|  |
| --- |
|  |
| PlantPal |
| Plant Management Systeem |

|  |
| --- |
| Sabawoon Enayat  2022-2023 |

# Het voorwoord

Ik ben Sabawoon Enayat en ik studeer Industriële ICT in Don Bosco Hoboken.

Mijn eindwerk is een plantmanagementsysteem. Ik heb dit onderwerp gekozen, omdat ik hou van planten en bloemen en als een programmeur wil ik altijd zo veel mogelijk automatiseren en op een makkelijke manier alles kunnen beheren.

Ik bedankt Dhr. K. van Reck en Dhr. L. Sienack om me te begeleiden met het eindwerk zowel op technische vlak, maar ook met de inhoud.

Inhoudsopgave

[Het voorwoord 1](#_Toc130210332)

[Inleiding 3](#_Toc130210333)

[Hardware 4](#_Toc130210334)

[Benodigdheden 4](#_Toc130210335)

[Werking 4](#_Toc130210336)

[Database 5](#_Toc130210337)

[PostgreSQL 5](#_Toc130210338)

[ORM 5](#_Toc130210339)

[Client 5](#_Toc130210340)

[WebAPI (Application Programming Interface) 6](#_Toc130210341)

[Python 6](#_Toc130210342)

[FastAPI 6](#_Toc130210343)

[Authenticatie 6](#_Toc130210344)

[Veiligheid 6](#_Toc130210345)

[Voorbeeld 6](#_Toc130210346)

[Web Interface 7](#_Toc130210347)

[TypeScript 7](#_Toc130210348)

[React.js 7](#_Toc130210349)

[Concepten 7](#_Toc130210350)

[Voorbeeld 7](#_Toc130210351)

[Bootstrap 8](#_Toc130210352)

[Voorbeeld 8](#_Toc130210353)

[Data fetching 9](#_Toc130210354)

[React Hook 9](#_Toc130210355)

[Besluit 10](#_Toc130210356)

[Bibliografie 11](#_Toc130210357)

# Inleiding

Mijn eindwerk is een plantmanagementsysteem. Dit wil zeggen dat een gebruiker zich kan registreren op de centrale website. Hij kan planten linken aan zijn profiel. Zo kan hij ook instellen op welke tijd de planten bewaterd worden en of dat het automatisch moet gebeuren. Hij kan in het dashboard ook zien hoe droog dat de grond van de plant is.

Er kan ook een grafiek bekeken worden van hoeveel water een plant heeft gekregen. Zo kunnen ook planten vergeleken worden.

Voor mijn eindwerk maak ik gebruik van heel wat technologieën om alles voor de “end-user”, zo makkelijk en beveiligd mogelijk te maken. Dit gebeurt door de “userinterface” en de rest van de applicatie te scheiden.

Voor de UI (userinterface) heb ik React.js met TypeScript gebruikt. React.js heb ik gebruikt omdat het alles versimpelt voor de ontwikkelaar en omdat er heel veel documentatie errond is. TypeScript omdat het strikt is en omdat het veel problemen elimineert.

Voor de API (Application Programming Interface, wat ik bedoelde met de “rest van de applicatie”), heb ik FastAPI gebruikt. De authenticatie gebeurt door JWT-tokens.

Ik heb PostgreSQL als database gebruikt, omdat ik ermee vertrouwd was en omdat er veel documentatie is.

Voor de ORM (Object Relation Mapping, de middleware tussen de API en de database), heb ik Prisma gebruikt, omdat het alles versimpelt.

Voor de bewatering van de planten, heb ik een ESP32 gebruikt. De ESP32 heeft ook een ingebouwde Wifi module, die in AP (Access Point) mode, STA (Station) mode, of beide modes gecombineerd kan zijn.

