



GEVORDERDE WEBAPPLICATIES

Verslag

Peter De Winne

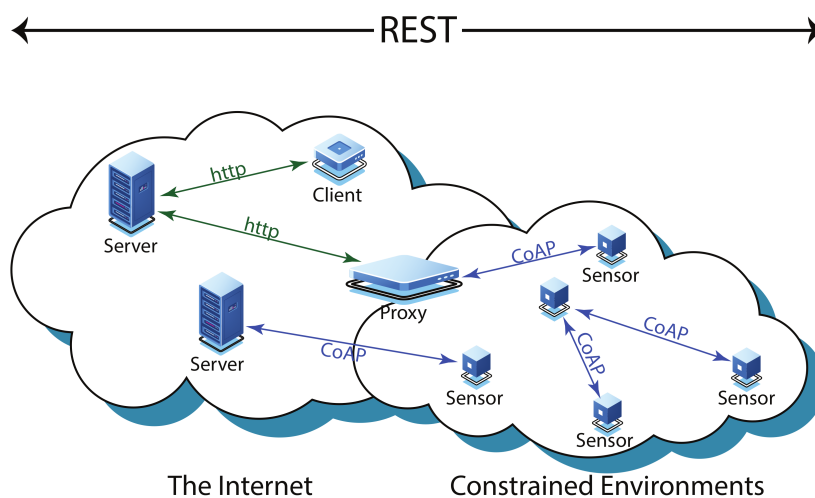
2022-2023

Inhoudsopgave	i
1 Inleiding	1
1.1 CoAP	1
2 HTML en CSS	2
2.1 HTML	2
2.2 CSS	2
2.2.1 Wat zijn de voordelen van CSS	3
3 jQuery	5
4 Flask	6
4.1 Websockets	6

Voor het vak gevorderde webapplicaties werd gevraagd een webapplicatie te bouwen. Deze webapplicatie moet in staat zijn om de lampen in het lokaal K4.04 aan te sturen. Om hier succesvol in te zijn, moeten we gebruik maken van verscheidene technologieën zoals websockets, jQuery, flask etc. Al deze technologieën zullen verder in dit verslag besproken worden.

1.1 CoAP

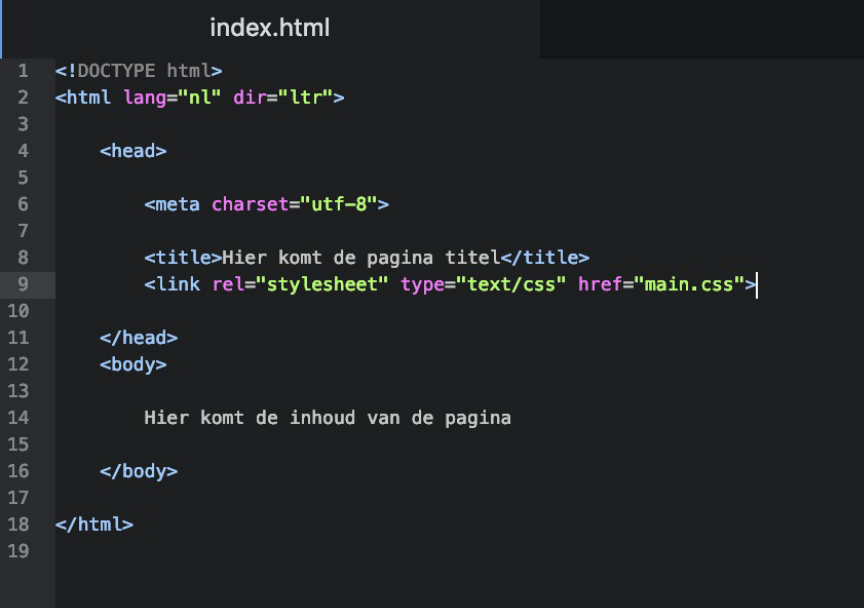
De lampen in het K4.04 lokaal maken gebruik van het CoAP protocol. CoAP of Constrained Application Protocol is een vereenvoudigde versie van HTTP en heeft vier methoden voor de request/response laag: 'POST', 'GET', 'PUT', 'DELETE'. CoAP maakt gebruik van het UDP protocol voor de uitwisseling van de berichten. Confirmable, Non-confirmable, Acknowledgment en reset zijn de vier soorten berichten. CoAP is ontwikkeld zodat deze eenvoudig naar HTTP kan vertaald worden. Een protocol dat voornamelijk gebruikt wordt voor communicatie over het world wide web.



Figuur 1.1: Voorstelling implementatie CoAP

2.1 HTML

HTML of Hyper Text Markup Language is een opmaak taal. Deze wordt voornamelijk gebruikt voor het opmaken van webpagina's. Hoewel HTML in staat is voor het verzorgen van de vormgeving en styling gebeurt dit vaak met CSS. Een HTML pagina is relatief gezien heel simpel opgebouwd. Het start steeds met een DOCTYPE-declaratie. Deze declaratie is voor de browser zodat deze weet welke standaarden ingeladen moeten worden. Hierna volgt de "head"declaratie hierin geven we onder andere de titel en links naar stylesheets en scripts door. Onder de "head" komt de "body", hierin komt alle informatie die bedoeld is voor de pagina. De meeste browsers maken het heel eenvoudig om de broncode van de website te bekijken. als je de combinatie ctrl + u geeft krijg je de broncode.

