Tussentijdse toets 2024

Deze toets bestaat uit 4 opdrachten. Elke opdracht staat hieronder beschreven.

Voor deze toets mag je **WEL** gebruik maken van de volgende hulpmiddelen:

- VS Code (+ intellisense en extensions). Het gebruik van devcontainers is niet verplicht.
- Eigen oplossingen van labo's (lokaal opgeslagen)
- Eigen samenvattingen van theorielessen (lokaal opgeslagen)
- Cursusmateriaal (https://similonap.github.io/webframeworks-cursus/)
- Officiele react documentatie (https://react.dev/)
- React Router documentatie (https://reactrouter.com/en/6.4.3)
- Mozilla Developer Network (MDN) (https://developer.mozilla.org/en-US/docs/Web)
- Quicktype (https://app.quicktype.io/)

Voor deze toets mag je **NIET** gebruik maken van de volgende hulpmiddelen:

- AI (chat-gpt, copilot, ...)
- Online bronnen (tenzij specifiek anders aangegeven in de opdracht)
- Communicatiemiddelen (chat, discord, fora, telefoon, ...)

Tips

- Er is geen verplichte volgorde om de opdrachten te maken. Je kan zelf kiezen in welke volgorde je de opdrachten maakt.
- Verlies niet te veel tijd aan de opmaak van de pagina's of andere visuele aspecten. Er staan geen punten op het uiterlijk van de pagina's.
- Er staan vaak tips in de opdrachten die je kunnen helpen bij het maken van de opdracht. Lees de opdrachten dus goed door.
- Je kan een live demo van de opdrachten bekijken op https://2024-toets-demo-1-wf-3o3qu0psm-similonaps-projects.vercel.app/

Boetesysteem

Als er een of meerdere van de volgende zaken van toepassing zijn, wordt er 25% van de punten afgetrokken:

- Als je de `node_modules` map meestuurt met je project. Zorg ervoor dat je de `node_modules` map verwijdert voor je je project inlevert.
- Als je ongevraagde features/code overneemt van andere oefeningen of labo's. Bv: toevoegen van een light/dark mode indien dit niet gevraagd

is.
- Als er naamgeving van variabelen, properties, states, routes, componenten overgenomen wordt van andere oefeningen of labo's. Bv: Je gebruikt de naam `fetchPokemon` voor het ophalen van de games.
- Als het project niet compileert en/of niet opstart. Dus zorg ervoor dat je niet werkende code in commentaar plaatst en duidelijk aangeeft dat dit niet werkt.

Dus haal je een 15/20 op de toets, dan wordt dit omgezet naar een 11.25/2 als je een van de bovenstaande zaken hebt gedaan. Dus let goed op als je code overneemt van andere oefeningen of labo's.

Opdracht

Routing en Pagina's (2pt)

Voeg de volgende pagina's toe aan de applicatie:

- [ ] Maak een `Home` component op de route `/`. Deze component bevat een welkomstboodschap die je zelf mag kiezen en een lijst met links naar de 3 opdrachten.
- [ ] Maak een `Opdracht1` component op de route `/opdracht-1`. Voorlopig hoeft deze component nog geen functionaliteit te bevatten.
- [ ] Maak een `Opdracht2` component op de route `/opdracht-2`. Voorlopig hoeft deze component nog geen functionaliteit te bevatten.
- [ ] Maak een `Opdracht3` component op de route `/opdracht-3`. Voorlopig hoeft deze component nog geen functionaliteit te bevatten.
- [ ] Voeg een navigatiebalk toe aan elke pagina met links naar de homepagina en de 3 opdrachten. Zorg voor een `Root` component die de gemeenschappelijke navigatiebalk bevat en een `Outlet` bevat.
- [ ] Zorg ervoor dat de navigatiebalk de huidige pagina aangeeft door een visuele indicatie te geven (bijvoorbeeld een andere kleur).
- [ ] Zorg voor een 404 pagina die getoond wordt wanneer een onbekende route wordt bezocht.

TIP: Lukt het je niet om de routing te implementeren? Dan kan je de opdrachten ook maken zonder routing. Je kan de `Opdracht1`, `Opdracht2` en `Opdracht3` componenten dan gewoon in de `App` component onder elkaar plaatsen. Je verliest dan enkel de punten op dit onderdeel.

