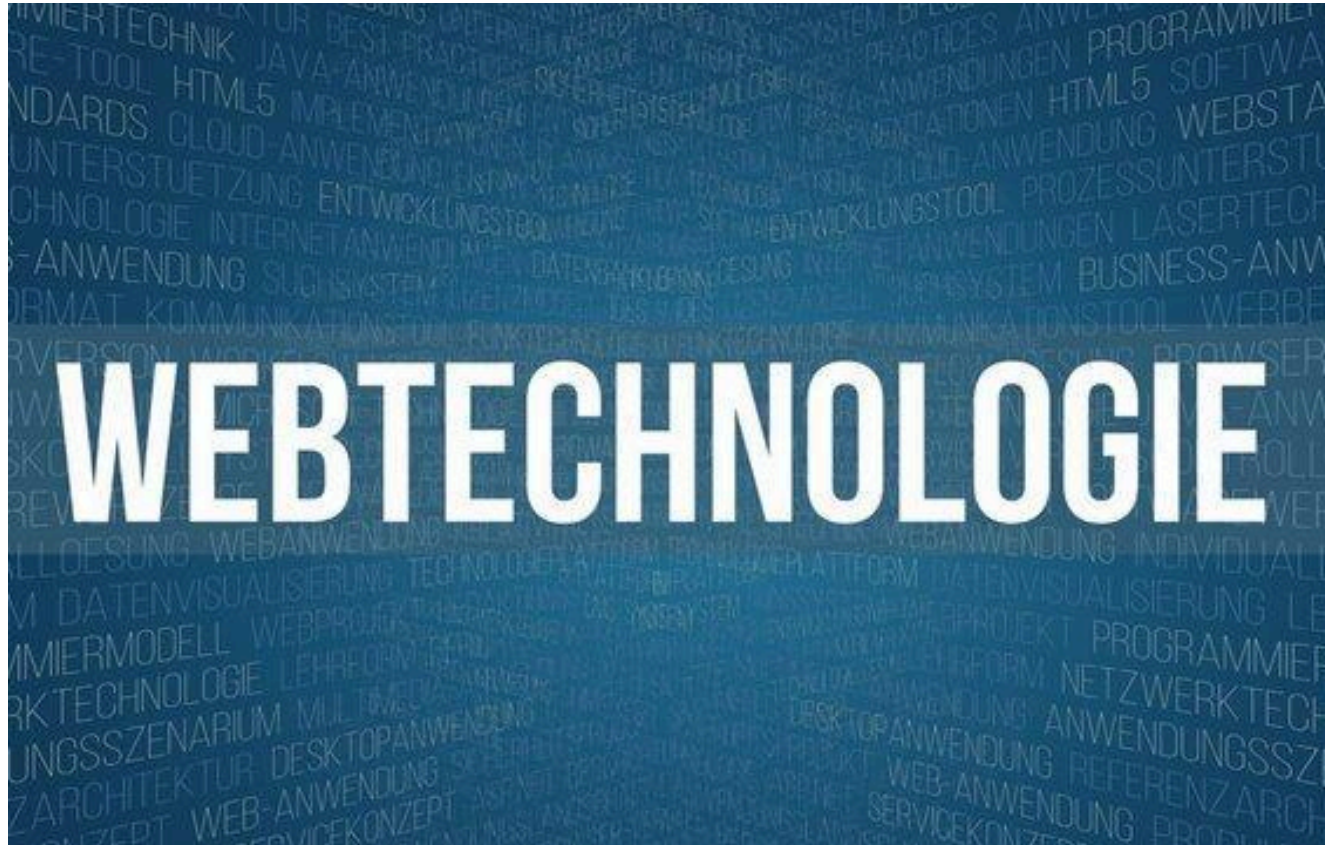


Projectdocumentatie – Webtechnologie



Van: Bram, Maciej en Timo

Inhoud









Wat is het project?	3
Wat zie je op onze github en wat betekent het?	3
Wat is .Gitignore	3
Wat doet een licentie?	3
Wat staat er allemaal in de .ReadMe file?	4
Wat is de Caddy File?	4
Wat is Index.hpp?	4
Wat is Login.php?	5
Wat is package-lock.json?	6
Wat is package.json?	6
Hoe zijn we te werk gegaan?	7
Opbouw van ons Project	9
Technische uitleg:	9
 1. Caddyfile	9
 2. index.php	10
 3. login.php (RESTful API)	10
 4. MySQL Database	10
 5. sensor.php & ESP32	10
 LOCALHOST als kern	11

Wat is het project?

