

Inleiding

In opdracht van het HAN Biocentre wordt de microflora in de compost van de champignonenteelt in kaart gebracht. Door de samenstelling van de microflora in kaart te brengen kan onderzocht worden hoe de compost kan worden verbeterd.

In de technische documentatie wordt beschreven hoe de applicatie is gemaakt en hoe hij gebruikt kan worden.

De applicatie is gemaakt voor de bioloog, om op een overzichtelijke manier te kunnen zien welke organismen in de compost gevonden zijn. De applicatie bevat een taartdiagram, waarin de vijf meest voorkomende organismen gevonden kunnen worden. Verder maakt de website een connectie met de database, om de gegevens uit de database overzichtelijk in een tabel weer te kunnen geven. De applicatie is te vinden via de link

<https://hannl-hlo-bi1a1-app.azurewebsites.net/>

Werking

Als men op de site komt, dan is figuur 1 de eerste pagina die zij zien.



Fig. 1. De homepage van de website. Dit is de eerste pagina die de gebruiker te zien krijgt.

Op deze pagina staat een uitleg over wat er op elke pagina te vinden is. Door de navigatiebalk kan de gebruiker makkelijk navigeren naar andere pagina's. Door het duidelijke en simpele design krijg je snel overzicht van wat belangrijk is. Door de kop, navigatiebalk en de footer een kleur te geven springt het eruit. De knoppen van de navigatiebalk hebben schaduw, zodat het duidelijk is dat de gebruiker erop kan klikken. Doordat de knop die naar de pagina navigeert die op dat moment actief is een witte kleur te geven, is het duidelijk voor de gebruiker op welke pagina hij/zij op dat moment is. Als de gebruiker op de 'over het project' knop klikt, dan komt hij/zij op de pagina die hieronder wordt weergegeven in figuur 2.



Fig. 2. De Over dit project pagina van de website. Hier staat meer informatie op over het project.

Als de gebruiker op de knop 'Statistieken' klikt, komt de gebruiker op de statistieken pagina.



Fig.3. De statistieken pagina met het taartdiagram met de vijf meest voorkomende organismen

Op de pagina statistieken kan je een taartdiagram vinden met de 5 meest voorkomende organismen. Door met de muis over de piechart te gaan, vind je meer informatie over het aantal organismen.

Op de homepage is er een tabel te vinden met de gegevens uit de database. De tabel is te zien in figuur 4. Dit zijn al onze blast resultaten.

resultaat id	accessiecode	e value	perc. id	total coverage	Query coverage	max scores	sequentie	seq. id	eiwit id	geslachtsnaam
195	CP002630.1	1.53459e-06	None	None	None	None		1	None	None
196	CP029344.1	None	None	None	None	None		1	None	None
197	AP010904.1	8.11738e-35	None	None	None	None		1	None	None
198	CP027022.1	0.00137754	None	None	None	None		1	None	None
199	CP001699.1	9.25806e-33	None	None	None	None		1	None	None
200	CP026265.1	1.30963e-08	None	None	None	None		1	None	None
201	LT607411.1	0.0387538	None	None	None	None		1	None	None
202	LR584432.1	1.30971	None	None	None	None		1	None	None
203	None	None	None	None	None	None		1	None	None
204	CP040020.1	0.000857235	None	None	None	None		1	None	None
205	LT550419.1	0.00090559	None	None	None	None		4	None	None

Groep EENhoorn bestaat uit vier groepsleden, die studeren aan de Hogeschool van Arnhem en Nijmegen. Deze site is gemaakt om de gevonden organismen en eiwitten uit compost op een overzichtelijke manier te laten zien aan biologen. Op deze site is tevens de functie van één van de eiwitten te vinden. De organismen en eiwitten zijn gevonden door middel van BLAST van NCBI. De groep heeft een geautomatiseerde BLAST tool ontwikkeld. De tool en uitkomsten zijn voor het laatst geüpdatet op

Fig.4. De tabel met de gegevens uit de database. De tabel is scrollbaar, dit betekent dat je meer informatie kan zien als je door de tabel heen scrollt.

