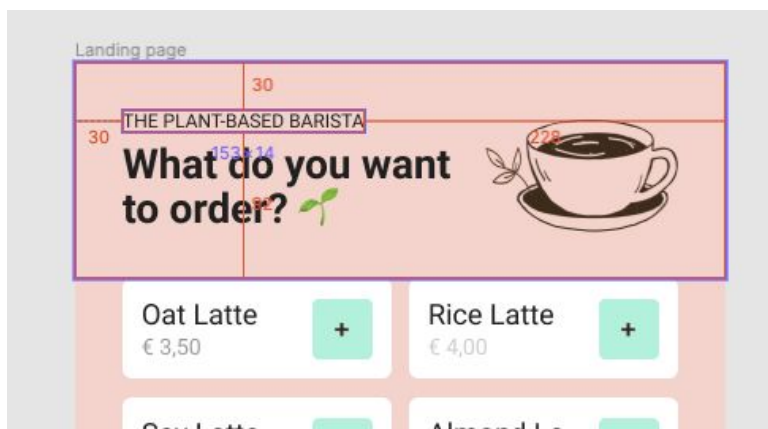


# Herhalingsoefening

## Barista app

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
- 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:



- Gebruik de volledige **Webpack config** voor geoptimaliseerde JavaScript, CSS files én images. Gebruik hiervoor het startproject met Webpack config.
- Maak een nieuw 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 Preset Env** 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?
  - Probeer je layout eens in CSS Grid én eens in Flexbox te maken. Maak hiervoor een `layout-flexbox.css` en een `layout-cssgrid.css` die je kan importeren en swappen in je `style.css` bestand.
- 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. 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.

Probeer ook eens `fetch` te doen van een externe API call. De API URL kan je hier terugvinden:

<https://api.myjson.com/bins/tcmfq>

- Toon enkel de coffees die een `plantbased: true` property hebben  
Voor elke koffie, voeg je een `<li>` 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? 🔥 Ontwerp een component waarbij je de koffie grootte kunt selecteren en implementeer deze.
- 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 de order
  - 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 gestijld? Alles geprogrammeerd? Je app werkt volledig bugvrij?
  - Merge je **develop** branch met je **master** branch zodat beide branches weer even staan.
  - Tijd om je **master** branch te **deployen**. Deploy het zowel op Now als op Netlify.
  - 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 **master**.
- Screenshots:

