**Assignment 1:**

**1.1/**

We started by typing

**telnet** [**www.weer.nl**](http://www.werl.nl/) **80|tee out**

on the terminal to both establish a connection with the domain and get a file named “out” which contains the entire conversation with the host.

We started by getting the header by typing:

**HEAD / HTTP/1.1**

**HOST:** [**www.weer.nl**](http://www.weer.nl/)

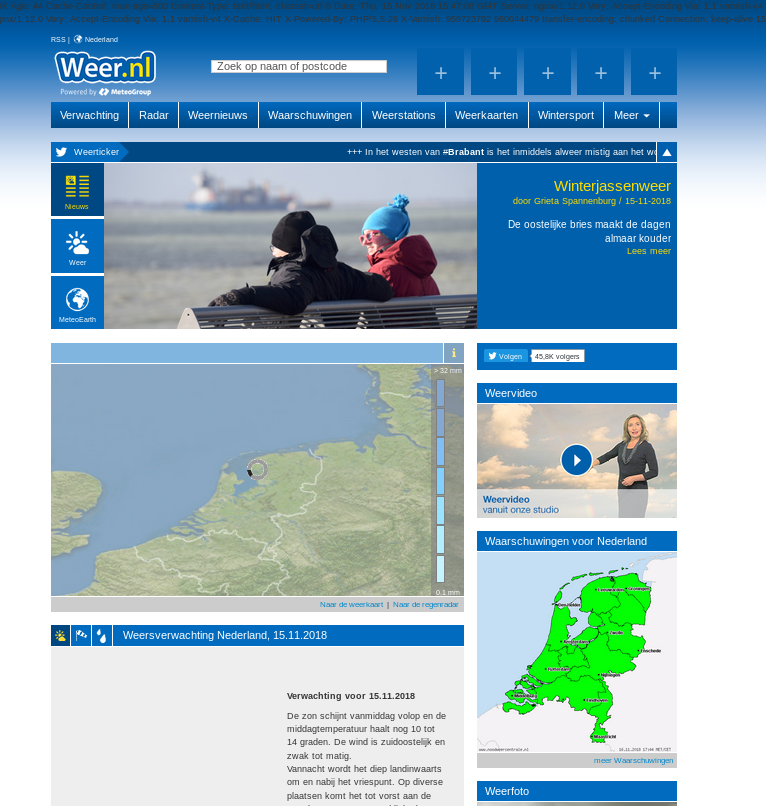
the next thing we did was to get the actual document from the host by typing:

**GET / HTTP/1.1**

**HOST:** [**www.weer.nl**](http://www.weer.nl/)

**1.2/**

We added the extension .html to the out file and opened it. When opening it we got a site that was similar to the actual weer.nl website, the main differences between the two were that the out.html had a visible text on top which was the response from the header request and the fact that live-updated things such as ads or the updated weather map weren’t there.

****

**Actual weer.nl out.html File**

**1.3/**

According to lifewire.com, the tag *X-UA-Compatible* is used to specify which version of Internet Explorer the site admits, it has to be on the top side of the header so IE can run that version.

Since in the case of our out file the X-Cache indicates a hit we conclude that it specifies whether or not a cached file of the site is available.

**1.4/**

Since the header says Cache-Control: max-age=600 we conclude that it specifies the requisites a cache version of the page needs to have to be valid, in this case the cached version can’t have an age older than 600.

**2.1/**

For this part of the assignment we tested all 3 possible scenarios regarding the content length when using the PUT command: equal, longer and shorter.

For this we used the text “Hello World!” which contains 12 characters.

* Our first scenario was the equal length, the data field returned “Hello World!” and the HTTP code was 200 so the resource was modified correctly
* When inputting a Content-length shorter than 12, in this case 8, the data field contained the first 8 characters: “Hello Wo” and we got a code 200 OK but after that we also got a bad request code from the last 4 characters.
* When the length field is longer, the site will just wait till all the content is provided so if you just press enter, the site will wait until the number of new lines equals the amount of characters needed to get to the number input from length.

In case the Content-length is 20, the data field you would get would be “Hello World!\r\n\r\n\r\n\r\n" and you would still get a 200 OK code.

**3.1/**

When we open the site the browser request a user and a password, when inputting the user and password provided in the github this is what we get:

{

"authenticated": true,

"user": "user"

}

After reloading the page, the same text appears.

**3.2/**

When inputting this:

telnet httpbin.org 80

HEAD /basic-auth/user/passwd HTTP/1.1

host:httpbin.org

<carriage return>

**4.1/**

The game we’ve chosen is Snakes and Ladders

**4.2/**

**Examples:**

* **ToyTheater** (<https://toytheater.com/snakes-and-ladders/>):
  + Positive:
    - Pretty self-explanatory, the die blinks when it’s the player’s turn.
    - Very low level of clicks per iteration, if the user needs to see the rules, exit the game or turn the sound on/off, just 2 clicks are needed.
  + Negative:
    - A lot of noise, pretty stacked rows, too many colors and pretty big ratio snake:square which means that in some squares you can’t see the number and in others you can’t tell whether a snake’s head is on a square or the next one.