20-3-2025

Eleonora Avaleva 20220545

De Haagse HOgeSchool Den Haag

Logboek

Web-applications 2025

# Inhoudsopgave

Inhoud

[Inhoudsopgave 1](#_Toc193571374)

[LOGBOEK FRONT-END 2](#_Toc193571375)

[Inleiding – front-end 2](#_Toc193571376)

[Leerdoelen 2](#_Toc193571377)

[Activiteiten front-end 2](#_Toc193571378)

[**Stap 1: Analyse en planning (Week 1)** 2](#_Toc193571379)

[**Stap 2: Wireframing en UI design (Week 1)** 3](#_Toc193571380)

[**Stap 3: HTML en CSS-basisstructuur (Week 1-2)** 3](#_Toc193571381)

[**Stap 4: JavaScript functionaliteit implementeren (Week 2)** 3](#_Toc193571382)

[**Stap 5: Testen en code review (Week 3)** 4](#_Toc193571383)

[Reflectie front-end 4](#_Toc193571384)

[LOGBOEK BACK-END 6](#_Toc193571385)

[Inleiding back-end 6](#_Toc193571386)

[Leerdoelen formuleren 6](#_Toc193571387)

[Activiteiten back-end 6](#_Toc193571388)

[**Stap 1: Analyse en planning van de API-structuur (Week 1)** 7](#_Toc193571389)

[**Stap 2: Opzetten ontwikkelomgeving en bestuderen tutorial’s (Week 1)** 7](#_Toc193571390)

[**Stap 4: Testen en documenteren met Postman (Week 2)** 7](#_Toc193571391)

[**Stap 5: Code review met studiegenoten (Week 3)** 8](#_Toc193571392)

[**Stap 6: Verbinden van frontend en backend (Week 3)** 8](#_Toc193571393)

[**Stap 7: Deployment naar Azure (Week 3)** 8](#_Toc193571394)

[Reflectie back-end 8](#_Toc193571395)

[LOGBOEK EINDPRODUCT 11](#_Toc193571396)

[Inleiding 11](#_Toc193571397)

[Leerdoelen formuleren: 11](#_Toc193571398)

[Activiteiten 11](#_Toc193571399)

[Reflectie 11](#_Toc193571400)

# LOGBOEK FRONT-END

## Inleiding – front-end

"My Favorite Books" webapplicatie voor de individuele opdracht Front-end. Het doel van dit logboek is om inzicht te geven in mijn aanpak, de gemaakte keuzes te onderbouwen, en mijn persoonlijke ontwikkeling tijdens dit opdracht te reflecteren.

Het logboek is opgebouwd uit drie hoofddelen:

1. Leerdoelen en uitgangspositie
2. Uitgevoerde activiteiten en gemaakte keuzes
3. Reflectie op het proces en de resultaten

Bij aanvang van dit semester had ik al enige ervaring met front-end ontwikkeling door anderhalf jaar werkervaring als Front-End ontwikkelaar, waar ik met JavaScript, React en MaterialUI heb gewerkt. Daarnaast heb ik de cursussen Frontend Engineering en Fullstack Engineering via Codecademy gevolgd. Voor deze opdracht was de uitdaging vooral om terug te gaan naar de basis en een applicatie te bouwen met alleen vanilla JavaScript, HTML en CSS, zonder gebruik te maken van frameworks of bibliotheken. Dit gaf me de kans om mijn fundamentele kennis te versterken en mijn begrip van core web technologieën te verdiepen.

## Leerdoelen

Gebaseerd op de beroepstaken uit het curriculum heb ik de volgende leerdoelen geformuleerd:

**Leerdoel 1: Realiseren van software - Versterken van vanilla JavaScript vaardigheden**  
Ondanks mijn ervaring met JavaScript in een React context, wil ik binnen 3 weken mijn vanilla JavaScript vaardigheden verbeteren om interactieve elementen (paginering, sortering en filtering) te kunnen implementeren zonder frameworks of bibliotheken. Ik meet dit door het succesvol implementeren van alle vereiste functionaliteiten in de "My Favorite Books" applicatie en feedback van medestudenten tijdens code review.

