

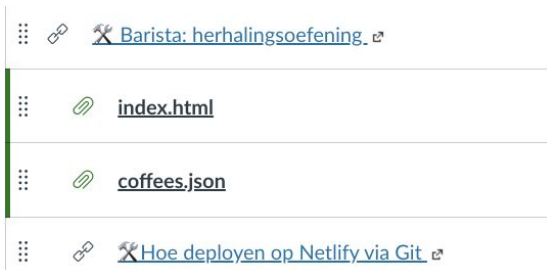


Herhalingsoefening

Plant-Based Barista app

Je krijgt van ons volgende files voor deze oefening:

1. **index.html**: hierin vind je de basisstructuur van de HTML + in commentaar de templates die je nodig hebt voor de JavaScript.
2. **coffees.json**: de koffie-data komt van een JSON file
3. Infodocument over hoe je dit Webpack project online kan zetten met GitHub + Netlify

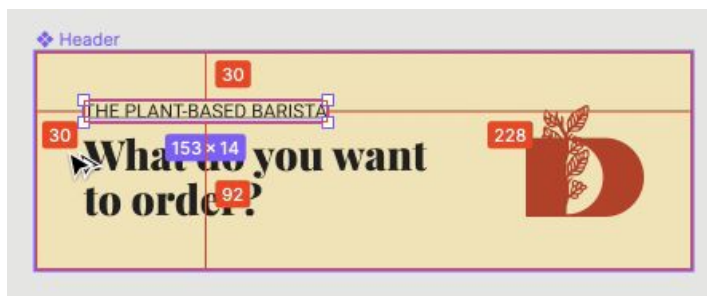


We hebben de voorbije weken al heel wat gezien. Tijd voor een herhalingsoefening dus. Probeer de volgende [The Plant-Based Barista app via het design in Figma](#) (zie ook screenshots verder in



dit document) zo goed mogelijk na te bouwen:

- Zorg dat er geen **eslint** of **stylelint** issues zijn: fix de errors tijdens het programmeren. Zie labo over Webpack voor meer info over *ESLint & Stylelint*.
- De website is **responsive**
- Gebruik **responsive images** (inclusief WebP) waar nodig
- Log je in Figma in om o.a. de pixel waardes, font-sizes en grids te bekijken en bestanden te exporteren:



- Integreer de volledige **Webpack config** voor geoptimaliseerde JavaScript & CSS files. Gebruik hiervoor het startproject met Webpack config uit de Webpack laboles.
- Maak een nieuw **private** project aan op **GitHub**. Voeg een `develop` branch toe en maak commits in deze branch per onderdeel dat je programmeert. Een onderdeel kan bijvoorbeeld zijn: het definiëren van je layout in CSS, het stijlen van de header, het inladen van de JSON file, ...
- Maak gebruik van de **PostCSS** features: variabelen, nesting en de nieuwe media query syntax
 - Denk aan de schaalbaarheid van je app: definieer bijvoorbeeld al je kleuren, font-sizes en spacers (marges en padding die de app gebruiken) in een `vars.css` bestand, definieer je layout in een `layout.css` bestand, etc.
 - Gebruik de BEM-notatie voor je CSS klassen
 - Merk op hoe de Figma is onderverdeeld in herbruikbare componenten, misschien kan je je CSS en HTML ook zo opbouwen en per component een CSS file aanmaken?



- JavaScript: zorg ervoor dat de items uit het menu dynamisch worden opgehaald via de meegeleverde JSON. Bekijk eerst aandachtig de structuur van de JSON-file.
 - De lokale JSON moet geïmporteerd worden omdat we gebruik maken van Webpack: die gaat de data dan ook automatisch importeren zodat je geen fetch meer moet doen (als je gebruik maakt van een PHP API / data uit de database ipv JSON file, werkt dit uiteraard niet). Dit kan als volgt gedaan worden (bovenaan index.js):

```
import data from './assets/data/coffees.json';
```

De data variabele bevat dan de inhoud van jouw JSON file. Log deze eens in je console.