Opdracht 1: Temperatuur conversie (5pt)

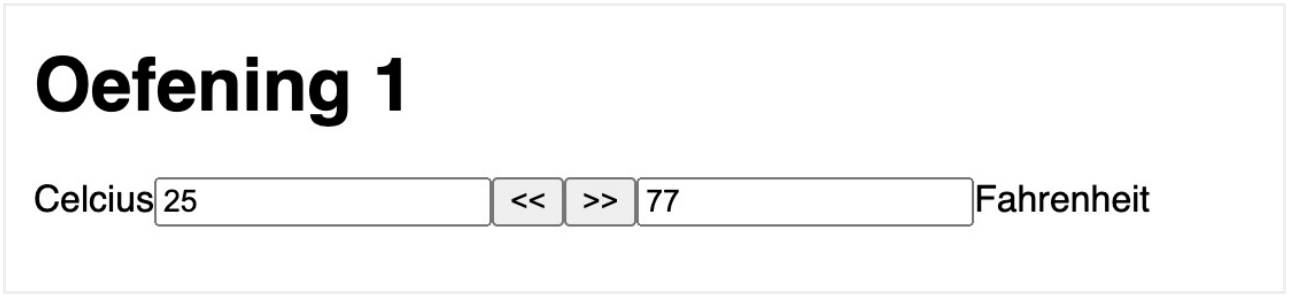
Voeg functionaliteit toe aan de `Opdracht1` component die het mogelijk maakt om een temperatuur in te geven in graden Celcius en deze om te zetten naar graden Fahrenheit en omgekeerd. Je kan de volgende formules gebruiken:

- [ ] Maak een input veld waarin de temperatuur in graden Celcius kan worden ingegeven.
- [ ] Maak een input veld waarin de temperatuur in graden Fahrenheit kan worden ingegeven.
- [ ] Maak twee knoppen om de conversie uit te voeren: één om van Celcius naar Fahrenheit te converteren en één om van Fahrenheit naar Celcius te converteren.
- [ ] Als je op de knop klikt om van Celcius naar Fahrenheit te converteren, moet de waarde in het input veld voor Fahrenheit automatisch worden aangepast en omgekeerd. De conversie mag enkel gebeuren wanneer de gebruiker op de knop klikt, dus niet automatisch bij het wijzigen van de input velden.

Je kan hiervoor gebruik maken van de volgende formules:

- Celcius naar Fahrenheit: `Fahrenheit = (Celcius * 9 / 5) + 32`
- Fahrenheit naar Celcius: `Celcius = (Fahrenheit - 32) * 5 / 9`

TIP: Je kan best twee functies aanmaken voor het omzetten van de temperatuur. Eén functie voor het omzetten van Celcius naar Fahrenheit en één functie voor het omzetten van Fahrenheit naar Celcius.



Opdracht 2: Steam Games (7pt)

Voeg functionaliteit toe aan de `Opdracht2` component die het mogelijk maakt om een lijst van Steam games te tonen. De data van de games kan je ophalen van de volgende URL: `https://raw.githubusercontent.com/similonap/json/refs/heads/master/steam.json`.

Je kan de volgende interface gebruiken voor een `SteamGame`:

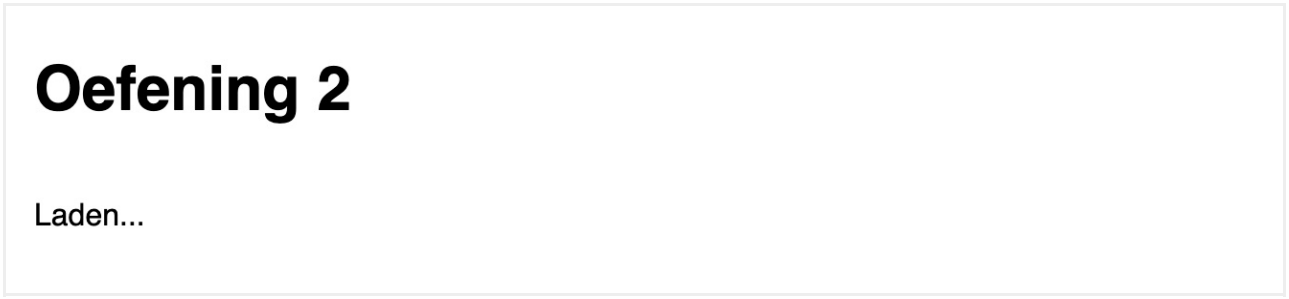
```
interface SteamGame {
    releaseYear: number;
    minimumAge: number;
    name: string;
    description: string;
    image: string;
    developer: string;
    platforms: PlatformDictionary
}

interface PlatformDictionary {
    windows: boolean;
    mac: boolean;
    linux: boolean
}
```