Dit project is ontwikkeld in het kader van het vak Webtechnologie en omvat een volledige webapplicatie die bestaat uit een frontend, backend, RESTful API, en een PostgreSQL-database. De applicatie demonstreert een eenvoudige gebruikersauthenticatie met veilige opslag van gebruikersgegevens en communicatie tussen client en server via API-endpoints.

Wat zie je op onze github en wat betekent het?

Als je onze GitHub opent zie je dit:

 .gitignore	caddyfile root verandert voor mij pc
 License	Created License
 README.md	Update README.md
 caddyfile	Fixed The API login, index.php changed (removed login from...
 index.php	Fixed The API login, index.php changed (removed login from...
 login.php	Fixed The API login, index.php changed (removed login from...
 package-lock.json	added gitignore and ignored some files waht will clutter the ...
 package.json	added gitignore and ignored some files waht will clutter the ...

Wat betekent dit allemaal?

Wat is .GitIgnore

.gitignore is een bestand dat Git vertelt welke bestanden of mappen **niet** in de repository moeten komen, zoals tijdelijke bestanden, build-mappen (zoals dist/), of dependencies (node_modules/). Zo blijft je project netjes en zet je geen onnodige bestanden online.

Wat doet een licentie?

Een licentie bepaalt wat anderen wel of niet mogen doen met jouw code of website. Het geeft aan of iemand de code mag kopiëren, aanpassen of gebruiken, en onder welke voorwaarden. Zonder licentie mag niemand de code zomaar gebruiken. Met een open source licentie, zoals de MIT- of GPL-licentie, mag dat wel, maar vaak met regels zoals naamsvermelding of het verplicht delen van aangepaste versies. Zo zorgt een licentie ervoor dat de rechten en plichten rond jouw project duidelijk zijn.

Wat staat er allemaal in de .ReadMe file?

Een README-bestand is vaak het eerste bestand dat iemand ziet bij een project. Hierin leg je uit wat het project is, hoe het werkt en hoe je het kunt gebruiken.

Onze README bevat:

- Onze To-Do lijst
- Een korte beschrijving van het project
- Installatie-instructies

Zo helpt de README anderen (en jezelf) om snel te snappen wat het project doet en hoe ze ermee kunnen werken.

Wat is de Caddy File?

Een Caddyfile is een configuratiebestand voor de Caddy webserver. Hierin staat hoe de server jouw website of webapp moet draaien.

In een Caddyfile bepaal je bijvoorbeeld:

- Op welk domein of poort de site draait
- Of de site automatisch HTTPS krijgt
- Waar de bestanden van de website staan
- Eventueel: omleidingen (redirects), omgevingsvariabelen, reverse proxies, etc.

Waarom is dit belangrijk?

De Caddyfile zorgt ervoor dat je website veilig, correct en automatisch online komt. Caddy regelt hierdoor zelf HTTPS-certificaten, wat je bij andere servers vaak handmatig moet doen.

Wat is Index.php?

Het bestand index.php is vaak de startpagina van een PHP-website. Als iemand jouw site bezoekt, zoekt de server automatisch naar index.php om te tonen.

Wat doet het?

- Het start je website of webapplicatie
- Voert PHP-code uit om pagina's dynamisch te maken (bijv. met gebruikers, formulieren of databases)
- Laadt andere onderdelen, zoals scripts, templates of routes

Waarom is dit belangrijk?

Zonder index.php weet de server niet wat hij moet tonen. Het is dus de ingang van je site.

Wat staat er meestal in?

- HTML + PHP
- Logica voor routing of opbouw van de pagina
- Koppeling met een database
- Laad instructies voor CSS of JavaScript

Wat is Login.php?

Het bestand login.php wordt gebruikt om gebruikers te laten inloggen op een website.

Wat doet het?

- Controleert de ingevoerde gebruikersnaam en wachtwoord
- Vergelijkt deze gegevens met de data uit een database
- Start een sessie als de gegevens kloppen
- Geeft een foutmelding als het fout is

Waarom is dit belangrijk?