Op de zelf blasten pagina, te zien in figuur 5, kan je zelf een BLAST uitvoeren door een sequentie in te voeren en een database te kiezen. Dit werkt alleen niet op de azure server, omdat er geen XML bestand aangemaakt kan worden op de server.

Fig.5. De zelf blasten pagina. Hier kan de gebruiker zelf een blast uitvoeren.

Als laatste is er nog de FAQ pagina. Deze is te zien in figuur 6. Op deze pagina staan veelgestelde vragen.

Fig.6. De Frequently Asked Questions pagina. Deze pagina bevat veelgestelde vragen.

Software architectuur

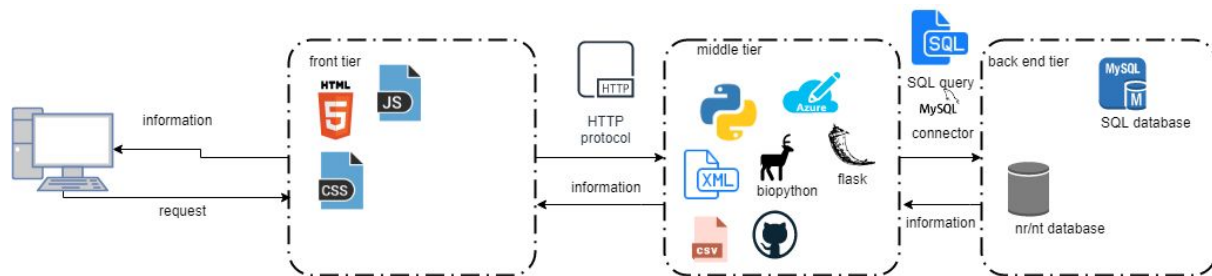


Fig. 7 tier applicatie architectuur. Dit is een client-server architectuur die bestaat uit de front tier, middle tier en backend tier. Ze worden ook wel presentation tier, application tier en data tier genoemd.

Het figuur hierboven (figuur 7) laat zien bij welke tier de gebruikte talen, bestanden, modules en databases horen. De gebruiker kan een request doen om bijvoorbeeld een webpagina op te vragen. Vervolgens wordt er door middel van de talen HTML5, JavaScript en CSS3 bepaald hoe de informatie moet worden weergegeven. De talen maken onderdeel uit van de front tier. De talen uit de front tier communiceren met de middle tier door middel van een HTTP (Hypertext Transfer Protocol) protocol. In de middle tier staat bijvoorbeeld de programmeertaal Python (Python 3.6+). De middle tier ondersteunt de belangrijkste functies van de applicatie. Dit is de reden waarom in de middle tier ook biopython(1.73) te vinden is. Door de modules Flask (1.0.2) en jinja(2.10.1) van python kan python gebruik maken van html5. De middle tier kan communiceren met de back end tier door middel van SQL queries. Hierdoor kan er informatie van de database worden opgevraagd en de informatie kan via de middle tier naar de front tier (die bepaalt hoe de informatie moet worden weergegeven) en de gebruiker krijgt de webpagina met informatie te zien. Om ervoor te zorgen dat het script een connectie kan maken met de database, moet er een sql-connector (8.0.11) worden geïnstalleerd. De site draait door middel van Azure.

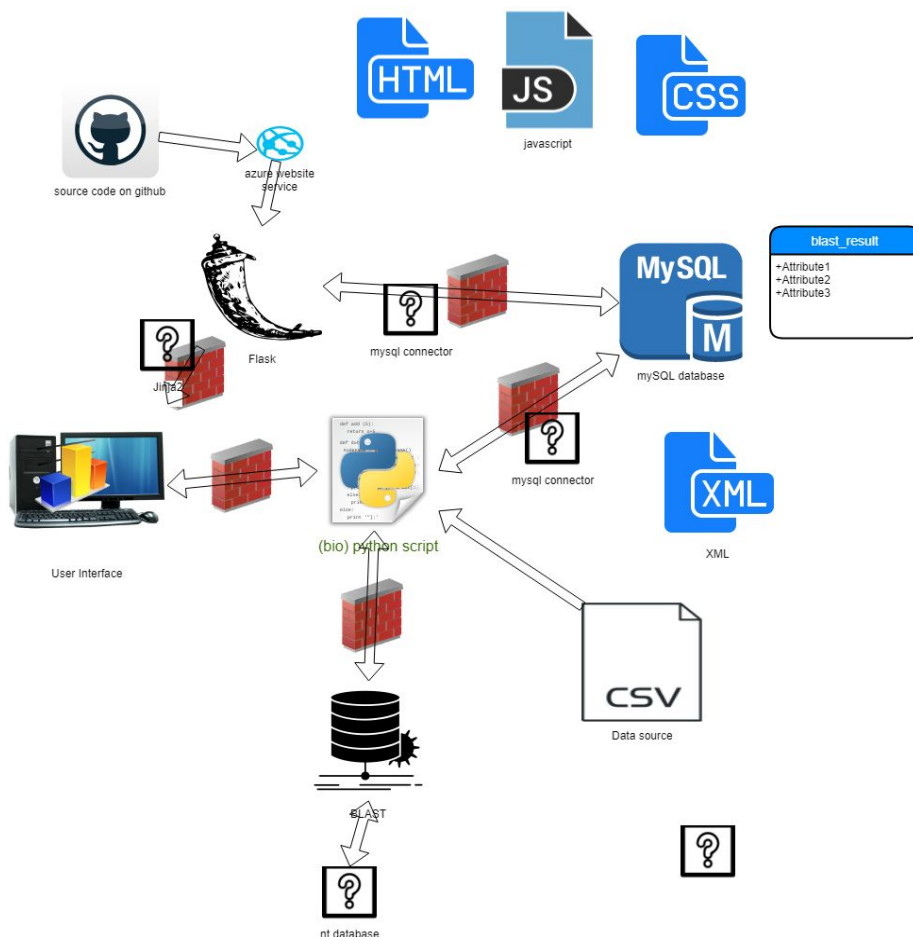


Fig. 8. Laat de onderlinge relatie zien tussen de modules en gebruikte talen.

Figuur 8 laat de onderlinge relaties tussen de gebruikte modules en de talen zien. Zo is te zien dat de code van github gebruikt wordt voor azure. Azure maakt gebruik van Flask, Flask maakt weer gebruik van jinja om de HTML pagina's, CSS en Python aan elkaar te linken. Zoals te zien is in figuur 8, staat het python script centraal. In het python script kan door middel van de mysql connector een connectie worden gelegd met de mySQL database. Het python script maakt gebruik van blast en verkrijgt daardoor informatie uit de nt database. Het python bestand maakt ook gebruik van een csv bestand als de data source.

De modules die moeten worden gedownload voor de werking van het script zijn biopython en flask.

Indien de modules nog niet zijn gedownload, dan kunnen de modules op een eenvoudige manier worden gedownload in PyCharm. Voor het runnen van de code is er ook gebruik gemaakt van PyCharm als IDE. Als de code wordt geopend door middel van PyCharm wordt er met rode strepen onder de nog niet geïnstalleerde modules weergegeven welke er nog moeten worden geïnstalleerd. Door linksboven op 'File' en dan op 'Settings' te klikken kom je bij de project interpreter. Hier wordt getoond welke modules er zijn geïnstalleerd. Onder het tandwielje staat een + teken. Als daarop wordt geklikt krijg je een venster waar alle modules in staan. Tik in de zoekbalk de module die nog moet worden gedownload, klik de module aan en druk onderaan het venster op 'install package'.

Er zijn in het script van de applicatie meerdere functies te vinden, dit heeft te maken met de route van de app. Telkens als er wat anders achter de

<https://hannl-hlo-bi1a1-app.azurewebsites.net/> staat, returnen we een ander

render_template. Als het alleen '/' is, returnen we de index.html pagina.