# Hardware

## Benodigdheden

* Waterpomp
* Debietmeter
* Vochtigheidsmeter voor grond
* Relais
* LEDs
* Knoppen

## Werking

# Database

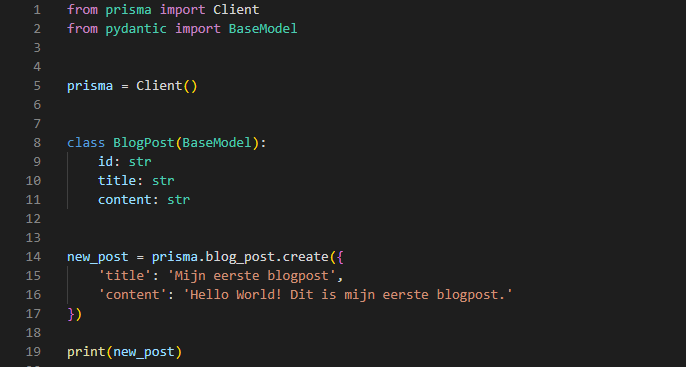
## PostgreSQL

PostgreSQL is een krachtige en veelzijdige database die ontwikkelaars kunnen gebruiken om hun applicaties te ondersteunen. Het is betrouwbaar en heeft veel handige functies, zoals ondersteuning voor complexe zoekopdrachten, ruimtelijke gegevens en tekst doorzoeken. PostgreSQL werkt op verschillende systemen en kan gebruikt worden met verschillende programmeertalen. Bovendien is het gratis en open-source, wat betekent dat iedereen het kan gebruiken en aanpassen. Dit maakt PostgreSQL een aantrekkelijke optie voor ontwikkelaars die op zoek zijn naar een krachtige, betrouwbare en flexibele database-oplossing.

## ORM: Prisma

Een ORM (Object-Relational Mapping) is een tool die ontwikkelaars kunnen gebruiken om te communiceren met hun database zonder SQL te schrijven. Prisma is een populair ORM dat werkt met Python en verschillende databases, waaronder PostgreSQL en MySQL. Prisma zorgt ervoor dat ontwikkelaars gemakkelijk objecten kunnen maken en ophalen vanuit hun database.

### Voorbeeld



# WebAPI (Application Programming Interface)

Dit is het gedeelte van de applicatie die voor de authenticatie en beheer van data zorgt. Het is in essentie een interface tussen de database en de toepassing. De programmeertaal die ik gebruikt heb voor de WebAPI, is Python. De web framework die ik gebruikt heb is FastAPI.

## Python

Python heb ik voor de volgende redenen gebruikt:

* Eenvoudig gebruik
* Veel documentatie

## FastAPI

FastAPI is een framework voor het bouwen van webapplicaties in de programmeertaal Python. Het heeft verschillende voordelen, waaronder:

1. Snelheid: FastAPI is erg snel en efficiënt, waardoor het ideaal is voor het bouwen van snelle en responsieve webapplicaties.
2. Eenvoudig te gebruiken: FastAPI heeft een eenvoudige en duidelijke documentatie, wat het gemakkelijk maakt om te leren en te gebruiken, zelfs als je geen ervaring hebt met webontwikkeling.
3. Asynchrone ondersteuning: FastAPI ondersteunt asynchrone programmering, wat betekent dat het meerdere taken tegelijk kan uitvoeren en zo de prestaties van de applicatie kan verbeteren.
4. Type hints: FastAPI maakt gebruik van type hints, waarmee ontwikkelaars het type van variabelen en functieparameters kunnen aangeven. Dit zorgt voor betere codekwaliteit en voorkomt fouten in de code.
5. Automatische documentatie: FastAPI genereert automatisch documentatie voor de API's die je bouwt, wat het gemakkelijker maakt om de API te begrijpen en te gebruiken voor anderen die ermee werken.

Kortom, FastAPI is een krachtig framework dat snel, eenvoudig te gebruiken, asynchroon en goed gedocumenteerd is. Het is daarom een goede keuze voor het bouwen van moderne webapplicaties met Python.

