

Integrale eindopdracht:

Frontend Programming

HB0-Bachelor Software Development

Michelle Royer
NOVI Hogeschool
Augustus 2022

Inleiding

Voor mijn integrale eindopdracht ben ik een vriendin gaan helpen, door een applicatie te bouwen waar zij yoga cursussen kan geven. In mijn probleemstelling zal ik dieper ingaan op het probleem en wat mijn applicatie hiermee oplost.

Tijdens de opbouw van mijn eindopdracht ben ik eerst begonnen met het vaststellen van het probleem en wat mijn applicatie hier dan allemaal nodig voor zou moeten hebben. Vervolgens ben ik verder gegaan met het template van mijn scriptie en vervolgens ben ik mijn figma project gaan bouwen. Ik weet overigens dat het hierbij de bedoeling was om dit eerst met de hand te doen, maar ik aangezien ik creatiever ben wanneer ik het visueel voor me zie en er meteen meer vorm aan kan geven, ben ik begonnen met mijn applicatie te bouwen in figma.

Als inspiratie heb ik mijn vriendin (de klant) gevraagd welke websites haar zelf heel erg aanspraken en heb ik haar wat voorbeelden gestuurd van ontwerpen die ik zelf heel mooi vond. Toen ik dit eenmaal aangeleverd kreeg ben ik aan de slag gegaan en toen ik na het figma ontwerp aan akkoord kreeg, ben ik aan mijn code gaan werken.

Hierbij wil ik ook toevoegen, dat ik het gebruik van Figma onwijs handig vind en dat dit mij ook heel erg heeft geholpen met mijn code. Het was hierdoor duidelijker wat er allemaal nodig was in het ontwerp en welke elementen ik hiervoor moest creëren.

Ik wil hierbij ook mijn excus aanbieden voor het late inleveren. Ik ben afgelopen jaar november naar het buitenland verhuisd wat alles al aardig op een hoop gooide, toen kreeg ik begin dit jaar een nieuwe baan en de afgelopen 2 maanden 4 keer van plek naar plek gevlogen voor werk. Doordat ik voor een nieuw bedrijf ging werken ging opeens grootste deel van mijn tijd naar werk (ook omdat ik daar veel nieuws moest leren) en had ik minder tijd voor de studie, maar omdat ik toch wel een goed project wilde inleveren (zeker omdat de applicatie ook daadwerkelijk gebruikt gaat worden) heb ik er iets meer tijd voor genomen om mijn applicatie en functioneel ontwerp te schrijven.

Inleiding	2
Functioneel ontwerp	4
Probleemstelling	5
Functionele en niet functionele eisen	6
Inspiratiebronnen	9
User Experience Design Explanation	10
Use Cases	11
Use case tabellen	13
Getekende wireframes	16
Schermontwerpen	21
Verantwoordingsdocument	26
Limitations:	29
Figma	31
Firebase	31
Admin login	31

Functioneel ontwerp

Voor het maken van een applicatie, moet je eerst precies weten wat je wilt gaan bouwen en hoe. Daarvoor moet je een aantal dingen op een rijtje hebben:

1. Probleemstelling: welke problemen wil je oplossen voor jouw gebruikers met deze applicatie en wat zijn de belangrijkste 4 à 5 kernkwaliteiten.
2. Use cases: kies vier kernkwaliteiten uit die je gaat uitwerken in use case tabellen. Hierin beschrijf je op gedetailleerde wijze welk proces de gebruiker doorloopt om zijn doel te behalen en hoe de applicatie hier zou moeten reageren. → denk hierbij ook na over gebruikersvriendelijkheid.
3. Systeemeisen: op basis van alle use-case informatie ga je een functioneel (wat het doet) en niet-functioneel (hoe het dit doet) ontwerp maken.
4. Wireframes: Elementen op de pagina.
5. Schermontwerpen: op basis van getekende wireframes kun je de schermen gaan ontwerpen

Probleemstelling

Een vriendin van mij is momenteel bezig met het geven van yogalessen. Nu dat zij begint te groeien met haar bedrijfje, wilt ze ook cursussen aanbieden. Zij heeft momenteel een website via waar mensen haar kunnen boeken voor massages of yogales, maar nu dat zij ook de kant op wilt gaan van cursussen geven, wilt ze hier een aparte website voor zodat ze het overzicht kan bewaren. Nu wil ik voor haar een applicatie maken, waarbij zij de cursussen kan gaan geven, een overzicht houdt en de website ook kan gebruiken om klanten te trekken. De backend van deze applicatie heb ik al gemaakt bij de backend richtlijn en deze wil ik nu verder uitwerken en afmaken tot een volledig werkende applicatie.