**Leerdoel 2: Ontwerpen software - Methodisch front-end ontwerpen**  
Hoewel ik ervaring heb met front-end ontwikkeling, wil ik binnen 3 weken een meer gestructureerde ontwerpmethodiek ontwikkelen en toepassen, waarbij ik een volledig uitgewerkt ontwerp maak (wireframes, UI componenten, responsieve design principes) voordat ik begin met coderen. Dit zal gemeten worden door de kwaliteit van mijn ontwerpdocumentatie en de mate waarin het eindproduct overeenkomt met het ontwerp.

## Activiteiten front-end

**Stap 1: Analyse en planning (Week 1)**

**Beroepstaak: Kritisch, onderzoekend & methodisch werken**

Voor de start van de implementatie heb ik tijd besteed aan het analyseren van de opdrachteisen en het plannen van mijn aanpak. Ik heb de vereisten opgedeeld in:

* Minimale eisen (3 pagina's, 13 items, 4 attributen per item, etc.)
* Extra functionaliteiten (variabel aantal items per pagina, items toevoegen)

Hierbij heb ik een tijdsplanning gemaakt met haalbare mijlpalen per week. Deze gestructureerde aanpak heeft me geholpen om focus te houden en tussentijds de voortgang te meten.

**Gemaakte keuze:** Ik heb gekozen voor een boekencollectie als thema omdat dit zich leent voor rijke metadata (auteur, jaar, genre, etc.) en visueel aantrekkelijke presentatie (boekomslagen).

**Stap 2: Wireframing en UI design (Week 1)**

**Beroepstaak: Ontwerpen software**

Voordat ik begon met coderen, heb ik wire-frames gemaakt voor zowel desktop als mobiele weergave. Dit heeft me geholpen om een duidelijk beeld te krijgen van de gewenste gebruikerservaring en de benodigde componenten.

**Gemaakte keuze:** Ik heb gekozen voor een card-based design in plaats van een tabelstructuur omdat:

* Cards visueel aantrekkelijker zijn voor het tonen van boeken met afbeeldingen
* Ze beter schalen naar verschillende schermformaten
* Ze meer flexibiliteit bieden voor het tonen van verschillende hoeveelheden content

**Stap 3: HTML en CSS-basisstructuur (Week 1-2)**

**Beroepstaak: Realiseren software**

Ik heb eerst de basisstructuur van de HTML en CSS ontwikkeld, met focus op:

* Semantische HTML elementen
* Modulaire CSS componenten
* Responsive design met flexbox/grid
* Consistente styling en navigatie

**Gemaakte keuze:** Hoewel ik gewend ben aan component-gebaseerde ontwikkeling met React, heb ik ervoor gekozen om terug te gaan naar de basis en eerst een solide HTML/CSS fundament te leggen dit is tevens een harde eis voor dit opdracht. Dit heeft me geholpen om me te concentreren op core principes van toegankelijkheid en responsiveness zonder afhankelijkheid van frameworks.

**Stap 4: JavaScript functionaliteit implementeren (Week 2)**

**Beroepstaak: Realiseren software**

Voor de interactieve elementen heb ik vanilla JavaScript gebruikt om de volgende functionaliteiten te implementeren:

* Paginering (6 items per pagina)
* Sortering (op- en aflopend)
* Filtering op categorie
* Zoekfunctionaliteit
* Flip-card effect voor boekinformatie

**Gemaakte keuze:** Ik heb bewust een modulaire aanpak gehanteerd met duidelijk gescheiden verantwoordelijkheden, vergelijkbaar met hoe ik gewend ben te werken in React, maar nu met vanilla JavaScript. Dit resulteerde in onderhoudbare, leesbare code zonder externe afhankelijkheden.

**Stap 5: Testen en code review (Week 3)**

**Beroepstaak: Kritisch, onderzoekend & methodisch werken + Effectief communiceren**

In de laatste fase heb ik uitgebreid getest op verschillende apparaten en browsers. Daarnaast heb ik deelgenomen aan peer code review sessies met medestudenten.

**Gemaakte keuze:** Ik heb een systematische testaanpak gehanteerd met concrete testscenario's voor elke functionaliteit. Voor de code review heb ik ervoor gekozen om face-to-face sessies te houden, wat leidde tot directere feedback en betere uitwisseling van ideeën.