- [ ] Plaats deze interfaces in een apart `types.ts` bestand en zorg ervoor dat ze correct geïmporteerd worden in de `Opdracht2` component.
- [ ] Als de pagina zichtbaar wordt moet er een `fetch` request gebeuren naar de bovenstaande URL.
- [ ] Toon een `Laden...` boodschap zolang de data aan het laden is.
- [ ] Toon de games in een lijst met afbeeldingen. De afbeeldingen kan je tonen door de `image` property van de game te gebruiken.
- [ ] Voeg een input veld toe waarin de gebruiker kan filteren op de naam

van de game. De lijst met games moet automatisch gefilterd worden wanneer de gebruiker iets intypt in het input veld. De filter moet hoofdletter ongevoelig zijn.
- [ ] Voeg radio buttons toe waarmee de gebruiker kan filteren op platform. De gebruiker kan kiezen tussen Windows, Mac en Linux. De lijst met games moet automatisch gefilterd worden wanneer de gebruiker een ander platform selecteert. Sommige games kunnen op meerdere platformen beschikbaar zijn.
- [ ] Toon het aantal resultaten dat overblijft na het filteren.

Laden:

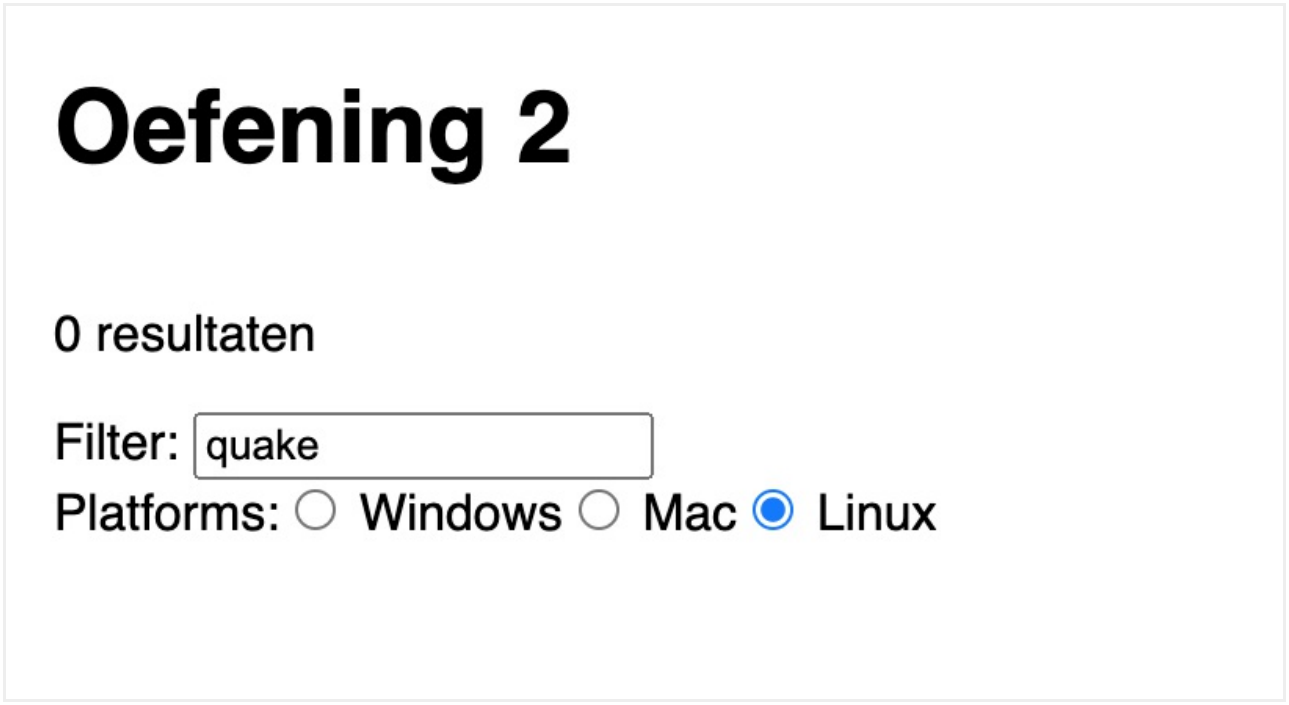


Filteren:

7 resultaten

Filter: quake

Platforms: Windows Mac Linux



Opdracht 3: Verkeerslicht (6pt)

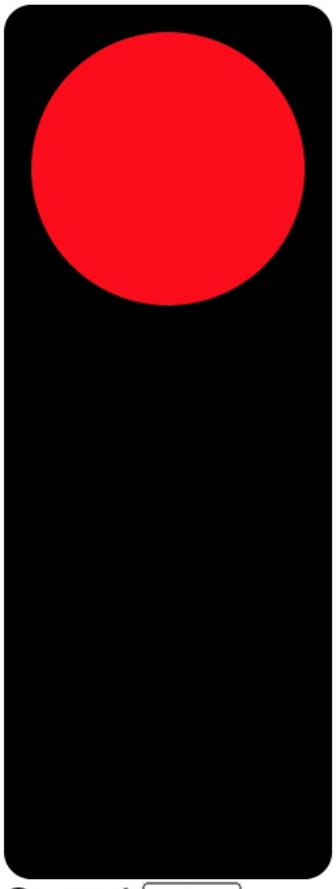
Voeg functionaliteit toe aan de `Opdracht3` component die een verkeerslicht simuleert. Het verkeerslicht heeft 3 standen: rood, oranje en groen. Het verkeerslicht moet elke 6 seconden van stand veranderen. De snelheid van het veranderen van stand moet instelbaar zijn door de gebruiker.

- [ ] Maak een component `Light` die de volgende properties heeft: `color` en `on`. `color` is een string die de kleur van het licht aangeeft en `on` is een boolean die aangeeft of het licht aan of uit is.
- [ ] Het `Light` component moet een div element zijn van 100x100 pixels met een borderRadius van 100. Dit zorgt ervoor dat het licht een cirkel is.
- [ ] De achtergrondkleur van het licht moet veranderen afhankelijk van de `color` property en de `on` property. Als `on` `true` is, moet de achtergrondkleur van het licht de kleur van de `color` property zijn. Als `on` `false` is, moet de achtergrondkleur zwart zijn.
- [ ] Plaats 3 `Light` componenten onder elkaar die de 3 standen van het verkeerslicht voorstellen: rood, oranje en groen.
- [ ] Hou de huidige stand van het verkeerslicht bij in een state.
- [ ] Gebruik een `useEffect` hook om de stand van het verkeerslicht elke 6 seconden te veranderen. De stand van het verkeerslicht moet veranderen in de volgorde rood -> groen -> oranje -> rood -> groen -> oranje -> ...

- [ ] Maak een select element waarmee de gebruiker de snelheid van het veranderen van stand kan instellen. De gebruiker kan kiezen tussen 1x, 2x en 4x. Als de gebruiker kiest voor 2x, moet het verkeerslicht elke 3 seconden van stand veranderen. Als de gebruiker kiest voor 4x, moet het verkeerslicht elke 1.5 seconden van stand veranderen.
- [ ] Zorg voor een cleanup functie voor de timeout/interval te clearen in de `useEffect` hook.
- [ ] Extra feature: het oranje licht moet maar 1/3 van de tijd branden van de andere lichten. Dit betekent dat het oranje licht maar 2 seconden moet branden in plaats van 6 seconden.

TIP: Je kan deze opdracht ook uitvoeren zonder een apart `Light` component te maken. Je kan de div elementen voor de lichten ook rechtstreeks in de `Opdracht3` component plaatsen. Je verliest dan enkel de punten op het gebruik van props.

Oefening 3



Inleveren

⚠ Checklist

- [ ] Verwijder de folder `node_modules` (zie boete systeem)
- [ ] Maak en .zip bestand van je project

Inzenden

Lever je project in als een **.zip** bestand op https://toets.ap.be.