Wat moet de applicatie kunnen:

- Gebruikers moet een account kunnen aanmaken;
- Kunnen inloggen;
- Gebruikers moeten inzicht hebben in hun eigen account (gegevens & te volgen lessen);
- Admin moet een overzicht hebben van alle gebruikers;
- Admin moet content kunnen uploaden en verwijderen;

De API die ik uiteindelijk heb gebruikt is:

<https://lightning-yoga-api.herokuapp.com/>

Ik wist niet precies wat ik kon toevoegen aan mijn applicatie waarvoor ik een open-source API nodig had. Ik heb dus tijdelijk een pagina toegevoegd aan mijn applicatie, waarbij diverse yoga oefeningen te zien zijn. Of ik dat later daadwerkelijk ga gebruiken, kan ik nog niet met zekerheid zeggen, maar dan klopt het in ieder geval voor de opdracht.

Functionele en niet functionele eisen

Functionele eisen	Niet functionele eisen
De gebruiker kan een account aanmaken en de volgende gegeven invullen: - Voornaam - Achternaam - Email - Wachtwoord invullen	Als de gebruiker een account aanmaakt wordt er in de backend gekeken of het e-mail een @ bevat.
	Als een email al bestaat wordt hier een melding van gemaakt
	Wachtwoord moet 6 karakters bevatten
Wanneer de gebruiker zijn/haar wachtwoord is vergeten kan hij/zij via de "forgot password" knop zijn/haar wachtwoord resetten	De gebruiker wordt naar een andere pagina genavigeerd
De gebruiker voert zijn/haar email in en ontvangt een verificatie code	er wordt een verificatie code gestuurd naar het opgegeven email adres
Deze code kan hij/zij uiteindelijk gebruiken om een nieuw wachtwoord in te stellen	In de backend wordt het gewijzigde wachtwoord aangepast.
De gebruiker kan in zijn scherm zien dat zijn account is aangemaakt	Het account van de gebruiker wordt opgeslagen
Als het e-mail al bestaat krijgt de gebruiker een melding dat er al een account bestaat met dit email adres	De gebruiker wordt genavigeerd naar de juiste homepagina en alle informatie wordt beschikbaar gesteld
Vervolgens worden alle functionaliteiten van het platform beschikbaar	Het wachtwoord en email worden gecontroleerd in de backend of deze bestaat en welke rol hierbij hoort (admin/user)
Wanneer de gebruiker uitlogt,	Na uitloggen wordt hij/zij terug

hoeft hij/zij niet telkens opnieuw een account aan te maken, maar kan hij/zij gewoon inloggen	genavigeerd naar de homepagina.
Wanneer de gebruiker zijn/haar wachtwoord is vergeten is er de mogelijkheid om een nieuw wachtwoord aan te maken door middel van email verificatie	In de backend wordt de e-mail gecontroleerd en wordt er een link gestuurd waarmee de gebruiker een nieuw wachtwoord kan aanmaken
Wanneer de gebruiker eenmaal op de homepagina is heeft kan hij/zij navigeren naar verschillende pagina's	In de backend is ervoor gezorgd dat elke link leidt naar de juiste pagina
De gebruiker kan in zijn/haar account zien hoeveel procent is voldaan van de gehele cursus	Dit wordt bijgehouden in de backend door mate van ...

Functie-eisen	Niet functie-eisen
De Admin heeft de mogelijkheid om modules toe te voegen. Hiervoor admin de volgende gegevens aanleveren: <ul style="list-style-type: none"> - Naam van de module - Naar welke module dit gepusht moet worden - De content uploaden (kan in een apart scherm) - als laatste stap wanneer de admin op uploaden drukt, wordt dit toegevoegd/aangepast. 	Deze worden (wanneer eenmaal uploaded) toegevoegd aan de content van de user.
De gebruiker kan in zijn/haar account de modules zien, de stof lezen (theorie) en filmpjes bekijken.	De backend houdt de modules bij en zorgt er ook voor dat deze op de juiste plek zichtbaar zijn.
De gebruiker kan zijn/haar gegevens aanpassen door op edit te drukken	Wanneer deze gegevens zijn aangepast wordt dit naar de backend gecommuniceerd en opgeslagen

