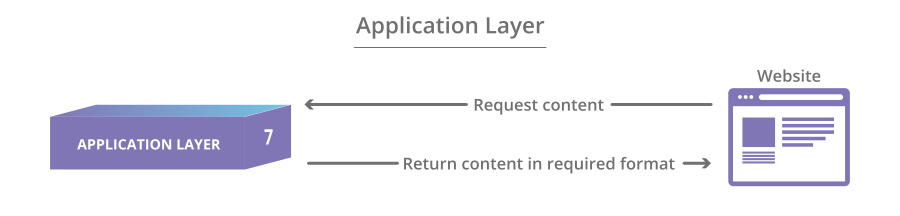
This exercise was to develop an understanding of how web proxy servers work as well as an understanding of cache. This exercise involved creating a small web proxy server which only accepted GET-requests and caching the web pages. This exercise was not exclusive to HTML pages but included images as well. It is important to note that this assignment and skeleton code was provided in Python. The final server code was instead written in Java as that is the only language I know. This assignment represented the fifth layer, the Application layer (layer 7) of the OSI model.

As a lesson learned from this exercise, I would say I should have learned python so I could start with the skeleton. I jest- but on a more serious note, I would say in future I would have spent more time ensuring that that I fully understand everything that was being supplied in the provided code before starting as my method of working on writing code as I read through the skeleton was not very effective and provided a lot of debugging opportunities. I did not manage to get the code running or ensure it was totally fulfilling the requests of the assignment because I did not know how to set up the proxy link on my internet browser. As such, I do not have screenshots showing completion. Part of this issue may come from my translations to Java- perhaps I was correctly setting up the proxy settings in my browser but did not have the coding correct.



*Figure 5. 1*

This exercise is part of the seventh layer of the OSI model- the application layer. The exercise involves creating a proxy server- this means that the request from the client go first to the proxy server which then forwards the requests onwards. The server then responds to the initial request, and sends back a response. The response is received by the proxy server, which then forwards the request to the client which initially requested the information. This code specifically includes a function to cache the webpages that way on a second request. In this scenario, the client requests a web page, and the request is received by the proxy server. Because the proxy server has cached the web page, the proxy server responds directly to the client. Figure 5.1, from the assignment, visualizes this process. All of this is done at the application layer, from application to application. Figure 5.2 shows how exactly Exercise 5 relates to the application layer work.



*Figure 5. 2*