### Authenticatie

FastAPI JWT (JSON Web Token) authenticatie is een manier om gebruikers te authentiseren (inloggen) op een webapplicatie met behulp van JWT's. JWT's zijn stukjes informatie die worden gebruikt om de identiteit van een gebruiker te verifiëren.

De voordelen van FastAPI JWT authenticatie zijn onder andere:

1. Veiligheid: Met JWT's kun je veilig en gemakkelijk gebruikersgegevens versleutelen en overdragen tussen client en server, waardoor de veiligheid van de applicatie wordt verbeterd.
2. Eenvoudig te gebruiken: FastAPI biedt een eenvoudige manier om JWT-authenticatie in te stellen en te gebruiken, zelfs als je weinig ervaring hebt met authenticatie of beveiliging.
3. Flexibiliteit: Met JWT's kun je aangepaste claims (metadata) toevoegen aan de authenticatietokens, waardoor je extra informatie kunt opslaan over de gebruiker, zoals hun rol of toegangsrechten.
4. Efficiëntie: Met JWT's hoef je geen gebruikersgegevens op te slaan op de server, waardoor je server efficiënter wordt en minder resources nodig heeft.

Kortom, FastAPI JWT authenticatie biedt een veilige, eenvoudige en schaalbare manier om gebruikers te authentiseren en te autoriseren op een webapplicatie. Het is daarom een goede keuze voor het bouwen van moderne, veilige webapplicaties met Python.

### Veiligheid

FastAPI en Pydantic zijn frameworks die zijn ontworpen om veiligheid te waarborgen in webapplicaties. FastAPI biedt ingebouwde beveiligingsmaatregelen om bescherming te bieden tegen verschillende soorten aanvallen, terwijl Pydantic automatisch inputdata valideert om het risico op kwetsbaarheden te verminderen. Beide frameworks zijn ontworpen om te voldoen aan de nieuwste veiligheidsnormen en -praktijken, wat betekent dat het gebruik van deze frameworks samen zorgt voor een sterke beveiligingslaag in je webapplicatie.

### Voorbeeld

# Web Interface

Voor de web-interface heb ik, TypeScript, React.js en bootstrap gebruikt. HET platform waarop dit gemaakt wordt is Node.js.

## TypeScript

TypeScript is een programmeertaal die, net als JavaScript, gebruikt wordt voor het bouwen van websites en webapplicaties. Het grote voordeel van TypeScript ten opzichte van JavaScript is dat het een type-systeem heeft. Dit type-systeem maakt het mogelijk om fouten in de code te voorkomen voordat de code daadwerkelijk wordt uitgevoerd. Het type-systeem controleert namelijk of de verschillende delen van de code correct met elkaar zijn verbonden. Dit leidt tot meer betrouwbare code en minder problemen in het gebruik van de applicatie.

Een ander voordeel van TypeScript is dat het de programmeur helpt bij het schrijven van begrijpelijke en goed gedocumenteerde code. TypeScript maakt het namelijk mogelijk om documentatie te schrijven die de functies van verschillende delen van de code uitlegt. Dit zorgt ervoor dat andere programmeurs de code gemakkelijker kunnen begrijpen en dat de code gemakkelijker te onderhouden is.

Daarnaast biedt TypeScript nog andere handige functies, zoals bijvoorbeeld de mogelijkheid om klassen en interfaces te definiëren. Dit maakt het schrijven van complexere applicaties veel makkelijker en overzichtelijker.

Kortom, TypeScript biedt veel voordelen voor ontwikkelaars van websites en webapplicaties. Het zorgt voor meer betrouwbare en onderhoudbare code, waardoor het gemakkelijker wordt om samen te werken aan een project.

## File:React-icon.svg - Wikimedia CommonsReact.js

### React.js is een populaire JavaScript-bibliotheek die wordt gebruikt om dynamische gebruikersinterfaces te bouwen voor webapplicaties. Het belangrijkste voordeel van React.js is dat het de ontwikkeling van complexe gebruikersinterfaces gemakkelijker maakt door het gebruik van componenten.