De admin kan de maandelijkse studentenaantal bijhouden en bekijken	De backend houdt het aantal aanmeldingen bij
De admin krijgt een melding wanneer in de admin overview van iedereen die via de contactpagina contact opneemt	de backend zorgt ervoor dat het bericht vanuit de contact pagina wordt gepushed naar de overview van de admin
Voor de gebruiker om toegang te krijgen tot de cursus kunnen ze zich onderaan aanmelden, de admin krijgt hier dan bericht van	in firebase kan de admin de gebruiker dan handmatig toevoegen
Voor het inloggen ziet de gebruiker een knop om in te loggen	
Functionele eisen	Niet functionele eisen
Na het inloggen verandert de knop naar uitloggen	
De gebruiker heeft de mogelijkheid om contact op te nemen via de website	Het bericht wordt verzonden naar het email adres van de admin
Admin kan inzien of een gebruiker nog actief is ja of nee	De backend houdt bij of de gebruiker actief is ja of nee (op basis van laatste inlog periode)
	Nadat de gebruiker is ingelogd, wordt hij/zij doorverwezen naar de account pagina en begroet met zijn/haar naam
Admin kan een gebruiker verwijderen	De gebruiker wordt uit de backend verwijderd en heeft geen toegang meer tot de cursus.
Wanneer de gebruiker een module heeft voltooid wordt deze lichtgrijs.	
Wanneer de gebruiker contact wilt opnemen kan hij/zij dit via de contactpagina doen en krijgt	In de backend van de applicatie wordt ervoor gezorgd dat deze aanvragen worden gepushed naar

hij/zij een melding dat zijn/haar bericht is verzonden.

de contactpagina van de admin overview.

Inspiratiebronnen

Your Sisterhood to Luscious Living

Glo is een self-empowerment en holistic health community voor vrouwen die ten volle willen leven en schitteren. Een plek om in sisterhood te groeien, helen, verbinden en samen te werken aan persoonlijke en professionele doelen.

OVER ONS



Ervaringen van onze coachees

Hierbij heb ik inspiratie opgedaan voor de homepagina en andere basis pagina's zoals de over ons en contact pagina.



MIJN CURSUS

INLOGGEN TERUG NAAR HOUSEOFMOTHERS.NL



HOI & WELKOM BIJ DE GEBOORTECURSUS BEVALLEN ALS EEN BAAS!

Wil je meer informatie over deze beroemde cursus of wil je je inschrijven? [Klik hier.](#)

Wil je inloggen en aan de slag?

Doe dat door de knop hieronder.

[Inloggen](#)

Hiervan kon ik niet echt meer laten zien (wat ik daadwerkelijk als

inspiratie van de website had gebruikt), omdat je hiervoor ingelogd moet zijn en ik het via mijn vriendin had bekeken. Hierbij heb ik voornamelijk gekeken naar de manier hoe de gebruiker rondom eigen account en cursussen navigeert.

The screenshot shows a user's learning progress for a 'PRO SKILL PATH' titled 'Create REST APIs with Spring and Java'. The progress bar indicates 35% completion. Below the title, there is a section titled 'Conditionals and Control Flow' containing three items: 'Lesson' (checkmark icon), 'Lesson' (book icon), and 'Quiz' (checkmark icon). At the bottom of the path, there are two buttons: 'View Syllabus' and 'Resume Learning'.

Hierbij vond ik het handig dat je je voortgang kunt bijhouden en daarbij kunt zien hoeveel procent je al hebt voldaan.

User Experience Design Explanation

Een ontwerp gebruiksvriendelijk ontwerpen kun je het beste afmeten aan vijf onderdelen:

- **Learnability:** hoe makkelijk is het voor gebruikers om taken uit te voeren als ze het systeem voor het eerst gebruiken?
- **Efficiency:** als gebruikers het systeem eenmaal kennen, hoe snel kunnen ze hun taken dan uitvoeren?
- **Memorability:** Als gebruikers het systeem een tijdje niet hebben gebruikt, hoe makkelijk is het dan om hun kennis weer terug te halen?

- **Errors:** hoeveel fouten maken gebruikers, hoe ernstig zijn ze en hoe gemakkelijk kunnen ze hun fouten weer terug draaien?
- **Satisfaction:** is het systeem prettig in gebruik?

Bij het ontwerpen van je applicatie denk eraan om:

- Controle en vrijheid te bieden voor de gebruiker: biedt altijd de 'nooduitgang' zodat hij acties kan terug draaien. (cmd + z)
- Consistentie en standaardisering: zorg ervoor dat ervoor dat je duidelijke taal en tekens gebruikt in de taal van je applicatie en dat als je ergens iets 'wissen' noemt, dit later niet ergens anders 'verwijderen' noemt.
- Voorkom fouten: Goede foutmeldingen zijn cruciaal, maar het vermijden van de mogelijkheid dat een gebruiker een fout kan maken is nog veel belangrijker.
- Herkenning boven herinnering: verklein de geheugenbelasting zoveel mogelijk door acties en opties direct zichtbaar te maken. Zorg dat de functionaliteit niet te diep 'begraven' ligt.
- Efficiëntie: Zorg dat het systeem bruikbaar is voor zowel ervaren, als onervaren gebruikers.
- Esthetische en minimalistisch ontwerp: Zorg dat er nooit informatie getoond wordt die irrelevant is voor het proces dat de gebruiker aan het uitvoeren is. Dit concurreert namelijk met de informatie die wel essentieel is. Als we telkens bij het opslaan van een bestand in WebStorm een pop-up zouden krijgen met 'yes het is opgeslagen' is dat heel leuk, maar vertraagt dat de overige processen wel.
- Hulp bij fouten: Communiceer in normale taal wat voor fout er aan de hand is.
- Hulp en documentatie: Als het mogelijk is, help de gebruiker met hoe hij/zij de fout kan oplossen.

