|  |  |
| --- | --- |
|  | ELEKTRONICA-ICT  Opleidingsonderdeel 2023-2024 |

**Titel :** Web Technologies Project

|  |  |
| --- | --- |
|  | Emiel Mangelschots |

Inhoudsopgave

[1 Introductie 2](#_Toc55308666)

[2 Bestandsstructuur 3](#_Toc55308667)

[3 Implementatie en redenering 3](#_Toc55308668)

[4 Conclusie 4](#_Toc55308676)

# Abstract

Het "Web Technologies" project biedt studenten hands-on ervaring in het ontwikkelen en beheren van een volledige webapplicatie met moderne technologieën. Ze leren een webserveromgeving opzetten op een Ubuntu-machine, waarbij Caddy wordt geconfigureerd voor het veilig serveren van content. Het project omvat dynamische contentverwerking met PHP, databasebeheer met PostgreSQL, en frontend-ontwikkeling met HTML, CSS en JavaScript. Daarnaast ontwikkelen studenten een RESTful API voor communicatie tussen de frontend, server en externe apparaten. Het project benadrukt ook versiebeheer en samenwerking via Git/GitHub, en sluit af met uitgebreide documentatie en een presentatie van de resultaten.

# Introductie

In dit project richten we ons op het uitlezen van CPU-gegevens via een gesimuleerd script dat de principes van CRUD (Create, Read, Update, Delete) en RESTful architectuur volgt. Het script, dat de CPU-status in real-time monitort, is ontworpen om periodiek gegevens te genereren die vervolgens via een RESTful API naar een PostgreSQL-database worden gepusht. Deze aanpak stelt studenten in staat om praktische ervaring op te doen in het bouwen van API's, het verwerken van gegevens en het implementeren van veilige en efficiënte data-interactie tussen verschillende systeemcomponenten. Het project biedt een complete workflow van data-acquisitie tot en met opslag en presentatie, waardoor studenten inzicht krijgen in zowel server-side scripting als API-gebaseerde datamanagementprocessen.

# Bestandsstructuur

‘backend/’:

Bevat de backend-code, geschreven in Python, die de RESTful API implementeert. Deze API is verantwoordelijk voor het ontvangen, verwerken en opslaan van de CPU-gegevens in een PostgreSQL-database.

‘html/’:

Bevat de frontend-code, inclusief HTML-, CSS- en JavaScript-bestanden, die worden gebruikt om de verzamelde gegevens op een webinterface te presenteren. Deze bestanden zorgen voor een visueel aantrekkelijke en gebruiksvriendelijke weergave van de CPU-gegevens.

‘script\_values.py’:

Dit script simuleert het uitlezen van CPU-gegevens. Het genereert periodieke CPU-gegevens die via de RESTful API naar de backend worden gestuurd. Dit script speelt een cruciale rol bij het testen van de API-functionaliteit zonder dat toegang tot echte hardware vereist is.

‘Caddyfile’:

De configuratiefile voor de Caddy-webserver, die wordt gebruikt om de statische en dynamische content van de applicatie te serveren. Caddy biedt eenvoudig SSL-beheer en automatische HTTPS-implementatie, wat bijdraagt aan de beveiliging en betrouwbaarheid van de applicatie.

‘docker-compose.yml’:

Dit bestand definieert de Docker-containers die nodig zijn voor de ontwikkeling en implementatie van het project. Het maakt het mogelijk om de serveromgeving eenvoudig te reproduceren en zorgt voor consistentie tussen verschillende ontwikkel- en productieomgevingen.

# Implementatiestappen en redenering

Simulatie met script\_values.py:

Het gesimuleerde script biedt een gecontroleerde, consistente manier om CPU-gegevens te genereren. Dit is essentieel voor het testen van de API en het waarborgen van betrouwbare data-invoer zonder afhankelijk te zijn van echte hardware.

RESTful API Implementatie:

Door de RESTful architectuur te volgen, kunnen de verschillende componenten van de applicatie (frontend, backend, en externe scripts) efficiënt met elkaar communiceren. Dit zorgt voor schaalbaarheid en maakt het mogelijk om eenvoudig extra functionaliteit toe te voegen.

Gebruik van Docker en Caddy:

Docker zorgt voor een gestandaardiseerde ontwikkelomgeving, waardoor "works on my machine"-problemen worden geminimaliseerd. Caddy wordt gekozen vanwege zijn eenvoud en ingebouwde HTTPS-ondersteuning, wat bijdraagt aan de beveiliging en het beheer van de webserver.

Dataopslag in PostgreSQL:

PostgreSQL is een robuuste en veilige database die goed geschikt is voor het opslaan van gestructureerde gegevens zoals die gegenereerd door het script. Het biedt ook ingebouwde beveiligingsmaatregelen en schaalbaarheid, wat het ideaal maakt voor productieomgevingen.

Gebruik van een W3Schools Websitetemplate

Voor de frontend van dit project hebben we een websitetemplate van W3Schools gekozen. Deze keuze sluit aan bij het leerproces dat we gedurende het tweede semester hebben gevolgd, waarin we veel van onze HTML kennis via W3Schools hebben opgedaan. Door een template van deze bron te gebruiken, konden we niet alleen onze kennis verder verdiepen, maar ook snel een solide basis voor de website ontwikkelen.

# Conclusie

Dit project biedt een uitgebreide hands-on ervaring in het ontwerpen, implementeren en beheren van een webapplicatie die CPU-gegevens uitleest en verwerkt. Door gebruik te maken van moderne tools en methoden zoals Docker, Caddy, en RESTful API’s, bereidt het project studenten voor op de uitdagingen die zij in de echte wereld kunnen tegenkomen bij het ontwikkelen van webgebaseerde oplossingen.