

## Opdrachtomschrijving

In deze module ga je een *full-stack applicatie* maken. Dit betekent dat je een **frontend** maakt in HTML, CSS en JavaScript. De frontend krijgt zijn informatie van jouw zelfgebouwde **backend**, geschreven in JavaScript met NodeJS en Express. Het onderwerp van de opdracht bepaal je zelf, maar je moet wel aan de onderstaande eisen voldoen.

Let op: Dit is een **individuele** opdracht.



### Eisen

- De applicatie moet bestaan uit een aparte frontend en backend
- De frontend moet geschreven zijn in HTML, CSS en JavaScript
- De backend moet geschreven zijn in JavaScript, met NodeJS en Express
  
- De backend moet de REST-specificatie volgen, inclusief de correcte statuscodes
- De backend moet uit minstens drie resources bestaan die door relaties met elkaar verbonden zijn
- Minstens één resource moet een one-to-many relatie hebben met een andere resource
- De backend moet volledige CRUD (Create, Read, Update, Delete) functionaliteit hebben voor minstens twee resources.
- De CRUD-functionaliteit moet de juiste HTTP Verbs gebruiken (GET, POST, PUT, DELETE).
- De backend moet zijn data opslaan in een SQLite database
- Minstens 3 backend endpoints moeten query parameters gebruiken
- Minstens 3 backend endpoints moeten path parameters gebruiken
  
- De frontend moet bestaan uit meerdere pagina's (HTML), die volledig opgemaakt zijn (CSS) en functioneel zijn (JS)
- De pagina's moeten hun data halen uit je eigen backend, opgehaald met de Fetch API
- Je pagina's moeten nieuwe data naar je eigen backend sturen, via de Fetch API
- Je aanroepen met de Fetch API moeten foutafhandeling hebben. Je moet checken of er iets fout gaat, inclusief foutmeldingen via statuscodes en andere (onverwachte) fouten
- Je pagina's moeten valideren wat ingevuld wordt in formulieren met JavaScript en de ingebouwde HTML-formuliervalidatie
- De frontend moet gestart kunnen worden door index.html te openen en maakt geen gebruik van package managers
- De frontend moet een zelfgeschreven CSS-bestand bevatten
- De opmaak mag met behulp van een CSS-only framework gemaakt worden dat geen JavaScript gebruikt, als je hier vooraf toestemming voor hebt gekregen van je docent
- De JavaScript in de frontend moet geschreven worden zonder externe JavaScript frameworks
  
- De REST-specificatie van de backend moet volledig gedocumenteerd zijn, inclusief alle mogelijke statuscodes, en voorbeeld requests en responses.
- Minimaal twee (wat complexere) requests moeten gedocumenteerd zijn door middel van sequence diagrammen.
- Ontwerpkeuzes, inclusief gebruikte frameworks of architectuurkeuzes, moeten gedocumenteerd worden.

### Extra informatie

De applicatie moet bestaan uit twee aparte projecten: een folder met de NodeJS/Express backend applicatie en een andere folder met de HTML, CSS en JavaScript frontend. Voor het opstarten van de applicatie zou zo simpel moeten zijn als alleen maar De NodeJS backend op te starten (zonder configuratie) en de index.html te openen.

Voorzie je problemen met het voldoen aan deze eisen? Neem dan zo vroeg mogelijk in het proces contact op met je docent. Door langer te wachten neem je een groot risico! Benader je je docent hier pas over na de deadline, dan ben je te laat en zal je vraag niet meer beantwoord worden.

## Deadline en beoordeling

Je moet de volgende eindproducten inleveren:

- Backend: je NodeJS/Express project (code) als een ZIP-bestand
- Frontend: je HTML-, CSS- en JS-bestanden inclusief een index.html als beginpunt. Dit ook als een ZIP-bestand.
- Je documentatie waarin op zijn minst de punten worden behandeld die hierboven onder Eisen genoemd worden. Zorg dat je document een fatsoenlijke voorpagina heeft, waarop je studentnummer en naam vermeld staan.
- **ALLEEN HERKANSING: Een changelog die een lijst bevat met wijzigingen die je hebt aangebracht ten opzichte van je eerste poging.**

De inleverdeadline voor de opdracht is de **maandagochtend van week 8 vóór 09:00 uur**.

Als je voor deze deadline niets inlevert, je een onvoldoende haalt, of niet voldoet aan de inlevervoorwaarden (zie hieronder) dan kun je de opdracht herkansen. De opdracht voor de herkansing is dezelfde als de oorspronkelijke opdracht, dus je kunt dezelfde applicaties inleveren (uiteraard dan wel verbeterd en compleet gemaakt). Zoals hierboven vermeld, voeg een changelog toe aan je inlevering, zodat we alleen de punten die je veranderd hebt opnieuw hoeven nakijken.

De deadline voor de herkansing is de **maandagochtend van week 11 vóór 09:00 uur**.

*Let er alsjeblieft op dat de deadlines dus afwijken van wat je tot nu toe gewend bent door de BSA-regeling. Houd hier rekening mee wanneer je je werk voor deze module aan het plannen bent.*

### Let op: inlevervoorwaarden!

Voordat we je ingeleverde opdracht beoordelen, controleren we of wat je inlevert voldoet aan een aantal voorwaarden. Als je niet aan deze voorwaarden voldoet, wordt je opdracht beoordeeld als “niet beoordeelbaar” en krijg je geen cijfer.

De inlevervoorwaarden zijn als volgt:

#### Backend:

- Is de backend geschreven in JavaScript, met NodeJS en Express?
- Werkt de backend als de applicatie wordt gestart “npm start” zonder extra configuratie?
- Heeft de backend een werkende SQLite databron?

#### Frontend:

- Is de frontend geschreven in HTML, CSS en standaard JavaScript?
- Werkt de frontend door de index.html te openen, zonder een package manager te hoeven starten of enig ander commando uit te voeren?
- Is de frontend verbonden met je backend (via Fetch)?

#### Documentatie:

- Is de vereiste documentatie ingeleverd?

### Beoordeling

Als je aan de inlevervoorwaarden voldoet dan beoordelen we je opdracht. Je krijgt een score tussen de 0 en 100 punten, waarbij 55 punten of hoger een voldoende is. Deze punten worden omgerekend tot een cijfer door je puntentotaal te delen door 10 en dan af te ronden naar het dichtstbijzijnde gehele getal.

De onderwerpen waarop we beoordelen en hun gewicht vind je op Blackboard. Ga hiervoor naar het inleverpunt en klik op de “Toon Rubric” knop.

Points Possible

**100**

[View Rubric](#)