Use Cases

Een use case is een beschrijving van de manier waarop gebruikers taken op een website uitvoeren. Dit helpt je nadenken over wat het systeem moet kunnen/doen, maar ook over alle alternatieve scenario's die kunnen voorkomen op de het moment dat een gebruiker zijn doel probeert te bereiken.

Use case elementen :

- Het main success scenario - waarin niks mis gaat → the happy flow:
 - Titel: Welke functionaliteit wordt er beschreven?
 - Actor: Wie interacteert er met het systeem in deze use case? Is dat een gewone gebruiker of bijvoorbeeld een admin?
 - Precondities: wat moet er al gebeurd zijn of aanwezig zijn voordat deze use case kan starten?
 - Trigger: Welke trigger wordt uitgevoerd?: Als de gebruikers bv op de wijzig account-knop drukt
 - Postconditie: wanneer alles goed is gegaan, wat moet het eindresultaat dan zijn?
- De Alternatieve scenario's - de scenario's die op specifieke punten van het main success scenario afsplitsen en beschrijven wat er mis kan gaan in die situatie, hoe het systeem hierop inspeelt.
 - Titel
 - Nummering: Bij welke stap uit het main success scenario begint dit alternatieve scenario?
 - Postconditie: Op welke plek kan de gebruiker weer instromen?

USE CASE x	Titel
Actor	Benoem actor
Trigger	Benoem trigger
Preconditie	Benoem precondities
Main success scenario	1. Gebruiker doet... 2. Systeem doet... 3. Gebruiker doet... etc.
Postconditie	Benoem postconditie

ALTERNATIEF SCENARIO x	Titel
	1. Gebruiker doet...
	2. Systeem doet...
Postconditie	Benoem postconditie

Use case tabellen

Use case 1	Inloggen
Actor	Gebruiker
Trigger	De gebruiker zit op de webpagina en klikt op de login button.
Preconditie	De gebruiker moet al geregistreerd zijn
Main success scenario	<ol style="list-style-type: none"> 1. De gebruiker voert zijn email in 2. De gebruiker voert zijn wachtwoord in 3. De gebruiker klikt op inloggen 4. De backend checkt of alles klopt 5. De gebruiker is succesvol ingelogd en wordt automatisch naar de account pagina genavigeerd.
Postconditie	De gebruiker is ingelogd en kan de content volledig bekijken.

Use case 1	Alternatief scenario
1A	De gebruiker heeft geen '@' in zijn/haar e-mail geplaatst
1B	De gebruiker heeft een typfout gemaakt
1C	De gebruiker heeft de verkeerde gegevens ingevoerd
1D	De gebruiker is zijn wachtwoord vergeten
1E	De gebruiker drukt op de knop 'forgot password' en kan opnieuw een wachtwoord instellen
1F	De login knop werkt niet
1G	De wachtwoord vergeten knop werkt niet

Use case 2	Content toevoegen
Actor	Admin
Trigger	Admin wilt een nieuw onderwerp toevoegen (content)
Preconditie	Admin moet hiervoor wel de juiste bestanden in beschikking hebben
Main success scenario	<ol style="list-style-type: none"> 1. Admin gaat naar de admin overview page 2. De admin klikt op content 3. Vervolgens typt admin de naam van het bestand 4. Vervolgens welke module moet worden toegevoegd 5. Vervolgens wordt het bestand gesleept naar het upload gedeelte 6. Admin drukt op uploaden
Postconditie	Het bestand wordt uploaded en zichtbaar gesteld voor de gebruikers.

Use case 2	Alternatief scenario
2A	De gebruiker load het verkeerde bestand op en wordt niet geaccepteerd
2B	Het bestand is te groot
2C	Admin voegt het verkeerde bestand toe
2D	Admin heeft een typfout gemaakt
2E	Admin heeft per ongeluk het bestand verwijderd in plaats van toegevoegd.