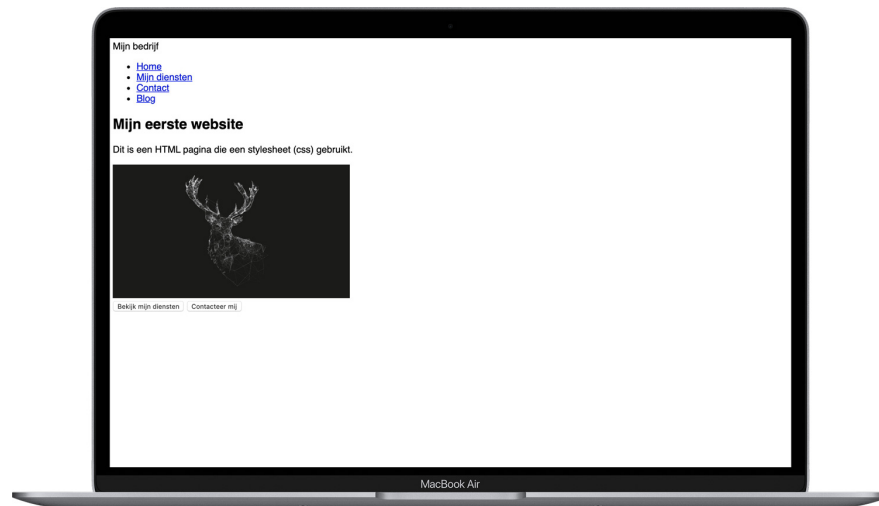


```
index.html
1 <!DOCTYPE html>
2 <html lang="nl" dir="ltr">
3
4   <head>
5
6     <meta charset="utf-8">
7
8     <title>Hier komt de pagina titel</title>
9     <link rel="stylesheet" type="text/css" href="main.css">
10
11   </head>
12   <body>
13
14     Hier komt de inhoud van de pagina
15
16   </body>
17
18 </html>
19
```

Figuur 2.1: HTML-voorbeeld

2.2 CSS

CSS of Cascading Style Sheets is een opmaak taal bedoeld voor de vormgeving van een HTML bestand. Een HTML bestand bevat enkel de inhoud (tekst, afbeeldingen, links...) die de gebruiker te zien krijgt. Elke HTML element is met CSS apart op te maken. Verder kan je ook gebruik maken van classes (class=) en identificeerders(id=) om dezelfde HTML elementen een verschillende vorm te geven. CSS werkt cascaderend oftewel wat onderaan staat in het document of wat het "diepst" staat in de HTML zal van toepassing zijn. Zo zal voor de class logo het lettertype 25px groot zijn terwijl voor de rest het lettertype 16px is. verder valt ook



Figuur 2.2: HTML zonder CSS

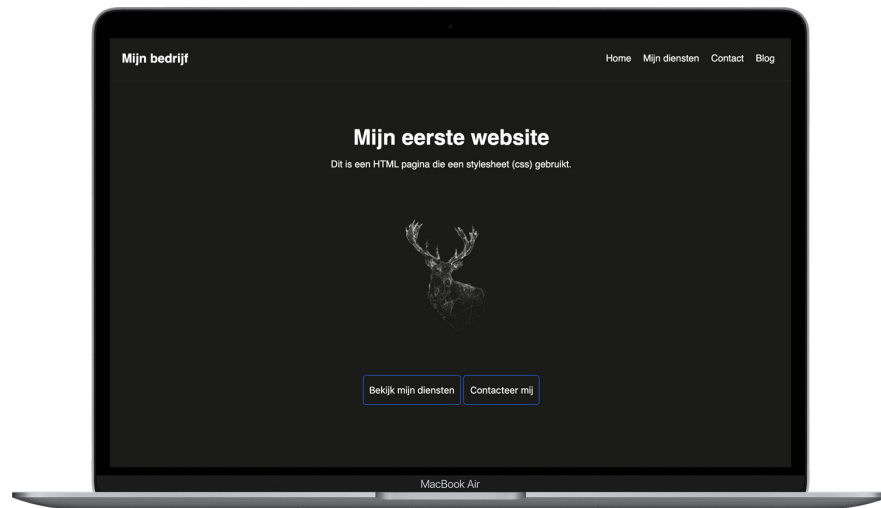
```
1  ▾ body {  
2  
3      margin: 0;  
4      padding: 0;  
5      font-family: Proxima-nova,Helvetica,sans-serif;  
6      font-size: 16px;  
7      background-color: white;  
8      color: black;  
9  }  
10  
11 ▾ .image {  
12  
13     width: 400px;  
14     margin-top: 50px;  
15     margin-bottom: 50px;  
16 }  
17  
18 ▾ ul {  
19  
20     list-style-type: none;  
21     display: flex;  
22     flex-direction: row;  
23 }  
24  
25 ▾ a {  
26  
27     text-decoration: none;  
28     color: white;  
29 }  
30  
31 ▾ a:hover {  
32  
33     text-decoration: underline;  
34     color: blue;  
35 }  
36  
37 ▾ .logo {  
38  
39     font-size: 21px;  
40     font-weight: 700;  
41     margin-left: 25px;  
42 }
```