## Reflectie front-end

**Leerdoel 1: Realiseren van software - Versterken van vanilla JavaScript vaardigheden**

*Situatie:* Hoewel ik ervaring had met JavaScript in een React context, was ik minder vertrouwd met het implementeren van complexe functionaliteiten in vanilla JavaScript.

*Taak:* Ik moest deze vaardigheden versterken om alle vereiste interactieve elementen te kunnen implementeren zonder gebruik te maken van frameworks.

*Actie:* Ik heb me gefocust op het ontwerpen van een modulaire JavaScript codestructuur en het stap voor stap implementeren van de functionaliteiten.

*Resultaat:* Alle vereiste interactieve elementen zijn succesvol geïmplementeerd met alleen vanilla JavaScript. De code review bevestigde dat mijn aanpak effectief was.

*Reflectie:* Dit leerdoel heb ik behaald. De ervaring heeft mijn fundamentele JavaScript vaardigheden versterkt en me een dieper inzicht gegeven in DOM-manipulatie zonder frameworks, wat waardevol is ondanks mijn eerdere ervaring met React.

**Leerdoel 2: Ontwerpen software - Methodisch front-end ontwerpen**

*Situatie:* In mijn eerdere werkervaring werd het ontwerpproces vaak door UX-designers geleid, waarbij ik me meer richtte op de implementatie.

*Taak:* Ik moest een vollediger ontwerpproces doorlopen, inclusief wireframes en het maken van bewuste ontwerpkeuzes.

*Actie:* Ik heb uitgebreide wireframes gemaakt en ontwerpbeslissingen gedocumenteerd voordat ik begon met coderen.

*Resultaat:* Het ontwerp diende als solide basis voor de implementatie, wat resulteerde in een coherente en gebruiksvriendelijke interface.

*Reflectie:* Dit leerdoel heb ik grotendeels behaald. De methodische ontwerpaanpak heeft me geholpen om een beter eindproduct te leveren en heeft mijn begrip van het volledige ontwikkelproces verdiept.

**Toegepaste leerstrategieën**

1. **Terug naar de basis:** Door bewust alleen met vanilla JavaScript, HTML en CSS te werken, heb ik mijn begrip van web fundamentals versterkt.
2. **Incrementeel ontwikkelen:** Ik heb complexe functionaliteiten opgedeeld in kleinere, beheersbare delen die ik één voor één heb geïmplementeerd en getest.
3. **Kennisdeling:** Door code reviews en discussies met medestudenten heb ik nieuwe perspectieven gekregen en mijn eigen kennis kunnen delen.

**Ontvangen feedback en verwerking**

Tijdens de code review sessies heb ik waardevolle feedback ontvangen:

1. **Feedback over font-compatibiliteit (Eline):** Eline wees me erop dat mijn gekozen font niet goed werkte op haar Windows/Chrome combinatie. Ik heb dit opgelost door fallback fonts toe te voegen, wat de cross-browser compatibiliteit verbeterde.
2. **Positieve feedback over JavaScript structuur (Ensar):** Ensar gaf aan dat mijn JavaScript code goed gestructureerd en leesbaar was. Dit bevestigde dat mijn benadering van modulaire functies met duidelijke verantwoordelijkheden effectief was.

**Problemen en oplossingen**

1. **Uitdaging: Consistente card layout bij verschillende content lengtes** *Oplossing:* Ik heb CSS flexbox gebruikt met min-height en overflow eigenschappen om ervoor te zorgen dat kaarten consistent blijven, ongeacht de inhoud. *Leerpunt:* Deze uitdaging heeft me herinnerd aan het belang van robuuste CSS-oplossingen, zelfs met mijn eerdere ervaring in front-end ontwikkeling.
2. **Uitdaging: Complexiteit van paginering zonder framework** *Oplossing:* Ik heb het probleem opgedeeld in kleinere stappen en een modulaire aanpak gehanteerd. *Leerpunt:* Deze ervaring heeft mijn begrip verdiept van hoe frameworks zoals React deze functionaliteiten abstraheren, en hoe ze onderliggend werken.