Use case 3	Cursus volgen
Actor	Gebruiker/student
Trigger	De gebruiker wilt starten met de cursus/modules
Preconditie	De gebruiker moet wel een account hebben
Main success scenario	<ol style="list-style-type: none"> 1. De gebruiker logt in (uitgaande dat hij/zij al een account heeft) 2. de gebruiker wordt genavigeerd naar de account pagina waar hij/zij een overzicht kan zien van de cursussen en waar hij/zij gebleven was. 3. De gebruiker drukt op de benodigde module 4. De module opent en de gebruiker kan de les volgen.
Postconditie	

Use case 3	Alternatief scenario
3A	De aangeklikte module wilt niet open (bestand mist)
3B	De gebruiker kiest de verkeerde module
3C	De gebruiker heeft nog geen account
3D	De gebruiker is uitgelogd

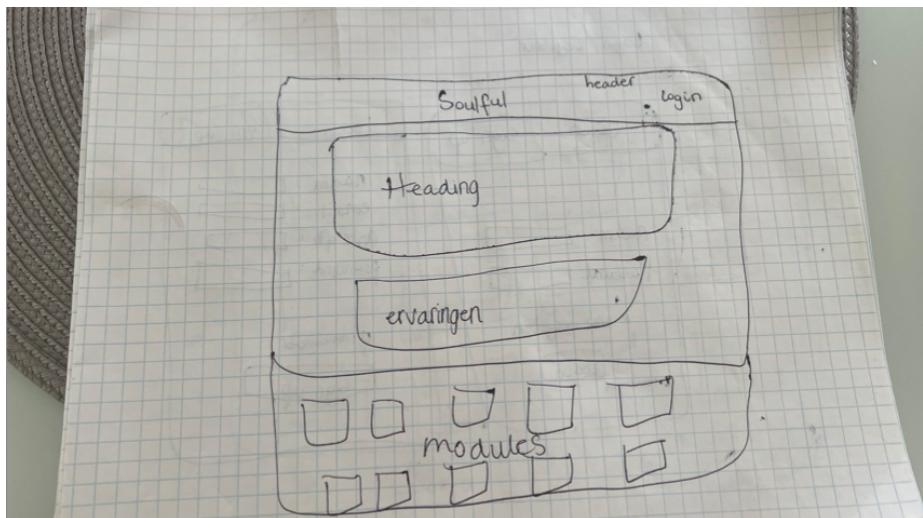
Use case 4	Account wijzigen
Actor	Gebruiker
Trigger	De gebruiker wilt zijn/haar gegevens wijzigen
Preconditie	Nieuwe gegevens hebben/foto wijzigen
Main success scenario	<ol style="list-style-type: none"> 1. De gebruiker gaat naar zijn/haar account 2. De gebruiker drukt op edit 3. De gebruiker verandert wat verandert wenst te worden
Postconditie	De gegevens zijn verwijderd

Use case 4	Alternatief scenario
4A	De foto die hij/zij probeert te veranderen bestaat niet uit de juiste file.
4B	De gebruiker voert de verkeerde gegevens in

Getekende wireframes

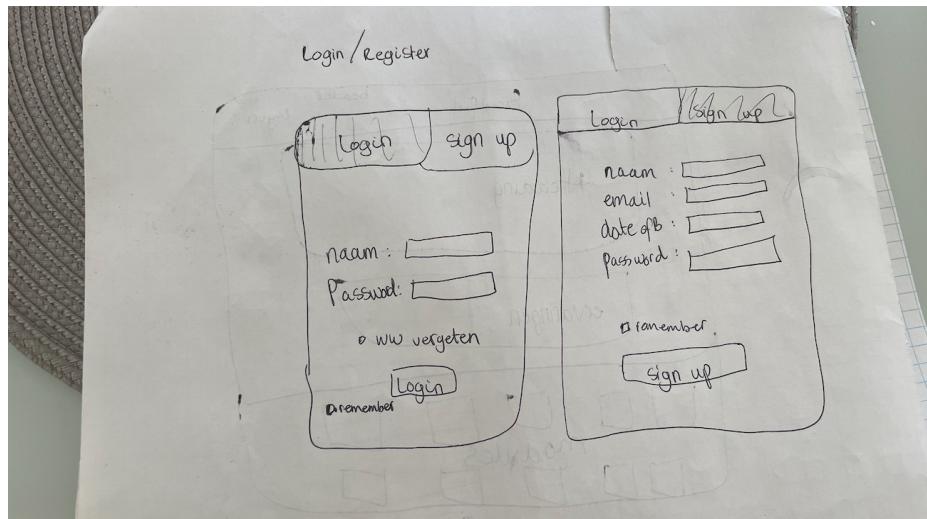
Homepagina

Dit is de homepagina voor het inloggen. De gebruiker krijgt een voorproefje van de website, maar kan niet de volledige inhoudelijke content bekijken.