Zonder login.php kan je website geen beveiligde login aanbieden. Dit bestand is dus nodig om toegang tot afgeschermd delen van de site te regelen, zoals een dashboard.

Wat staat er meestal in?

PHP-code om gegevens uit een formulier (POST) op te halen

Connectie met een database

- Validatie van de invoer
- Start van een sessie (bij succes) of een foutmelding (bij mislukking)

Wat is package-lock.json?

Het bestand package-lock.json wordt automatisch aangemaakt door npm (Node Package Manager) als je afhankelijkheden (packages) installeert.

Wat doet het?

- Legt precies vast welke versies van alle packages (en hun sub-packages) zijn geïnstalleerd.
- Zorgt dat anderen exact dezelfde versie van de code krijgen als ze het project installeren.

Waarom is dit belangrijk?

Zonder package-lock.json kan het gebeuren dat iemand anders nieuwere (en mogelijk incompatibele) versies van packages binnenhaalt. Dit bestand zorgt dus voor betrouwbare en stabiele builds.

Wat staat erin?

- Namen en versies van alle packages
- Download-URL's
- Hashes (voor controle op fouten of manipulatie)

Je hoeft dit bestand meestal niet handmatig aan te passen — npm beheert het automatisch.

Wat is package.json?

Het bestand package.json is het hart van een Node.js-project. Hierin staat alle belangrijke informatie over je project en zijn afhankelijkheden.

Wat doet het?

- Beschrijft het project: naam, versie, beschrijving, auteur
- Lijst de nodige packages op (dependencies)
- Waarom is dit belangrijk?

Zonder package.json weet npm niet welke dependencies of scripts het project nodig heeft. Het maakt je project herbruikbaar en deelbaar met anderen.

Wat staat erin?

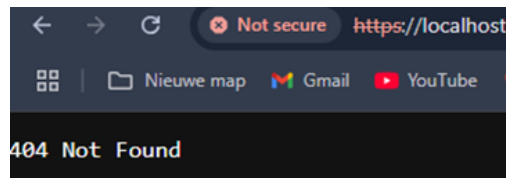
Projectinformatie (naam, versie)

Dependencies (zoals React, Express, etc.)

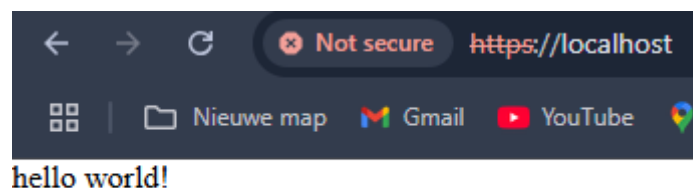
- Scripts (zoals npm start, npm run build)

Hoe zijn we te werk gegaan?

- 1) We zijn begonnen met het maken van een eenvoudige *Caddyfile* en hebben gecontroleerd of er iets zichtbaar was op localhost.
 - We zagen een "404 Not Found"-melding.
 - Dit betekent dat de pagina niet gevonden kon worden of nog niet bestond.



- 2) Vervolgens hebben we een eenvoudige index.php-bestand toegevoegd in dezelfde map als de *Caddyfile*, die een simpele "Hello World!" toont op localhost.
 - Deze werd correct weergegeven, dus dat werkte zoals verwacht.



- 3) Daarna hebben we beslist om een kader met timer erop te zetten die aftelt tot Bram zijn verjaardag. Hierbij hebben we ook gebruikgemaakt van HTML.
 - Dit hebben we gedaan omdat we iets dynamisch / bewegend wilden hebben.
- 4) Vervolgens hebben we een eenvoudige database opgezet met behulp van SQL, waarin we een gebruikersnaam en wachtwoord hebben opgeslagen.
- 5) Daarna hebben we de index.php aangepast zodat er een login formulier werd weergegeven.
 - Alleen wanneer je succesvol inlogde met de gegevens uit de database, kreeg je toegang tot de hoofdinhoud van de site met de timer.