**Kritisch denken**

Tijdens het project heb ik kritisch denken toegepast door:

1. **Evalueren van ontwerpbeslissingen:** Bij elke ontwerpkeuze heb ik kritisch gekeken naar de voor- en nadelen, en heb ik beslissingen gemaakt op basis van gebruikservaring, onderhoudsvriendelijkheid en schaalbaarheid.
2. **Analyseren van eigen code:** Bij het reviewen van mijn eigen code heb ik continu gezocht naar verbeteringen in structuur en efficiëntie.
3. **Balanceren van eerdere ervaring met opdrachteisen:** Ik heb bewust mijn neiging weerstaan om geavanceerdere technieken te gebruiken die buiten de opdrachteisen vielen, om me te concentreren op het versterken van fundamentele vaardigheden.

**Volgende stappen**

Voor de volgende fase van mijn ontwikkeling wil ik:

1. GitHub integreren in mijn werkstroom voor betere versiebeheer en gestructureerde code reviews
2. Mijn kennis van accessibility best practices verder verdiepen
3. Mijn begrip van JavaScript fundamentals verder uitbreiden
4. Geautomatiseerde tests implementeren voor front-end applicaties

Deze stappen zullen me helpen om mijn bestaande vaardigheden verder te ontwikkelen en mijn kennis te verbreden binnen het vakgebied van front-end ontwikkeling.

# LOGBOEK BACK-END

## Inleiding back-end

"My Favorite Books" webapplicatie voor de individuele opdracht back-end en verlenging op opdracht een Front-end. Het doel van dit logboek is om inzicht te geven in mijn aanpak, de gemaakte keuzes te onderbouwen, en mijn persoonlijke ontwikkeling tijdens dit opdracht te reflecteren.

Het logboek is opgebouwd uit drie hoofddelen:

1. Leerdoelen en uitgangspositie
2. Uitgevoerde activiteiten en gemaakte keuzes
3. Reflectie op het proces en de resultaten

Bij aanvang van dit semester had ik al enige ervaring met backend-end ontwikkeling door het volgen van een Full-stack cursus via (Codecademy, 2024) en het bouwen van kleinschalige persoonlijke projecten. Ik heb enige ervaring met het opzetten van een server, database, CRUD-acties en het bouwen API’s.

Voor deze opdracht is het doel om de basisprincipes van backend development te leren door te werken met JSON-server en Postman. Je leert queries maken, tests schrijven en collecties aanmaken om een correct werkende API-server te evalueren. Als extra uitbreiding kan de applicatie worden verbeterd door (Sequelize, n.d.) te implementeren, een object-relationele mapping tool die zorgt voor eenvoudige interactie met databases zoals PostgreSQL of SQLite. In dat geval vervangt Sequelize de functionaliteit van de JSON-server. De applicatie kan worden gedeployed in Azure. Tot slot is het de bedoeling dat de frontend wordt verbonden met de backend via de ontwikkelde API.

## Leerdoelen formuleren

Gebaseerd op de beroepstaken uit het curriculum heb ik de volgende leerdoelen geformuleerd:

**Leerdoel 1: Ontwerpen en realiseren van software (C-1 en D-1)**   
Binnen deze periode ga ik een werkende RESTful API bouwen met Express.js die mijn frontend applicatie kan bedienen. Ik zal deze testen met Postman en documenteren hoe elk endpoint functioneert. *Link met beroepstaken:* Dit leerdoel sluit aan bij het ontwerpen en realiseren van software (C-1 en D-1) en helpt me bij het methodisch ontwikkelen van backend oplossingen.

**Leerdoel 2: Realiseren van software (D-1) en het beheersbaar laten verlopen van ontwikkelprocessen (F-1)**  
Ik ga mijn frontend en backend lokaal met elkaar verbinden, waarbij ik leer hoe ik CORS-issues oplos en API-calls vanuit de frontend implementeer. Aan het eind van de periode zal ik de applicatie succesvol in Azure deployen. *Link met beroepstaken:* Dit sluit aan bij het realiseren van software (D-1) en het beheersbaar laten verlopen van ontwikkelprocessen (F-1). Door de volledige stack werkend te krijgen en te deployen, bereid ik me voor op de praktijk waar applicaties moeten worden uitgerold naar productieomgevingen.

## Activiteiten back-end

Hieronder beschrijf ik welke stappen ik genomen heb om mijn huidige doelen en de doelen die ik niet (volledig) behaald heb uit de vorige sprint aan te pakken in relatie tot succesvol uitvoeren van de opdracht.