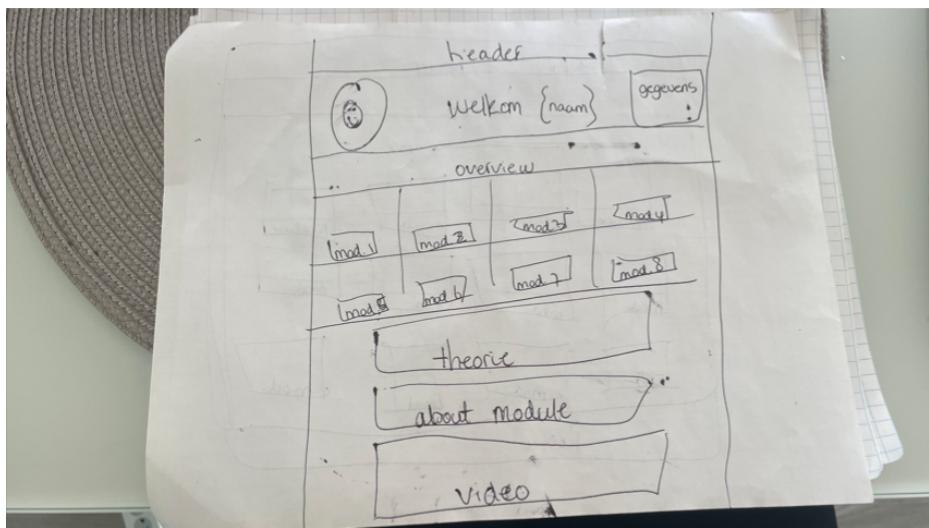
Login/Aanmeldpagina

Hier kan de gebruiker zicht inloggen. Wanneer de gebruiker de gebruiker meer informatie wilt bekijken wordt hij/zij automatisch hiernaartoe geleid. Hier kan de gebruiker inloggen, indien er al een account bestaat of een account aanmaken indien niet.



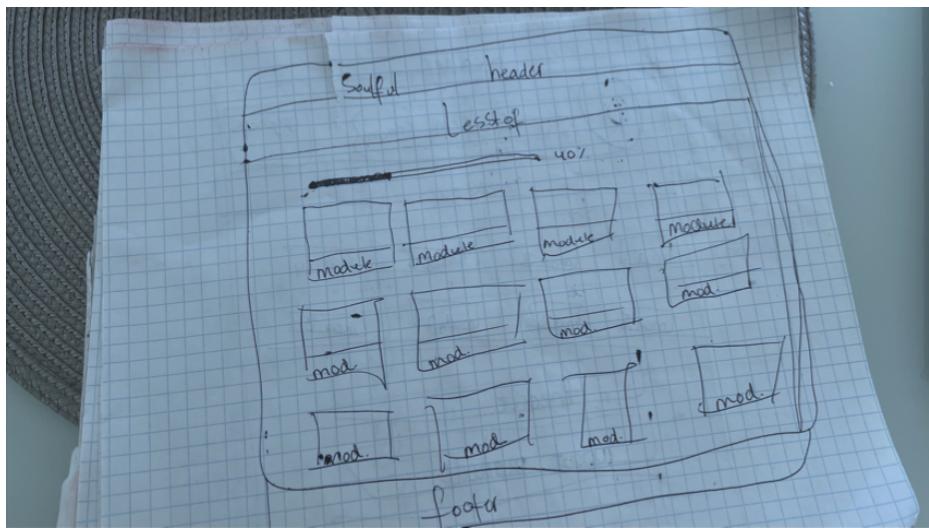
Account pagina

Dit is de account pagina. Hier kan de gebruiker zijn/haar gegevens inzien en wijzigen. Ook kun je de cursus hier volgen en heb je een overzicht van de modules, plus waar je gebleven bent.