### Componenten zijn kleine, herbruikbare stukjes code die specifieke functionaliteit bieden en kunnen worden gecombineerd om complexere interfaces te maken. Dit maakt het gemakkelijk om code te organiseren en te hergebruiken, wat tijd en moeite bespaart tijdens de ontwikkeling.

### Een ander voordeel van React.js is dat het een efficiënte manier biedt om webpagina's dynamisch te updaten zonder de hele pagina opnieuw te laden. Dit zorgt voor een snellere en betere gebruikerservaring.

### Ten slotte heeft React.js een grote en actieve community van ontwikkelaars die bijdragen aan de ontwikkeling van nieuwe functies, bugfixes en documentatie, waardoor het een betrouwbare en up-to-date bibliotheek is om te gebruiken in webontwikkelingsprojecten.

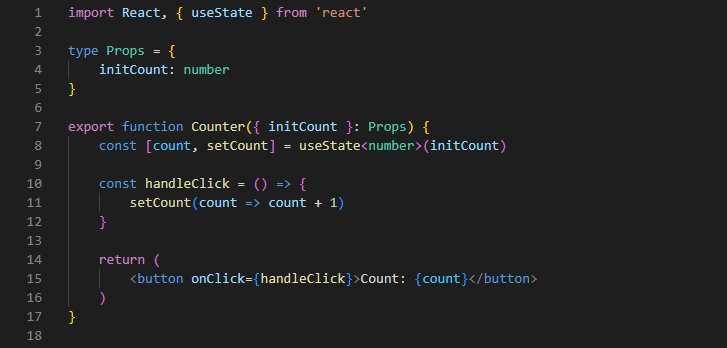
TSX (TypeScript XML) is een uitbreiding van TypeScript die wordt gebruikt in combinatie met React.js om JSX-syntax te ondersteunen. JSX staat voor JavaScript XML en is een syntaxisuitbreiding voor JavaScript die wordt gebruikt om React-componenten te maken.

Met JSX kan HTML-achtige code worden geschreven in JavaScript om gebruikersinterfaces te definiëren. Dit maakt het gemakkelijker om React-componenten te maken en de code leesbaarder te maken, omdat het lijkt op het schrijven van HTML-code.

### Concepten

1. Hooks: Hooks zijn functies in React.js die de mogelijkheid bieden om state en andere React-functies te gebruiken in functionele componenten. Hooks stellen ontwikkelaars in staat om complexe logica te schrijven in functionele componenten, wat het schrijven van code eenvoudiger maakt en de leesbaarheid verbetert.
2. Context: Context is een manier om gegevens door te geven tussen componenten in een React-applicatie zonder de gegevens expliciet te moeten doorgeven via props. Context kan worden gebruikt om thema's, gebruikersgegevens en andere gegevens te delen tussen componenten.
3. Props: Props is een afkorting voor "properties" en worden gebruikt om gegevens door te geven aan componenten. Props kunnen strings, objecten, functies en andere waarden bevatten en stellen componenten in staat om te communiceren met andere componenten in een React-applicatie.
4. State: State is een manier om gegevens op te slaan in een component. Wanneer de staat van een component verandert, wordt de component opnieuw gerenderd om de wijzigingen weer te geven. State kan worden gebruikt om gegevens op te slaan die veranderen tijdens de levensduur van een component.

### Voorbeeld



## File:Bootstrap logo.svg - Wikimedia CommonsBootstrap

Bootstrap is een styling bibliotheek, die vaak gebruikt wordt. Het zorgt voor de basis CSS-classes, zodat we de styling niet moeten schrijven, maar gewoon gebruiken. Er bestaan meer geavanceerde styling bibliotheken, zoals, mui, maar ik wou de styling simpel houden.

### Voorbeeld

## Data fetching

# Besluit

# Bibliografie