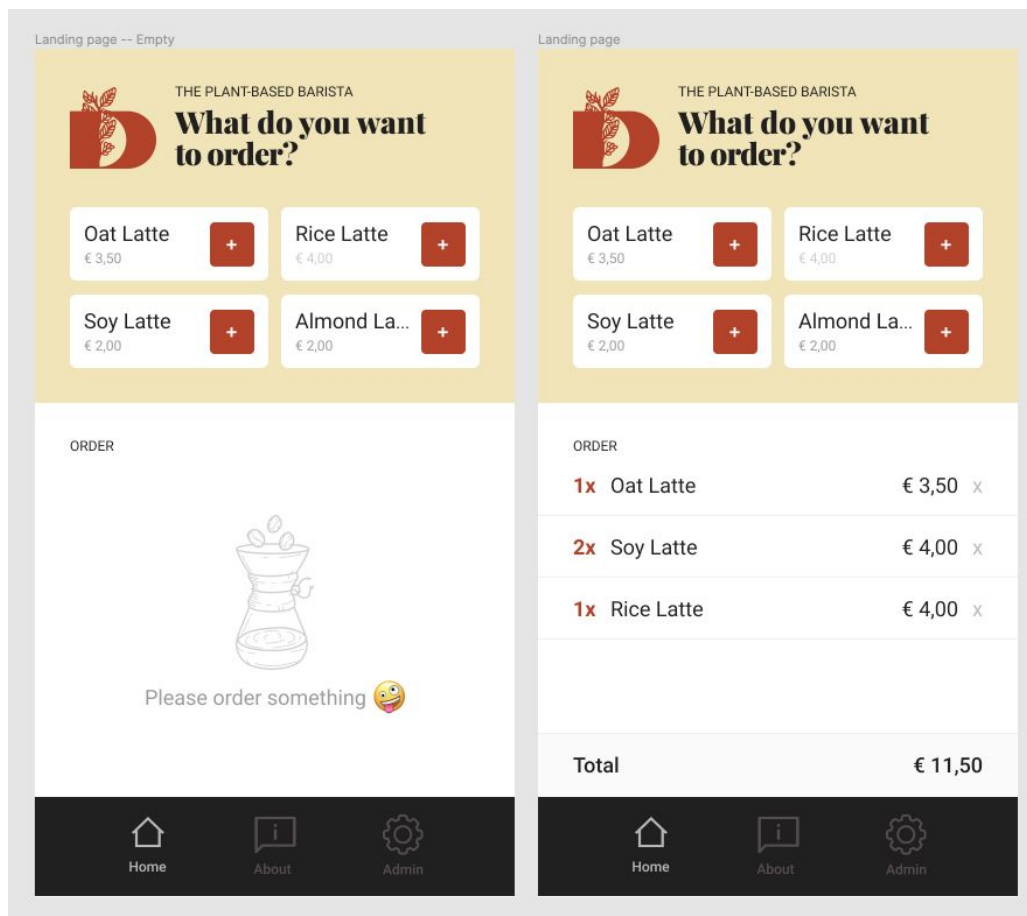
- Toon enkel de coffees die een plantbased: true property hebben
Voor elke koffie, voeg je een toe aan de .prices__list (zie ook template in de HTML). Sla het ID op van de koffie (tip: dataset), zodat je achteraf - wanneer je de koffie bestelt - ook de informatie over die koffie kan terugvinden (prijs, hoeveel keer besteld, ...)

```
><li class="price" data-id="1">...</li>  
><li class="price" data-id="2">...</li>  
><li class="price" data-id="3">...</li>  
><li class="price" data-id="4">...</li>
```

- Toon enkel de regular price. Goesting over? 🔥 Zorg ervoor dat je de koffie grootte kunt selecteren en implementeer het.
- Voorzie een **empty state** wanneer er nog niets besteld is ("*Please order something 😊*", zie design).
- Bij het klikken op de "+" knop van een koffie wordt de geselecteerde item toegevoegd aan lijst
 - Indien een item verschillende keren is aangeklikt wordt het aantal verhoogd
 - Je kan globaal bvb. een array orders globaal bijhouden dat per soort bestelde koffie het aantal keer dat het besteld is, bijhoudt



- De totaalprijs wordt telkens berekend en getoond bij een update van de order
- Alles gestyled? Alles geprogrammeerd? Je app werkt volledig bugvrij?
 - Merge je `develop` branch met je `main` branch zodat beide branches weer *even* staan (reminder: een merge naar `main` doe je enkel wanneer je app bugvrij is, als je aan het programmeren bent, werk je altijd in de *develop branch*).
 - Tijd om je `main` branch te **deployen**. Deploy het op Netlify. Hoe? [Meer lees je hier over](#).
 - Zet de url van jouw Netlify deploys in de `README.md` van je repo.
 - Merk je nog een bug op in je app? Wil je nog iets wijzigen? Switch dan terug naar je `develop` branch, fix de bug én merge uiteindelijk weer met `main`.
- [Dien dit project in op Leho](#): je hoeft enkel de link van je GitHub project in te dienen. (vergeet niet om `@benoitboucart` & `@newmovieclip` uit te nodigen op je GitHub project (via *Settings* -> *Manage Access*) → zie Leho assignment. Enkel eigen code. Geen copy / paste van iemand anders.
- Screenshots:



Probeer ook eens met een ander kleurthema door je CSS variabelen aan te passen, et voilà:



DEVELOPMENT III