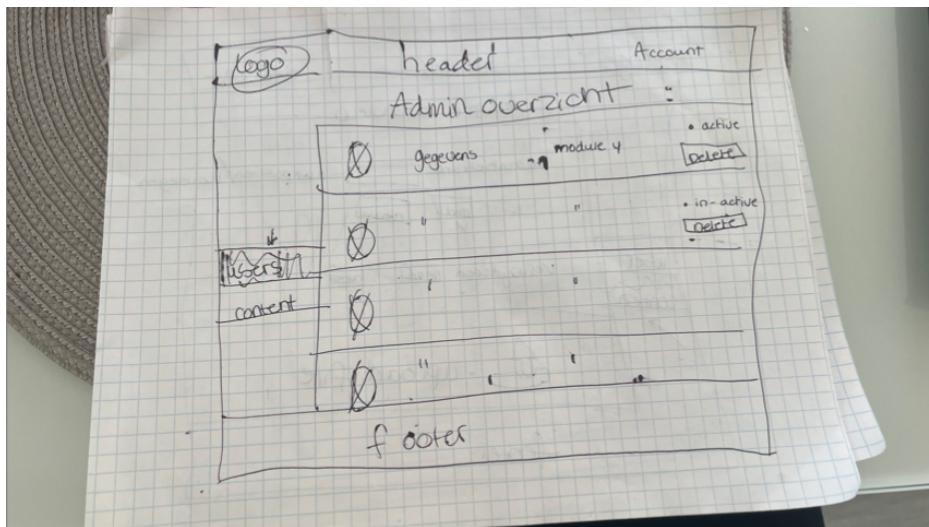
Les overzicht

Hier heeft de student een overzicht van alle lesstoffen plus kan hij/zij zien hoeveel percentage al gedaan is.



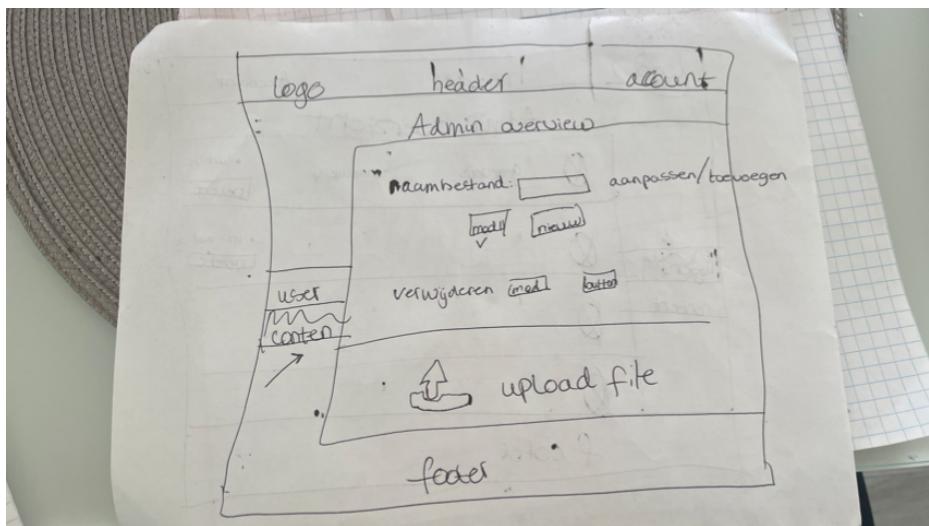
Admin overzicht

Dit is voor de admin, waarbij hij/zij alle studenten kan inzien en verwijderen.



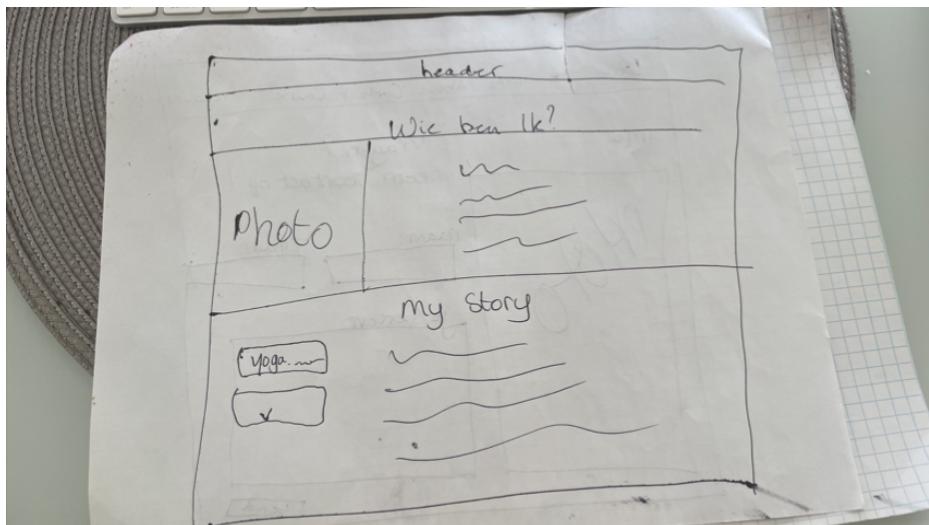
Content uploaden

De 'Admin Overview' heeft dus twee opties een is het inzien van de gebruikers en de andere is voor het uploaden van de content. Daar is deze pagina voor. Hier kan de admin een nieuwe module toevoegen, een oude wijzigen of verwijderen.