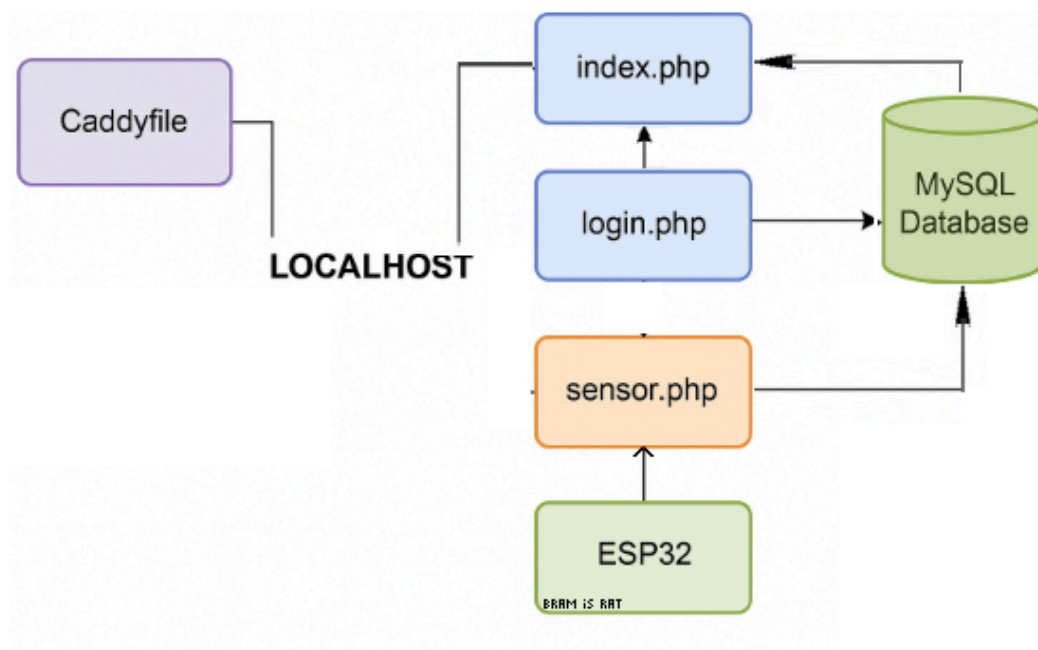
6) Vervolgens zijn we begonnen met het opzetten van een RESTful API. Deze hebben we in een apart PHP-bestand genaamd login.php geplaatst.

- Toen deze correct werkte, hebben we het login-gedeelte uit index.php gehaald en het volledige login-proces via login.php laten verlopen.

7)

Opbouw van ons Project

Hieronder is de globale architectuur van ons project weergegeven. Het schema laat zien hoe de verschillende onderdelen, zoals de ESP32, de temperatuur- en vochtigheidssensor, de PHP-scripts en de MySQL-database, met elkaar verbonden zijn. Deze structuur vormt de basis voor de communicatie tussen de hardware en de webserver.



Technische uitleg:

🔧 1. Caddyfile

- De Caddyfile fungeert als de configuratie voor de webserver (Caddy) op jullie localhost.
- Hij bepaalt hoe verzoeken naar de juiste bestanden (zoals index.php, login.php, ...) worden geleid.
- Alles wat je in je browser opent via localhost, wordt door de Caddyfile verwerkt.

2. index.php

- Dit is de hoofdpagina van de site.
- Aanvankelijk toonde dit bestand een eenvoudige "Hello World", later bevatte het ook:
 - Een aftellende timer tot Bram zijn verjaardag.
 - HTML-elementen voor layout.

De pagina is alleen toegankelijk nadat je succesvol bent ingelogd via de API.

3. login.php (RESTful API)

- Hier hebben jullie een RESTful API gemaakt in PHP.
- Het controleert gebruikersinvoer (gebruikersnaam & wachtwoord) en vergelijkt deze met de gegevens uit de database.
- Als de login klopt, stuurt deze een bevestiging terug die toegang verleent tot index.php.

4. MySQL Database

- De database bevat een tabel met gebruikersnamen en wachtwoorden.
- login.php maakt verbinding met deze database om logingegevens te verifiëren.
- Later kan hier ook data van de sensor in opgeslagen worden, indien gewenst.

5. sensor.php & ESP32

- Jullie hebben een ESP32 gebruikt in combinatie met een temperatuur- en luchtvochtigheidssensor.
- Deze maakt via sensor.php verbinding met de localhost op poort 8001.
- De gemeten data kan eventueel doorgegeven worden aan de database of zichtbaar gemaakt worden op de website.

LOCALHOST als kern

- Alle componenten draaien op jullie lokale server (localhost).

Zowel de webpagina als de communicatie met de sensor en database gebeurt binnen deze lokale omgeving.