

React Webshop

### Introductie

Voor deze opdracht willen we dat je een complexe webshop bouwt. Deze webshop hoeft geen echte producten te verkopen - welke nep producten er op jouw webshop te koop staan mag je dan ook zelf invullen. Wel moet aan een aantal randvoorwaarden voldaan worden.

Voor deze opdracht heb je 2 sprints van 2 weken. De eerste sprint kan je zien als een voorbereidende fase, zodat je in de 2e sprint een heel duidelijk idee hebt wat je gaat maken en hoe je dat wil doen.

### Randvoorwaarden

Deze randvoorwaarden zijn als volgt:

* **Meerdere pagina’s** - *minstens* het volgende:
  + Home Page
  + Product Overview Page
  + Product Details Page
  + Shopping Cart - deze mag je ook inbouwen via een sidebar i.p.v. aparte pagina
* **User Stories** van alle functionaliteiten
* **Wireframes** van elke pagina.
* **Sitemap** die de indeling van de verschillende pagina’s weergeeft.
* En natuurlijk moet je website ook goed **responsive** zijn.

Het is prima als je nog niet al deze begrippen kent - verderop wordt het kort toegelicht.

### Tech Stack

De *Tech Stack* van een project is de verzameling van alle gebruikte libraries, talen/frameworks, tools etc. Bij sommige projecten mag je de tech stack zelf bepalen, maar vaak wordt het (deels) opgelegd. Zo ook bij dit project:

* **Verplicht** is het gebruik van **React Router**. Dit is een bekende library die meerdere pagina’s als componenten gebruikt laat worden. De documentatie van React Router kan je [hier](https://reactrouter.com/) vinden.
* **Ook verplicht** is het gebruik van **Redux.** Dit is een ontzettend bekende library die voor state management gebruikt wordt. De complexiteit ervan is wel vrij hoog!
* **Sterk aangeraden** is het gebruik van **Firebase.** Dat is een platform van Google dat ontzettend veel functionaliteit heeft voor het ontwikkelen, publiceren en onthouden van je eigen gemaakte websites.

Firebase en Redux worden verderop nog toegelicht.

### Planning

Het is misschien verleidend om gelijk te beginnen met maar wat te proberen / coderen. Toch is dit niet verstandig. Als je jezelf goed voorbereidt is het uiteindelijk veel beter te doen.

Voor dit project **moet** deze planning in de eerste Sprint afgemaakt worden. Deze voorbereidende fase bestaat uit het volgende:

* User Stories, Sitemap en Wireframes zijn gemaakt
* Indeling van je componenten is gemaakt
* Tech Stack is onderzocht en mee geoefend.
* Fonts/logo/kleurenpatroon zijn gekozen.

Zoals je bij de vorige opdracht al hebt gemerkt is de structuur van je React componenten erg belangrijk. Eerder heb je met HTML een schets gemaakt van de structuur van een pagina - nu kan je dat ook met React doen. Dit is niet verplicht, maar mocht je er moeite mee hebben helpt dit wel.

Als je het lastig vindt om een font/logo/kleurenpatroon te verzinnen, kopieer er dan een van een veelgebruikte website. Let op dat je dan wel goed duidelijk maakt dat het om een nepversie gaat om copyright problemen te voorkomen.

### User Stories

User stories ben je inmiddels wel bekend mee. Misschien heb je eerder kort opgeschreven wat een project moet doen - dat is ook bij het maken van user stories een belangrijke stap. Schrijf alles wat je te binnen schiet qua eisen / functionaliteiten op! Deze informatie wordt vervolgens tot user stories verwerkt. Qua format zien die er zo uit:

**Als** *<soort user>* **wil ik** *<actie / functionaliteit>* **zodat ik** *<doel>*

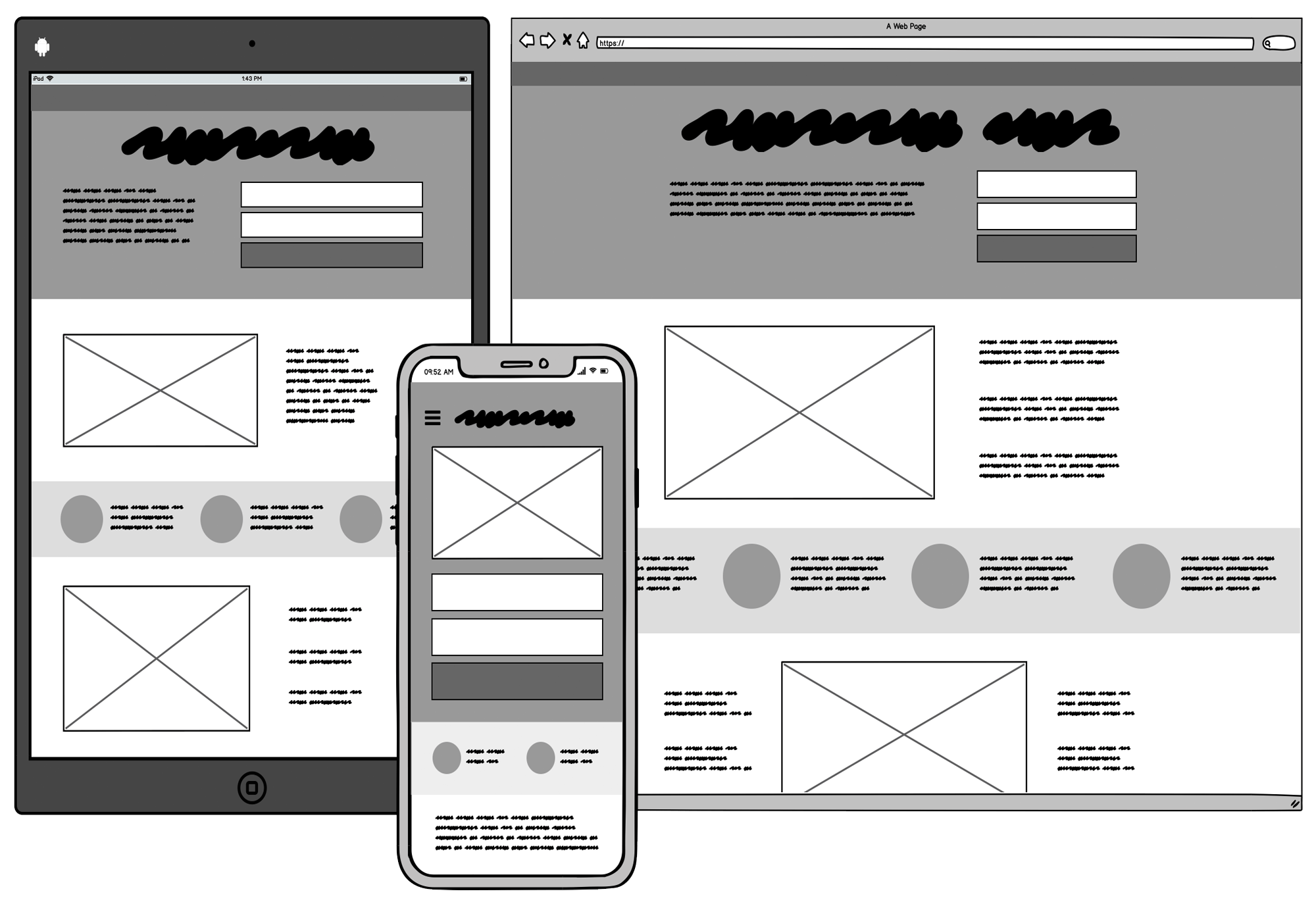
Voorbeelden van user stories:

* Als bezoeker van de website wil ik in kunnen loggen, zodat de website mijn adresgegevens kan bewaren.
* Als deelnemer aan TechGrounds wil ik user stories begrijpen, zodat ik een goede web developer word.