**Reflectiepunt uit vorige sprint**GitHub integreren in mijn werkstroom voor betere versiebeheer en gestructureerde code reviews **Beroepstaak**: Beheersbaar laten verlopen van ontwikkelprocessen (F-1)  
In navolging van mijn leerdoel uit de front-end fase om GitHub beter te integreren in mijn werkstroom, heb ik voor deze back-end opdracht vanaf het begin GitHub gebruikt voor versiebeheer. Ik heb:

* Een repository opgezet specifiek voor de back-end code
* Gewerkt met feature branches voor verschillende API endpoints
* Pull requests gebruikt voor code review
* Duidelijke commit messages geschreven die de functionaliteit beschrijven

Tijdens deze sprint heb ik mijn collega’s gevraagd om mijn code via GitHub pull requests te reviewen. Dit heeft geleid tot waardevolle feedback over de structuur van mijn API-endpoints en verbeterde mijn code kwaliteit.

Ik heb bewust gekozen voor feature branches in plaats van direct op de main branch te werken, omdat dit me de mogelijkheid gaf om functionele delen van de API geïsoleerd te ontwikkelen en te testen voordat ze werden samengevoegd.

**Stap 1: Analyse en planning van de API-structuur (Week 1)**

Ik heb de opdrachtvereisten grondig geanalyseerd en mijn bestaande frontend data bestudeerd om te bepalen welke endpoints nodig zijn. Hiervoor heb ik een planning gemaakt met concrete mijlpalen.

Ik heb gekozen voor een resource-gebaseerde API-structuur die direct aansluit bij mijn frontend boekendata, omdat dit de meest logische en onderhoudbare manier is om mijn data te ontsluiten. Dit maakt ook het koppelen van de frontend later eenvoudiger.

**Stap 2: Opzetten ontwikkelomgeving en bestuderen tutorial’s (Week 1)**

Ik heb Node.js en Express.js geïnstalleerd en geconfigureerd. Daarnaast heb ik meerdere tutorials bekeken over RESTful API ontwikkeling en JSON-server.

Ik heb gekozen voor een modulaire projectstructuur met aparte bestanden voor routes, controllers en middleware, in plaats van alles in één bestand te plaatsen. Dit bevordert de onderhoudbaarheid en maakt het eenvoudiger om individuele componenten te testen.

**Stap 3: Implementatie van de API met Express.js (Week 2)**

Ik heb de API endpoints geïmplementeerd volgens RESTful principes, inclusief CRUD-operaties voor mijn boekendata.

Ik heb ervoor gekozen om HTTP-statuscodes consistent te gebruiken (200 voor succes, 404 voor niet gevonden resources, 400 voor ongeldige requests) om de client duidelijke feedback te geven. Daarnaast heb ik besloten om een gestandaardiseerd responseformaat te hanteren voor alle endpoints.

**Stap 4: Testen en documenteren met Postman (Week 2)**

Ik heb een Postman-collection gemaakt met tests voor elk endpoint en deze gebruikt om systematisch mijn API te testen.

Ik heb gekozen voor geautomatiseerde tests in Postman in plaats van handmatige tests, omdat dit me in staat stelt om snel regressietests uit te voeren na het maken van wijzigingen. Voor elk endpoint heb ik tests geschreven voor zowel het succespad als foutscenario's.

**Stap 5: Code review met studiegenoten (Week 3)**

Ik heb via GitHub een pull request gemaakt en een medestudent gevraagd om mijn code te reviewen. Na ontvangst van feedback heb ik verbeteringen doorgevoerd.

In plaats van alleen zelf mijn code te beoordelen, heb ik bewust andere ogen naar mijn werk laten kijken. Dit heeft geleid tot verbeteringen in mijn codestructuur en betere error handling.

**Stap 6: Verbinden van frontend en backend (Week 3)**

Ik heb CORS geconfigureerd in mijn Express.js applicatie en mijn frontend aangepast om gebruik te maken van fetch-calls naar de API in plaats van hardcoded data.

Ik heb ervoor gekozen om fetch API te gebruiken voor mijn API-calls in plaats van een library zoals Axios, om de afhankelijkheid van externe libraries te minimaliseren en mijn begrip van de fundamentele web APIs te vergroten.

