

datum: 14/11/2024

groep: n/a

naam student:

voornaam:

studentennummer:

punten: /14

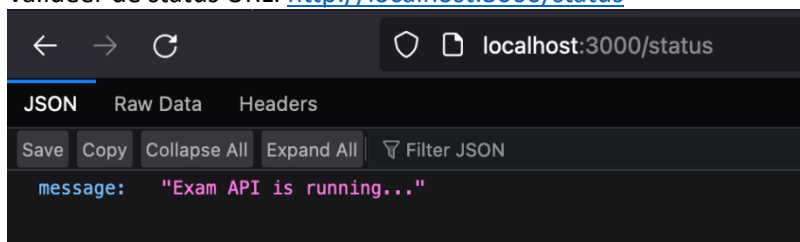
Deel 1: opstarten en testen startcode

Vorbereiding:

- 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
- Via Swagger kan je nu het /users endpoint uittesten. Dat zou dezelfde admin user moeten teruggeven.

Test front-end:

- Opstarten: **npm run dev**
- Valideer URL <http://localhost:8080> (het is normaal dat enkel de link naar de login pagina werkt)

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

groep:	datum:pg. 2
naam student:	lector:
voornaam:	
studentennummer:	

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.
- 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:
naam student:
voornaam:
studentennummer:

datum:pg. 3

lector:

OPGAVE

VRAAG 1 - 5 punten

Demo: vraag1.mp4

Zorg voor een functioneel werkende teachers pagina, zoals getoond in het demo filmpje. Je zal zowel in de back-end en front-end moeten werken. Er is reeds bestaande code waarop je moet verder werken. In de code staan belangrijke aanwijzingen in commentaar!

- Voeg een "Teachers" item toe aan het menu die leidt naar een pagina op het pad `/teachers`. Deze pagina is voor iedereen, inclusief anonieme gebruikers, toegankelijk.
- Deze pagina moet een tabel tonen met alle gebruikers die de rol "teacher" hebben. In elke rij staat de naam van de teacher en het "learning path" waarin deze actief is als platte tekst.
- Enkel wanneer je als gebruiker ingelogd bent met een "admin" rol, dan wordt het learning path op de teachers pagina getoond als een dropdown. In alle andere gevallen blijft dit platte tekst.
- Het wijzigen van het learning path in de dropdown wordt onmiddellijk doorgevoerd in de database.

groep:	datum:pg. 4
naam student:	lector:
voornaam:	
studentennummer:	

VRAAG 2 - 9 punten

Demo: vraag2.mp4

Maak een formulier zodat admins nieuwe classrooms kunnen toevoegen, zoals getoond in het demo filmpje. Hiervoor bestaat nog geen enkele code in het project! Je zal dus alles (back-end en front-end) zelf moeten programmeren. Zorg ervoor dat alle lagen van de architectuur correct geïmplementeerd worden.

- Voeg een "Add classroom" item toe aan het menu die leidt naar een pagina op het pad `"/classroom"`. Dit menu item mag enkel voor ingelogde gebruikers met de rol "admin" zichtbaar zijn.
- Wanneer je rechtstreeks naar de pagina `"/classroom"` navigeert, zonder dat je ben ingelogd als gebruiker met "admin" rechten, dan krijg je een foutboodschap.
- Alle titels, labels en tekst die geschreven worden voor deze pagina en bijhorende componenten mogen daarin niet als hard-coded tekst geplaatst worden. Je moet de mogelijkheid voorzien dat deze vertaald kunnen worden volgens de principes van i18n. Je hoeft enkel de Engelstalige tekst te voorzien, er hoeft ook geen knop te zijn om te veranderen van taal. Tip: na het toevoegen/wijzigen van een vertaling herstart je best `next.js`.
- Op de "classroom" pagina staat een formulier waar je de naam van de classroom kan ingeven. Dat formulier moet een herbruikbare component zijn. Bouw validatie in zodat je een (vertaalbare) foutboodschap krijgt wanneer je het formulier leeg submit.
- Wanneer je het formulier geldig submit stuur je een request naar een end-point "classrooms" in de back-end API en wordt deze classroom toegevoegd in de database. Implementeer dit end-point in de back-end zodat deze als antwoord de toegevoegde classroom (inclusief database record ID) terugstuurt.
- Wanneer het back-end request om toe te voegen succesvol was toon je een succes boodschap boven het formulier, waarin duidelijk het ID van het database record staat. Het inputveld in het formulier moet dan terug leeggemaakt worden.
- Indien je een classroom probeert toe te voegen dat reeds bestaat in de database, toon dan een foutboodschap boven het formulier en laat de ingevulde waarde in het inputveld staan.
(Opmerking: wanneer je niet klaar geraakt met deze vraag, zorg ervoor dat de functionaliteit t.e.m. vraag F zeker nog werkt).