Web & Database Technology

## 1. HTTP request messages: GET/HEAD

##### 1.1

Input:

1. telnet weer.nl 80| tee out
2. ...
3. HEAD / HTTP/1.1

Output:

1. HTTP/1.1 301 Moved Permanently
2. Date: Tue, 20 Nov 2018 14:19:04 GMT
3. Server: Apache
4. Status: 301 Moved Permanently
5. Location: http://www.weer.nl/
6. Content-Type: text/html; charset=UTF-8

Conclusion: The website is moved to a different location.

##### 1.2

Input:

1. telnet www.weer.nl 80| tee out2
2. ...
3. GET /regenradar/nederland HTTP/1.1

Output:

1. HTTP/1.1 200 OK
2. Accept-Ranges: bytes
3. Age: 75
4. Cache-Control: max-age=600
5. Content-Type: text/html; charset=utf-8
6. Date: Tue, 20 Nov 2018 13:46:18 GMT
7. Server: nginx/1.12.0
8. Vary: Accept-Encoding
9. Via: 1.1 varnish-v4
10. X-Cache: HIT
11. X-Powered-By: PHP/5.5.26
12. X-Varnish: 1045901517 1038459742
13. transfer-encoding: chunked
14. Connection: keep-alive
16. 4dcd
17. <!DOCTYPE html**>**
18. **<html** lang="nl"**>**
19. ...
20. **</html>**

Conclusion: except for several .css and .js files the website is identical

##### 1.3

The purpose of the X-cache : HIT is to show the client that the website wasn’t accessed from the origin server directly but instead it was accessed from an cache server.

##### 1.4

The purpose of cache-control tells us how long a cache server considers the copy of the website to be up-to-date. After the time limit the cache server send a new request to the origin server to get a new copy. For the website [www.weer.nl](http://www.weer.nl) the Cache-Control had a max-age of 600 seconds.

## 2. HTTP request messages: PUT

##### 2.1

Input:

1. telnet httpbin.org 80
2. ..
3. PUT /put HTTP/1.1
4. host:httpbin.org
5. Content-type:text/plain
6. Content-length:12
7. **<carriage** return**>**
8. Hello everyone!
9. **<carriage** return**>**

Output for exceeding the content-length:

1. HTTP/1.1 200 OK
2. Connection: keep-alive
3. Server: gunicorn/19.9.0
4. Date: Tue, 20 Nov 2018 14:25:00 GMT
5. Content-Type: application/json
6. Content-Length: 298
7. Access-Control-Allow-Origin: \*
8. Access-Control-Allow-Credentials: true
9. Via: 1.1 vegur
11. {
12. "args": {},
13. "data": "Hello everyo",
14. "files": {},
15. "form": {},
16. "headers": {
17. "Connection": "close",
18. "Content-Length": "12",
19. "Content-Type": "text/plain",
20. "Host": "httpbin.org"
21. },
22. "json": null,
23. "origin": "145.94.177.131",
24. "url": "http://httpbin.org/put"
25. }
26. HTTP/1.1 400 Bad Request
27. Connection: close
28. Server: Cowboy
29. Date: Tue, 20 Nov 2018 14:24:59 GMT
30. Content-Length: 0

Conclusion: The server only stores the given amount of characters and afterwards closes the connection. e.g. We tried to store “Hello everyone” while given the content-length:12. The server only store “Hello everyo” and closed the connection.

Output for using less that the content-length:

1. HTTP/1.1 200 OK
2. Connection: keep-alive
3. Server: gunicorn/19.9.0
4. Date: Tue, 20 Nov 2018 14:30:44 GMT
5. Content-Type: application/json
6. Content-Length: 304
7. Access-Control-Allow-Origin: \*
8. Access-Control-Allow-Credentials: true
9. Via: 1.1 vegur
11. {
12. "args": {},
13. "data": "Hello!\r\n\r\n\r\n",
14. "files": {},
15. "form": {},
16. "headers": {
17. "Connection": "close",
18. "Content-Length": "12",
19. "Content-Type": "text/plain",
20. "Host": "httpbin.org"
21. },
22. "json": null,
23. "origin": "145.94.177.131",
24. "url": "http://httpbin.org/put"
25. }

Conclusion: The server waits until the client has fulfilled the amount of characters. E.g. We used “Hello!” and the server did not respond until 12 characters were send.