Er is ook een functie die formpje heet. De functie maakt gebruik van een html pagina waar een form op staat. In het form staat een invoerveld waar de gebruiker een sequentie kan invoeren. De sequentie wordt dan in hoofdletters omgezet, zodat er kan worden geblast.

Door het gebruik van biopython wordt er een blast uitgevoerd. De uitkomsten daarvan worden in een XML file geschreven. Vervolgens wordt het XML file geopend en gelezen. De titels worden dan gereturned en weergegeven op de pagina.

Als de root '/index' is, returnen we ook de index.html pagina.

Als de root '/about' is, dan returnen we de pagina waar meer informatie te vinden is over het project.

Als de root '/FAQ' is dan wordt de pagina gereturned met de vragen die vaak gesteld worden.

Als de root '/statistieken' is, dan wordt de statistieken.html pagina gereturned.

Als de root '/zelf_blasten' is, dan wordt de pagina zelf_blasten.html gereturend.

Als de root '/selectdemo' is, dan wordt de pagina getoond met de tabel die gegevens bevat uit onze database. Daarvoor wordt een connectie gemaakt met de database door de DbConnector. De functie createHtmlTable(rows) maakt de tabel waar de uitkomsten van de database in staan. Door de table head een tabel te geven en de table body een tabel te geven kon er een tabel worden gemaakt die je kan scrollen.

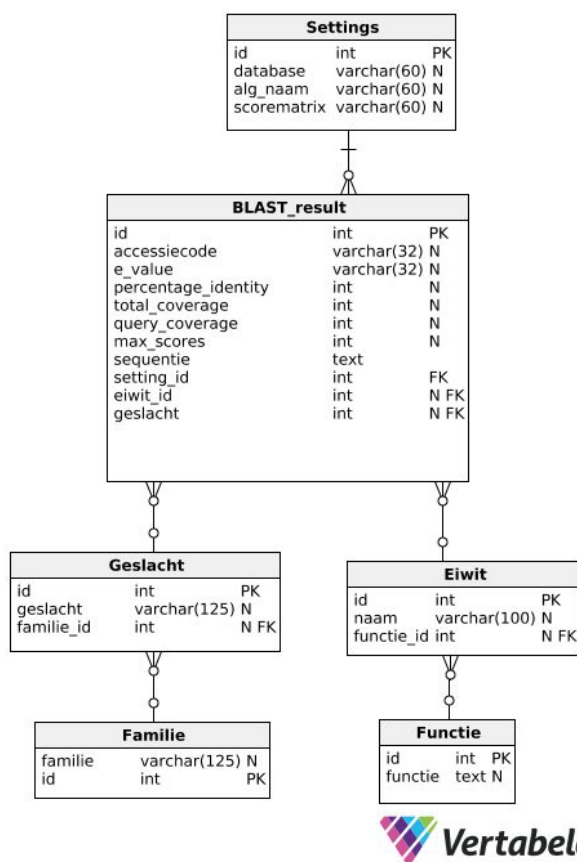


Fig. 9. Het Entity-Relationship Diagram van de database waarin de tabellen en kolommen weergegeven staan. Hierin kunnen BLAST resultaten worden opgeslagen om later te voorzien van annotatie.