**Stap 7: Deployment naar Azure (Week 3)**

Ik heb me verdiept in Azure deployment door tutorials te bekijken en documentatie te lezen. Vervolgens heb ik mijn applicatie succesvol gedeployed naar Azure.

Ik heb gekozen voor Azure App Service in plaats van andere deployment-opties, omdat dit het best aansluit bij mijn Node.js applicatie en relatief eenvoudig te configureren is, terwijl het toch professionele hosting biedt.

Dit aangepaste format voldoet aan de eisen van minimaal 5 stappen, bevat bewijsmateriaal en beargumenteert duidelijk de gemaakte keuzes voor elke stap.

## Reflectie back-end

Door gebruik te maken van STARR-methodiek evalueer ik hieronder tot hoeverre ik de opgestelde leerdoelen voor dit onderdeel behaald heb.

**Leerdoel 1: Ontwerpen en realiseren van software - Een werkende RESTful API bouwen**

**Situatie:** Hoewel ik alle eisen inclusief de optionele onderdelen wilde implementeren, kwam ik in de afgelopen periode in de knoop met timemanagement in de knoop. De reden was omdat ik ziek ben geworden en tegelijkertijd aan een nieuwe baan begon. Hierdoor had ik minder tijd dan oorspronkelijk.

**Taak:** Ik wilde de extra stappen 8,9 en 10 uit opdracht 2 implementeren.

**Actie:** Toen ik besefte dat ik onvoldoende tijd had om alle taken correct uit te voeren, heb ik een overweging gemaakt om alleen de verplichte onderdelen uit te voeren en mij te focussen op de documentatie (logboek en ontwerpdocument). Als ik tijd toch over hou, zal ik een poging doen om de overige onderdelen uit te voeren.

**Resultaat:** Alle verplichten onderdelen zijn naar mijn idee correct uitgevoerd en ik heb aandacht besteed aan een volledige documentatie.

**Reflectie:** Dit leerdoel heb ik (bijna) volledig behaald. Signaleren wanneer een andere aanpak nodig is dan oorspronkelijk gewild is essentieel voor het succesvol uitvoeren van projecten, zowel op school als op de werkvloer. Ik heb niet alle optionele onderdelen van dit opdracht uitgevoerd. Maar dat vind ik minder erg gezien ik hier eerder ervaring opgedaan heb. Daarentegen heb ik een volledige documentatie dat samenhang met wat ik gemaakt heb.

**Leerdoel 2: Frontend en backend verbinden en publiceren (deployment)**

**Situatie:** De problemen met mijn Microsoft Student licentie waren frustrerend, maar hebben me geleerd om alternatieven te overwegen en niet afhankelijk te zijn van één specifiek platform voor deployment.

**Taak:** Ik moest de volledige applicatie publiceren naar Azure.

**Actie:** Alhoewel mij niet gelukt is om de applicatie in Azure te publiceren, heb ik zowel de frontend als het backend lokaal met elkaar laten communiceren

**Resultaat:** De frontend communiceert nu naadloos met de backend op lokale omgeving.

**Reflectie:** Dit leerdoel heb ik deels behaald. De Azure deployment ging niet gelukt, maar ik hoop deze leerdoel mee te kunnen nemen naar het groepsproject waarin ik op de Azure omgeving van een klasgenoot toch nog ervaring hierop kan doen. Daarnaast, heb ik dagelijks met de Azure omgeving op werk te maken. Ik verwacht dat mijn werkplek mijn in de gelegenheid stelt om mijn kennis in Azure te versterken en mij voor zal bereiden op de 5de semester.

**Toegepaste leerstrategieën**

**Hands-on leren:** Door direct te werken aan de implementatie van de API en praktische problemen op te lossen, heb ik de concepten veel beter begrepen dan alleen uit theorie mogelijk zou zijn.

**Iteratieve aanpak:** Ik heb eerst een werkende basisversie gemaakt en deze stapsgewijs verbeterd, wat me in staat stelde om snel resultaten te zien en bij te sturen waar nodig.