## 3. Basic authentication

##### 3.1

When we logged in to the site with the given username and password we got the following response:

1. {
2. "authenticated": true,
3. "user": "user"
4. }

After refreshing it page we got the same message::

1. {
2. "authenticated": true,
3. "user": "user"
4. }

Conclusion: The user stays logged in the browser

##### 3.2

Input with credentials given:

1. HEAD /basic-auth/user/passwd HTTP/1.1
2. host:httpbin.org
3. Authorization: Basic dXNlcjpwYXNzd2Q=

Output:

1. HTTP/1.1 200 OK
2. Connection: keep-alive
3. Server: gunicorn/19.9.0
4. Date: Wed, 21 Nov 2018 10:14:23 GMT
5. Content-Type: application/json
6. Content-Length: 47
7. Access-Control-Allow-Origin: \*
8. Access-Control-Allow-Credentials: true
9. Via: 1.1 vegur

Input without (again) giving credentials:

1. HEAD /basic-auth/user/passwd HTTP/1.1
2. host:httpbin.org

Output:

1. HTTP/1.1 401 UNAUTHORIZED
2. Connection: keep-alive
3. Server: gunicorn/19.9.0
4. Date: Wed, 21 Nov 2018 10:14:57 GMT
5. Www-Authenticate: Basic realm="Fake Realm"
6. Access-Control-Allow-Origin: \*
7. Access-Control-Allow-Credentials: true
8. Content-Length: 0
9. Via: 1.1 vegur

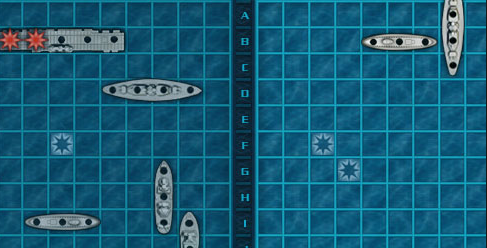
Conclusion: If we tried to get the HEAD of the page without logging in again, we got an authorization error. Meaning, with telnet the user has to log in every time over again.