In figuur 9 is het ERD te zien. Het ERD bestaat uit zes tabellen. In de tabel BLAST_result zijn de belangrijkste resultaten die uit de BLAST komen opgeslagen. Verder wordt ook de sequentie in BLAST_result opgeslagen. De tabel BLAST_result is verbonden met de tabellen Settings, Geslacht en Eiwit. In de tabel Settings worden de instellingen opgenomen die zijn gebruikt voor het blasten. Tussen de tabellen BLAST_result en Settings zit een 1 op veel relatie. Dit is omdat een BLAST resultaat 1 bepaalde combinatie aan settings heeft, maar die combinatie bij meerdere BLAST resultaten gebruikt kan worden. In de Geslacht tabel wordt de geslachtsnaam van het organisme opgeslagen. De relatie tussen BLAST_result en Geslacht is weer een 1 op veel relatie, omdat een geslachtsnaam bij meerdere BLAST resultaten kan horen, maar een BLAST resultaat maar 1 geslachtsnaam heeft. In de Eiwit tabel wordt de naam van het eiwit opgeslagen. Tussen Eiwit en BLAST_result zit weer een 1 op veel relatie, omdat er maar 1 eiwit naam per BLAST wordt opgeslagen, maar die eiwit naam wel meerdere keren voor kan komen. De tabel Geslacht is verder nog verbonden aan de tabel Familie. In de Familie tabel wordt de familienaam van het organisme dat uit de BLAST komt opgeslagen. De relatie tussen Geslacht en Familie is 1 op veel, omdat familie boven geslacht staat. Dus een familie heeft meerdere geslachten, maar een geslacht heeft maar 1 familie. Verder is er nog de Functie tabel. In die tabel wordt de functie van het eiwit opgeslagen. Die tabel is met een 1 op veel relatie aan de Eiwit tabel verbonden, omdat we er voor gekozen hebben om maar 1 functie per eiwit op te slaan, maar die functie wel meerdere keren voor kan komen. In alle tabellen is ook een id te vinden. Dit is omdat het dan makkelijker is om de tabellen aan elkaar te koppelen. Alle relaties zijn niet verplicht, omdat we niet altijd alle tabellen konden vullen met wat er uit de BLAST kwam. Alleen BLAST_result moet settings hebben, aangezien er altijd bepaalde settings zijn gebruikt voor het BLASTen.

Testscripts

Website: Database ophalen/tonen


Stap #	Instructie	Verwacht resultaat	Feitelijk resultaat	Opmerking (niet verplicht)
1	klik op < database knop >	naar de database pagina	niet geïmplementeerd	
2	klik op < submit dna > (moet nog hernoemd worden)	Database tabel komt gevuld tevoorschijn	werkt	

Website: Grafiek tonen

Stap #	Instructie	Verwacht resultaat	Feitelijk resultaat	Opmerking (niet verplicht)

1	Klik op statistieken	Gaat naar de pagina statistieken en toont de grafiek	werkt	
2	Ga met je muis over een blokje van de grafiek	Laat zien welk organisme bij het blokje hoort en hoe vaak hij voorkomt	werkt	

Website: Zelf Blasten

Stap #	Instructie	Verwacht resultaat	Feitelijk resultaat	Opmerking (niet verplicht)
1	Klik op 	laadt de zelf_blasten pagina https://hannl-hlo-bi1a1-app.azurewebsites.net/zelf_blasten	werkt	
2	voer sequentie in	sequentie wordt ingevoerd	werkt	
3	klik op submit	Blast wordt uitgevoerd	werkt niet	Werkt alleen niet op de Azure server, omdat er geen bestand aangemaakt kan worden

Biopython: data automatisch blasten

Stap #	Instructie	Verwacht resultaat	Feitelijk resultaat	Opmerking (niet verplicht)
1	Clone de git repository om de code te bemachtigen om het script te runnen.	Je krijgt de code op je computer.	werkt	
2	open de gedownloade folder in een IDE (zoals PyCharm).	Alle bestanden van het project	werkt	

		staan in de IDE.		
3	Download de benodigde modules (als je die nog niet hebt geïnstalleerd).	De modules worden gedownload	werkt	
4	Run het script	Blast wordt uitgevoerd	werkt	

Bijlage

- Verwijzing naar ontwikkelde software (link naar github)
<https://github.com/Queuebee2/Course4Tutor>
- Benodigde wachtwoorden

Bronnen

citatie (kort)	complete APA citatie	waarvoor?
		Biopython 1.73
		mySQL (connector) mysql-connector-python 8.0.11 of mysql-connector 2.2.9
		Pickle
		Python 3.6+
		Jinja 2.10.1
		Flask 1.0.2
		Microsoft Azure - github host
		Github
		Microsoft Azure mySQL database

		Windows (?)
		HTML
		CSS
		Javascript
		Google Chrome
		Mozilla Firefox
		Internet explorer