

aantal pagina's.:	lector: Jeroen Rombouts opleidingsonderdeel (OPO/OLA): Full-Stack Software Development Afstandsonderwijs code OPO/OLA: MBI05h
datum: 13/11/2025 groep: n/a	naam student: ....Weustenraad..... voornaam: ..Riccardo..... studentennummer: ..r0751962.....
	punten: /14

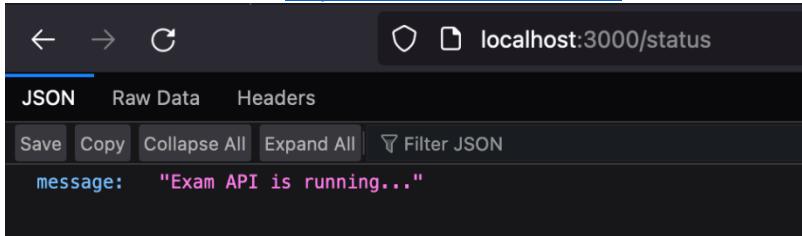
## Deel 1: opstarten en testen startcode

### Voorbereiding:

- Voor zowel back-end als front-end voer je `npm install` uit.
- Zorg dat in de back-end folder een `.env file` staat met connectie details naar je **lokale** postgres database.
  - `DATABASE_URL="postgresql://postgres:postgres@localhost:5432/fullstack?schema=..."`
  - Uiteraard zorg je dat die database en schema (naam naar keuze) reeds aangemaakt zijn.
- Voor front-end is er reeds een `.env` file voorzien

### Test back-end:

- Start de back-end: `npm start`
- Valideer de status URL: <http://localhost:3000/status>



- Valideer dat de Swagger URL <http://localhost:3000/api-docs> werkt. De endpoints doen nog niets op dit moment.
- Loop de database migratie: `npx prisma migrate dev` en kijk na of die tabellen aangemaakt zijn.
- Initiële testdata voorzien: `npx ts-node util/seed.ts` en kijk na of de user tabel een record bevat voor een admin user

### Test front-end:

- Opstarten: `npm run dev`
- Valideer URL <http://localhost:8080>

Als je hier niet geraakt: contacteer me dan asap via email.

groep: .ME-2TIAF.....	datum: ...13/11/25.....pg. 2
naam student: ....Weustenraad.....	lector: ..Jeroen.Rombout.....
voornaam: Riccardo.....	
studentennummer: ..r0751962.....	

### **De opdracht:**

Bij elke vraag hoort een begeleidend filmpje (zonder geluid) dat een beeld geeft van het verwachte eindresultaat. Bekijk dit eerst, lees dan de volledige vraag. Bekijk indien nodig per deelvraag het filmpje opnieuw.

### **Voor het mondeling examen:**

Push je code naar een publieke github. Het eerste deel van het mondelinge examen is het tonen en doorlopen van de code. Hierbij worden deelvragen gesteld om te testen of de concepten gekend zijn.

Score: 14 punten zoals bij de vragen staan, 7 punten op concepten

Het tweede deel van het examen gaan we je project samen doorlopen en daar gaan een paar theorievragen over gesteld worden:

score: 7 punten op kennis van concepten

Totale score wordt herleid tot 14. (de andere 6 staan op het permanente evaluatie project)

Tijdens het examen is er niets van hulpmiddelen toegestaan: enkel VScode, een postgresql client en een browser.

### **Evaluatiecriteria**

- De functionaliteit van elke deelvraag moet volledig werken, anders krijg je een 0 voor die deelvraag.  
Je kan punten verdienen met elke deelvraag.
- Als de functionaliteit werkt, moet de code ook correct geïmplementeerd zijn volgens de aangeleerde principes om de volledige punten van een deelvraag te verdienen.

### **Algemene hints**

- Het kan handig zijn om tijdens het examen het **seed** script meerdere keren uit te voeren als je iets opnieuw wil testen. Je mag hier ook extra data aan toevoegen als je dat nodig hebt tijdens het coderen.
- Bekijk de bestaande code. Er staan alvast hints in.
- Bij wijzigingen aan het prisma schema en het genereren van een nieuwe prisma client moet je mogelijks VSCode herstarten. VSCode werkt soms nog met een oude versie in cache en geeft dan bijgevolg compile fouten in database klassen en mappers. Herstarten maakt die cache leeg.

groep: ....ME-2TIAF.....	datum: ....13/11/25.....pg. 3
naam student: ..Weustenraad.....	lector: ..Jeroen Rombouts.....
voornaam: ...Riccardo.....	
studentennummer: ....r0751962.....	

---

## OPGAVE

---

### VRAAG 1 - 5 punten

Demo: vraag1.mp4

Een bezoeker moet reizen kunnen bekijken in een overzicht en in detail.

- a) Pas de layered architecture toe om de gebruiker een lijst te laten zien van alle holiday trips. Dit wil ook zeggen: check het prisma schema en maak de database aan. Laat ook de seed lopen.
- b) Elke holiday trip heeft een button “meer info” Dit navigeert naar een pagina met de info in meer detail over 1 trip en een lijst van alle attendees. Zie filmpje voor details.
- c) Schrijf minstens 4 tests.
- d) Doe de login werken zoals in de clip. Let op: Engelse spelling is overal gebruikt voor organiser (met een s, en niet met een z)

groep: .....ME-2TIAF.....	datum: 13/11/25.....pg. 4
naam student: Weustenraad.....	lector: Jeroen Rombouts.....
voornaam: Riccardo.....	
studentennummer: r0751962.....	

## VRAAG 2 - 9 punten

Demo: vraag2.mp4

Iemand met de rol organiser moet een experience kunnen toevoegen en een lijst kunnen bekijken van zijn of haar eigen experiences. Een klant moet enkel de lijst kunnen bekijken. Let er op dat de Experiences in de back-end een andere naam hebben.

- a) Maak in de back-end, volgens de layered architecture, alle nodige types , klassen en routes aan om dit mogelijk te maken. (1pt)
- b) Als een user inlogt met de rol "CLIENT", moet hij of zij alle experiences kunnen zien. Het aantal getoonde Experiences staat tussen haakjes in de ondertitel. Een niet ingelogde user krijgt een boodschap te zien en een link naar de login-pagina (1 punt)
- c) Als de ingelogde user de rol "Organiser" heeft, moet er een knop bovenaan de pagina komen te staan met de tekst "Show Only Mine" als daar op geklikt wordt, komen van de back-end (niet filteren in de front-end) de lijst terug van de experiences waar de ingelogde gebruiker de organisator van is. De tekst van de knop verandert in "show all experiences" in deze view en de ondertitel ook. Zie clip voor de duidelijkheid. (2ptn)
- d) De ingelogde organisator moet ook nieuwe experiences kunnen aanmaken: pas hiervoor een aparte component CreateExperienceForm aan. En voorzie de functionaliteit. (2ptn)
- e) Schrijf minstens 4 relevante testen voor de functionaliteit rond Travel Experiences. (2ptn)
- f) Indien een organisator een nieuwe experience wil aanmaken op een dag waarop hij er al een heeft, krijgt hij een error terug. (niet getoond in mp4) (1pnt)