## 4. Web programming project: board game app

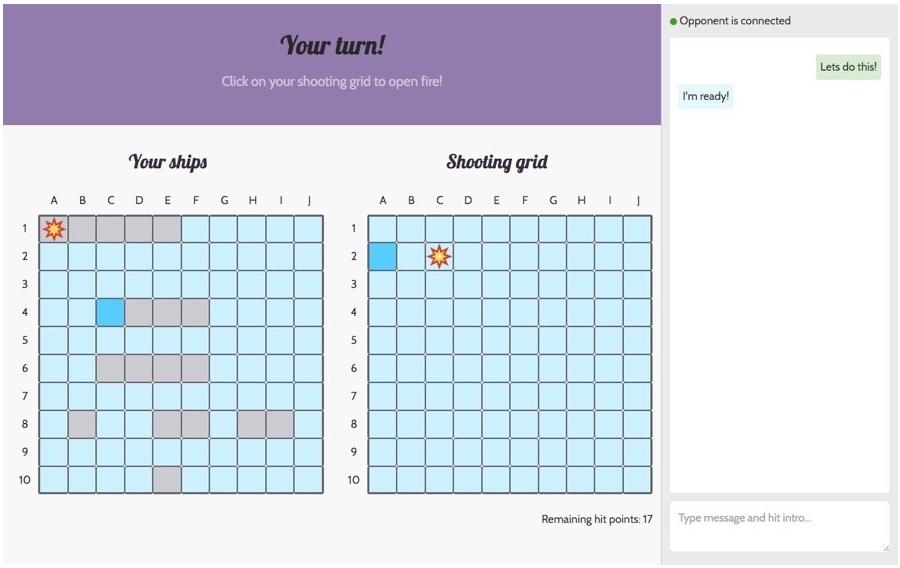
##### 4.1

We have chosen to recreate battleships

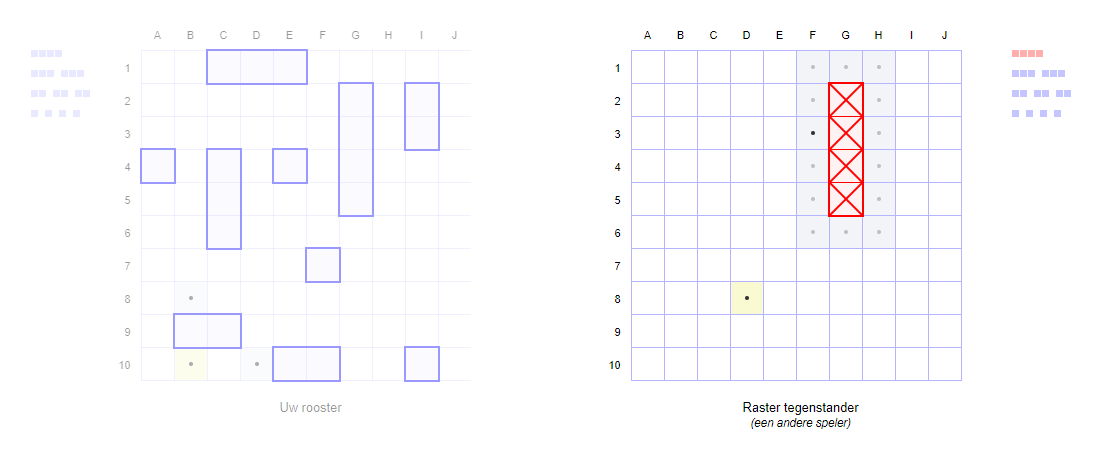
##### 4.2



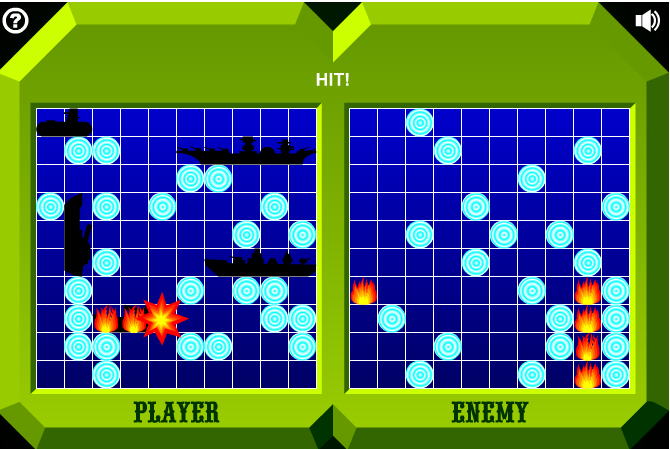
Example 1



Example 2 – source: https://alternativeto.net/software/phoenix-battleship/



Example 3 – source: http://nl.battleship-game.org/



Example 4 – source: https://www.mathplayground.com/battleship.html

Pros:

* + Since the game can be created rather simple (with square blocks and different shades of colors), there are no needs for sprites. But using sprites will make the game more appealing to the players
  + All the user interfaces from the examples show a clear distinction between the player’s field board and the enemy. In contrast to using only one playing field.
  + The game uses a grid-based layout, therefore it is easy to distinguish different sections
  + Anyone familiar with the game will understand immediately what actions to take. People unfamiliar with the game will need to go through a set of instructions but since the game is simple the player will probably understand the game rather quickly

Cons:

* + Finding sprites from the same theme is definitely is challenge. Since we both don’t have any knowledge about photoshopping or image editing we have to find single-themed sprite’s ourselves. And in addition these sprite’s must be somewhat similar themed with our backgrounds.
  + Drag ‘n drop battleships before the start of the game might be difficult to implement, since the ships must auto-align to the nearest cells of the grid. Alternative solution might be to use on-screen controls to manoeuvrethe ships along the grid.