**Ontvangen feedback en verwerking**

Tijdens het ontwikkelproces heb ik feedback verzameld via:

**Code review via GitHub:** Mijn medestudent Joris heeft een code review uitgevoerd en suggesties gedaan voor het verbeteren van mijn error handling. Hij merkte op dat ik in sommige gevallen geen specifieke HTTP-statuscodes terugstuurde. Ik heb dit verbeterd door consistente error responses te implementeren met de juiste statuscodes.

**Feedback tijdens API-testen:** Tijdens een online Teams sessie met medestudent Ensar hebben we de API samen getest het bleek dat mijn endpoints niet consistent waren in het formaat van de responses. Ik heb vervolgens een standaard responseformaat geïmplementeerd voor alle endpoints.

**Problemen en oplossingen**

**Uitdaging:** Omgevingsconfiguratie voor Azure deployment werkte niet vanwege licentie problemen   
**Oplossing:** Ik heb de frontend en het backend lokaal met elkaar laten communiceren.

**Kritisch denken**

Tijdens de ontwikkeling van de back-end heb ik kritisch denken toegepast door:

**Evalueren van verschillende API-ontwerpen:** Ik heb verschillende API-structuren overwogen en de voor- en nadelen geanalyseerd voordat ik een keuze maakte. Hierbij heb ik gekeken naar schaalbaarheid, onderhoudsvriendelijkheid en gebruiksgemak voor de frontend.

**Herkennen van eigen kennishiaten:** Ik heb actief mijn eigen kennis geëvalueerd en specifieke gebieden geïdentificeerd waar ik meer moest leren (bijvoorbeeld over Sequelize configuratie), waarna ik gericht aanvullende bronnen heb gezocht.

Door deze aanpak van kritisch denken ben ik tot weloverwogen beslissingen gekomen die hebben bijgedragen aan de kwaliteit van de uiteindelijke applicatie. Het heeft me ook geholpen om efficiënter te werken door problemen systematisch aan te pakken in plaats van via trial-and-error.

# LOGBOEK EINDPRODUCT

## Inleiding

* Geef duidelijk aan wat het doel is van dit logboek; de opbouw van het rapport en de aanpak.
* Geeft ook aan in welke mate je bekend bent met de technieken die gevraagd worden en wat je nog moet leren.

## Leerdoelen formuleren:

* Formuleer twee leerdoelen (SMART) waar je deze periode, tijdens het ontwerpen van het eindproduct, aan gaat werken, bijv. hoe ontwikkelprocessen kunnen worden beheerd of dat je wil leren hoe je jullie oplossingen het beste testen en evalueren.
* Zorg ervoor dat er een duidelijke link is met de beroepstaken en de producten die deze periode aan bod komen.

## Activiteiten

* Beschrijf de stappen (minimaal 5 stappen) die je hebt genomen om je doelen te bereiken, bijv. het plannen van de opdracht, het verdelen van taken, het opstellen van een tijdschema, coderen, het lezen van boeken, het bekijken van tutorials, het voeren van gesprekken met studiegenoten of collega’s, code review (gegeven en ontvangen) enz. Het bewijsmateriaal kan bestaan uit een afbeelding, bestand(en), link, etc.
* Beargumenteer ook je gemaakte keuzes.

## Reflectie

Hier beschrijf je:

* In hoeverre je de gestelde leerdoelen hebt bereikt en hoe je dat weet. Bij voorkeur gebruik je de STARR(T)-methodiek.
* Welke leerstrategieën je hebt toegepast /experimenteert (stellen van vragen, maken van aantekeningen, zoeken naar aanvullende bronnen, samenwerken, etc.) en wat het nut daarvan was voor je werk.
* Welke feedback je hebt ontvangen en wat je daarmee hebt gedaan. Op welke manier heb je feedback verzameld?
* Beschrijf de problemen waar je tegenaan bent gelopen, hoe je ze opgelost hebt en wat je hiervan geleerd hebt.
* Leg uit hoe je kritisch denken hebt toegepast in je werk, bijv. het evalueren van bronnen, het identificeren van zwakke punten in je benadering, enz.
* Eventuele extra aantekeningen of observaties die relevant zijn voor je opdracht.

Transfer naar het volgende semester:

* Beschrijf wat van wat je hebt geleerd, je mee gaat nemen naar het volgende semester en waar je aandacht aan wil besteden om het verder te verbeteren.