Ook bij grote bedrijven is dit de norm, hoewel het doel soms wordt weggelaten. Maak voor elke functionaliteit die je maar kan bedenken een bijpassende user story. Vervolgens kan je deze user stories gebruiken om prioriteiten te stellen binnen je project.

### Wireframes en Sitemap

Een wireframe van een pagina is een ruwe schets van de pagina-indeling. Soms worden hier tools voor gebruikt, maar potlood/pen en papier is ook prima. Een wireframe ziet er zo uit:



Besteed niet langer dan 5-10 minuten aan de wireframe van een pagina.

Een sitemap is een overzicht dat laat zien hoe je pagina’s onderling gestructureerd zijn - welke pagina heeft welke subpagina, welke pagina linkt naar welke pagina etc. Sitemaps zijn er over het algemeen in drie verschillende vormen.

* Een sitemap die door web crawlers (bots) gebruikt wordt om je pagina te begrijpen
* Een sitemap die door bezoekers gebruikt wordt om je pagina te bekijken
* Een sitemap die door developers gebruikt wordt om je pagina’s te maken.

De sitemap die ik van jullie wil zien is er dus een van die laatste soort. Ook hierbij is pen/potlood op papier voldoende, en moet je er niet langer dan 5-10 minuten over doen.

### Firebase

Firebase is een platform van Google dat helpt bij het ontwikkelen van applicaties. Het is onderverdeeld in 3 categorieën:

* ***Build***
* ***Release & Monitor***
* ***Engage***

Elke categorie heeft zijn eigen hulpmiddelen, met meerdere versies voor iOs, Android en Web Development. Je kan er bijvoorbeeld ook webhosting mee regelen!

Een van de meest voorkomende use case van Firebase is om een inlogsysteem toe te voegen aan een website zonder een hele back-end applicatie te hoeven bouwen. Zoiets wordt ook wel *Backend-as-a-Service* (**BaaS**) genoemd. Voor kleine websites is dit gratis!

Als je wil oefenen met Firebase, kan je dus een inlogsysteem aan je webshop toevoegen. Waar je bij deze opdracht mee gaat werken is dus de ***Build*** categorie, en dan specifiek de ***Authentication*** tool. Vaak wordt dit gecombineerd met gebruik van de Realtime Database tool, maar dat is voor deze opdracht niet nodig.

Ga je Firebase aan je project toevoegen? Dat betekent wel dat je dit ook in je Sitemap en User Stories moet verwerken. Denk ook aan password reset functionaliteit en alle bijbehorende pagina’s!

Firebase kan je [hier](https://firebase.google.com/) vinden. Een hele goede guide ervoor kan je [hier](https://www.freecodecamp.org/news/react-firebase-authentication-and-crud-operations/) vinden.

### Redux

Tenslotte is er dan ook nog Redux. Redux is een veelgebruikte library die state management wat makkelijker maakt. Redux bestaat ook zonder React, maar wordt wel vaak in combinatie gebruikt. Als je beiden kent (en dit kan bewijzen) ben je dan ook ontzettend competitief op de arbeidsmarkt. Maar waar gebruik je Redux nou eigenlijk voor?

Als je een complexe state hebt die op veel verschillende delen in je pagina nodig is. Denk aan bijvoorbeeld de inhoud van de shopping cart! Dat is informatie die vaak op veel plekken weergegeven wordt - al die componenten hebben dus toegang nodig tot dezelfde state.

Dit soort state management is in principe wel mogelijk met alleen React. Wel zal het dan veel werk zijn om de informatie langs alle componenten door te geven totdat hij uiteindelijk bij de goede subcomponenten belandt. Daarnaast kan je ook in de problemen komen als je state door verschillende componenten tegelijk wordt aangepast. [Hier](https://medium.com/@fastphrase/when-to-use-redux-f0aa70b5b1e2) nog een goed artikel over wanneer Redux een goede oplossing voor je state management is.

Meer weten over Redux? De zeer uitgebreide documentatie kan je [hier](https://redux.js.org/) vinden.