

Dynamic Webapps

Eindopdracht

1 Evaluatie

1.1 Dynamic_oefeningen Project

We werken gedurende de lessen een **project** uit waaraan de **leerstof** wordt opgehangen. Omdat het essentieel is dat elke stap gekend is, wordt wekelijks het gerealiseerde project ingeleverd.

1.2 Toetsmomenten

Er zijn twee toetsmomenten voorzien.

De eerste toets telt mee voor **20%** van het eindcijfer en de tweede toets telt mee voor **30%** van het eindcijfer.

Hierin vragen we om een aanpassing te doen in code die de student op voorhand krijgt.

1.3 Eindopdracht

De student krijgt ook een **eindopdracht** voor **50%** van het eindcijfer waarin alle aspecten van de leerstof aan bod komen. De details van die opdracht worden in het begin van het kwartaal bekendgemaakt.

Studenten kunnen steeds feedback vragen en krijgen over de ingeleverde tussentijdse opdrachten waardoor ze de eindtaak tot een goed einde kunnen volbrengen.

Op het einde van het kwartaal is er een **mondeling examen** over de wekelijkse opdrachten én de eindopdracht om na te gaan of dit eigen werk is en of de student voldoende inzicht heeft verworven in de materie. Het niet kunnen antwoorden op relevante technische vragen leidt tot een onvoldoende op de opdracht. De student kan tijdens deze verdediging gevraagd worden om evidente technische zaken aan te passen, teneinde na te gaan of hij voldoende inzicht heeft verworven.

2 EindOpdracht specificaties

Je mag zelf een thema bedenken voor je applicatie.

Elke student maakt **individueel** een project.

Tijdens de mondelinge evaluatie moet je dit kunnen tonen in je code en ook in de code die we gebruikt hebben tijdens de les.

Minimum technische specificaties:

dit betekent: als dit allemaal in je project zit, het is correct uitgevoerd en je kan het uitleggen – dan heb je de helft van de punten voor deze opdracht (dus 5/10)

- minstens **15** verschillende **eigen** Components (al dan niet in aparte files)
- minstens 1 keer een list/array/collection van Components correct renderen
- minstens 1 keer props.children gebruiken
- minstens 1 keer een Component **anders** (of niet) laten renderen afhankelijk van een conditie
- minstens 1 keer een Fragment gebruiken
- minstens 5 react-bootstrap Components gebruiken
- minstens 5 eventHandlers
- minstens 5 keer “use_state”
- minstens 1 state die door meerdere components gebruikt wordt
- minstens 2 **eigen** contexts
- localStorage
- minstens 5 input elementen waar de gebruiker iets moet ingeven **en** doe iets nuttigs hiermee
- maak een database in FireBase
- lees data uit de database
- schrijf data in de database
- gebruik minstens 1 async functie
- minstens 1 useEffect

Let ook op het volgende:

- probeer voor edit/new-pagina's de input velden zo handig mogelijk te maken voor de gebruiker (geen text veld als een select mogelijk is)
- applicatie moet nog werken als connectie met de database faalt of als er geen data aanwezig is voor jouw project

Gebruik git en Github:

- je project is gepushed in github
- docent is toegevoegd als collaborator
- je hebt op branches gewerkt (branches niet deleten na mergen naar master)
- je hebt op regelmatige basis gecommited

Extra punten bij gebruik van technieken verder dan de leerstof:

- meer toegevoegd dan nodig, met behulp van documentatie in Canvas of iets dat je zelf opgezocht hebt
- opdracht is verrassend en creatief uitgewerkt

Noot:

- Je mag mijn componenten hergebruiken maar die tellen niet mee als **eigen** Components
- Je mag mijn Contexts hergebruiken maar die tellen niet mee als **eigen** Contexts
- Je mag code-snipjets gebruiken die je op het internet gevonden hebt maar je moet deze kunnen verklaren en aanpassen. Als het grotere stukken zijn vermeld dan zeker ook waar je ze gehaald hebt.

Je kan **niet** slagen als:

- code in github werkt niet
- je hebt niet op regelmatige basis gecommitt
- je code is niet gedeployed
- thema en opbouw is niet voldoende verschillend van het voorbeeld-project of onderdelen daarvan
- je hebt je project gestart als copy van voorbeeld-project. Bouw je eigen applicatie op!

Ik trek punten af als ik dit zie:

- ik moet eerst aanpassingen doen aan de code van je project voor het werkt
- ik kan je project niet runnen omdat niet alle nodige gegevens aanwezig zijn in canvas (login etc)
- je hebt de docent niet uitgenodigd op je GitHub repository
- je GitHub repository is public
- Look&Feel: broken links, niet goed uitgewerkt
- Look&Feel: het geheel ziet er niet professioneel uit
- je project is niet correct gestructureerd in files en directories
- filenames en directory names volgen niet de coding conventies van de taal
- code volgt niet de coding conventies van de taal
- de code is niet proper geformatteerd
- er is overbodige code of overbodige variabelen, functies, ...
- edit-pagina voor bestaande data: input velden in een form moeten altijd pre-filled zijn, zodat het bruikbaar is voor de gebruiker
- gebruik van classes (als je code van het internet haalt en die werkt met classes zet dat dan om naar functionele components)
- warnings in console
- gewone for-loop om een array te verwerken – behalve als je echt een duidelijke reden hebt waarom het niet anders kan (en dat ook kan uitleggen)
- je past een array of object aan (in state)
- je gebruikt let of var – behalve als je echt een duidelijke reden hebt waarom het niet anders kan (en dat ook kan uitleggen)

3 Project dynamic_oefeningen indienen

Op het einde van elke bundel dien je de aanpassingen in die je gedaan hebt aan het project dat we samen in de klas opbouwen (dynamic_oefeningen).

Er zal steeds een opdracht zijn in Canvas.

De juiste datums worden tijdig gecommuniceerd op Canvas.

4 EindOpdracht resultaten indienen

Tijdens de cursus dient de student 2 keer een tussentijds resultaat in.

In het begin van de laatste week dien je de definitieve versie in.

De juiste datums worden gecommuniceerd op Canvas.

5 Een opdracht indienen in Canvas

- Dien iets in dat werkt! Zelfs al is niet alles af, hetgeen je gemaakt hebt moet correct werken.
- Alles moet gecommit en gepushed zijn en gemerged naar master. Alleen code op master telt mee. Alleen commits die op tijd gebeurd zijn tellen mee.
- Zip je project, zonder de folders **target**, **.git**, **build**, **node_modules**, **.idea**, **.vscode**
- Upload in **Canvas** in de juiste opdracht:
 - je **gezipte** project. Upload nooit een zip waar **target**, **.git (etc)** folders inzitten!
 - url van je github repository
 - url van de gedeployde applicatie