Figuur 2.3: CSS voorbeeld

op te merken dat dat je bij bepaalde acties, je opmaak kan veranderen. Zo zal als je over het a element zweeft met je muis de kleur veranderen en de tekst onderlijnd worden.

2.2.1 WAT ZIJN DE VOORDELEN VAN CSS

- Je kan je design kan tot de pixel nauwkeurig uitgevoerd worden.
- Je kan de vormgeving van een website in één bestand kwijt. Hierdoor hoeft je niet alle pagina's afzonderlijk aan te passen.
- CSS wordt door alle browser ondersteund



Figuur 2.4: HTML met CSS

- Alle webpagina's kunnen er het zelfde uitzien.
- mogelijkheid om een website eenvoudig responsive te maken

jQuery is kleine maar snelle JavaScript bibliotheek. Het is ontworpen om de client-side scripting van de HTML pagina te versimpelen.

Een belangrijke jQuery functie is de `change()` functie. Hiermee kan je als je de waarde van een slider hebt aangepast, een functie uitvoeren. Bijvoorbeeld kan je de waarde van de slider met behulp van de functie `val()` weergegeven in een paragraaf. Een klein nadeel aan deze functie is dat deze pas wordt uitgevoerd als de slider losgelaten wordt. Dit wilt zeggen dat je eigenlijk een beetje moet gokken of de slider op de juiste positie staat. Een oplossing hier voor is de `on('input', ...)` functie. Hiermee zal de waarde die wordt weergegeven steeds aanpassen als de slider een beetje verschuift zonder deze los te laten. Nadat een juiste slider waarde is bekomen is het heel interessant omdeze waarde door te spelen naar de API hiervoor gebruiken we de AJAX functie. Hierbij moeten we enkel de url, methode en data meegeven en alles word simpel weg doorgestuurd.

```
$(document).on('input', '.slider', function(){
    $("#"+this.id+"Value").text($("#"+this.id).val());
})

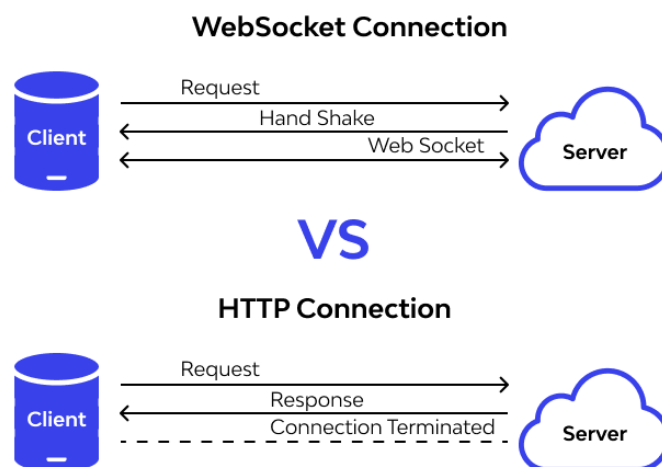
$("#mijn2eslider").change(function() {
    console.log("slider 2 aangepast naar " + $("#mijn2eslider").val());
    $("#mijn2ewaarde").text($("#mijn2eslider").val());
});

$.ajax({
    type: "POST",
    url: "post.php",
    data: {"First Name": vnm, "last name": anm},
    success: function (tekst) {
        $("#result").html(tekst);
    },
    error: function (request, error) {
        console.log ("ERROR:" + error);
    }
});
```

Flask is een eenvoudige raamwerk dat het mogelijk maakt een webapplicatie te bouwen in Python. In dit project gebruiken we het om de API op te bouwen.

4.1 WEBSOCKETS

Websockets is een netwerk protocol dat full-duplex communicatie via één enkele TCP verbinding biedt. Dit is een vorm van communicatie tussen een webbrowser en webserver. Websockets maakt het mogelijk voor een webpagina om automatisch upgedate te worden zonder een HTTP request te sturen. Tijdens de lessen hebben we gebruik gemaakt van de socketio module in flask. Hierdoor is de integratie van de websocket in de API probleemloos. Echter geeft deze oplossing één probleem. Deze module kan niet samenwerken met de asyncio module. Hierdoor ontstaat het probleem dat we met gebruik van socketio de waarden van de lampen van lokaal K4.04 niet kan opvragen. Na veel zoek werk was het resultaat dat het nooit ging werken. Eén oplossing was om pySocket te gebruiken of via een extra server te werken die via flask de data opvraagt en zo de data soorsturt.



Figuur 4.1: WebSocket