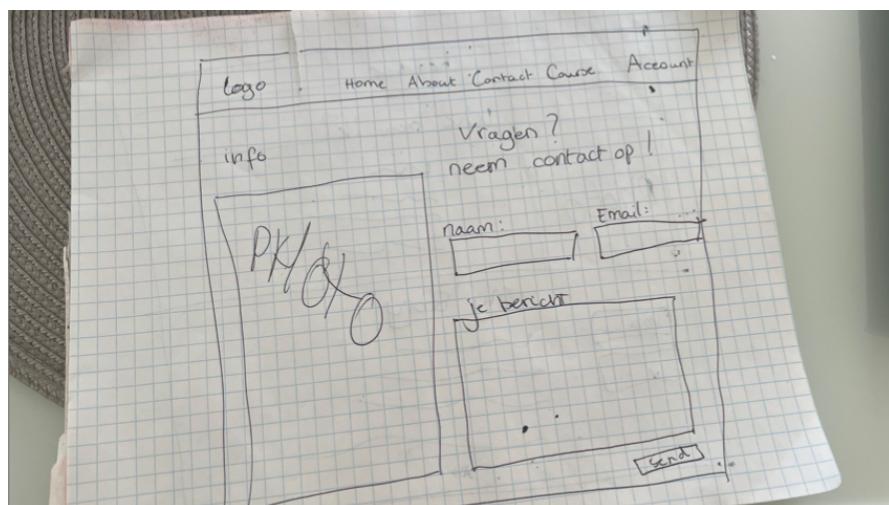
Wie ben ik?

Dit is de 'over ons' pagina. Waar mensen over Gytha (voor wie de website is) kunnen lezen en meer informatie krijgen over de persoon achter de cursussen.



Contact pagina

En dan *last but not least*, de contact pagina. Hier kunnen mensen contact opnemen met gytha bij eventuele vragen.



Schermontwerpen

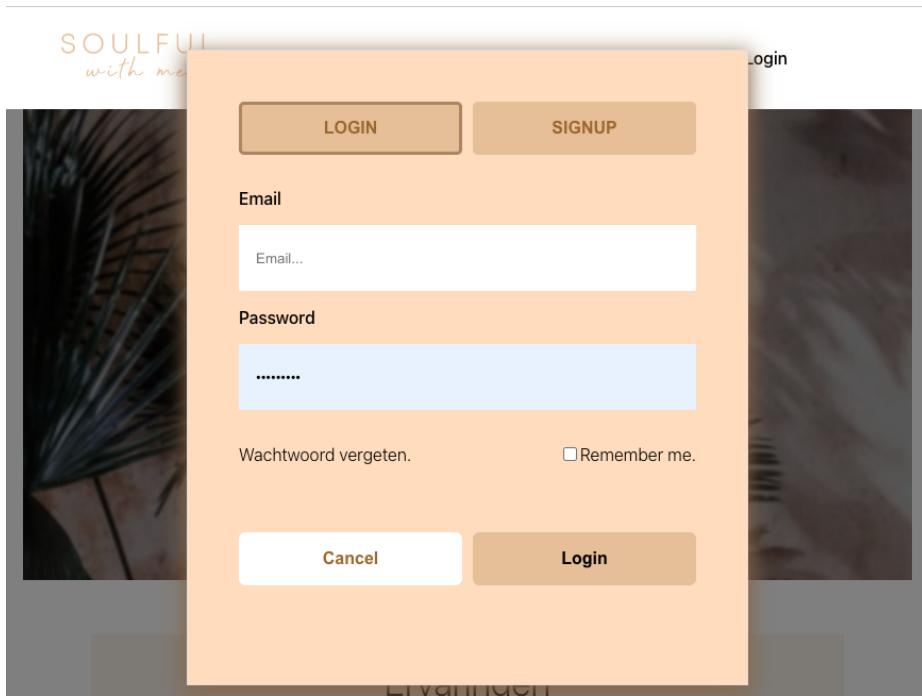
Homepagina

Dit is de homepagina voor het inloggen. De gebruiker krijgt een voorproefje van de website, maar kan niet de volledige inhoudelijke content bekijken.

The screenshot shows the homepage of a website called "SOULFUL with me". The header includes the logo "SOULFUL with me" and navigation links for "Home", "Over mij", "Contact", "Cursus", and "Login". The main visual is a photograph of tropical leaves. Overlaid text reads "SOULFUL LIVING" in large letters, with a small diamond-shaped logo below it. A button labeled "START DE CURSUS" is visible. Below this, a section titled "Ervaringen" contains placeholder text: "Lorem ipsum dolor sit amet consectetur, adipisicing elit. Praesentium, a nihil neque earum doloremque quos provident quas iusto nostrum debitis excepturi magni eius tempora aperiam asperiores blanditiis cumque! Suscipit, quas? 1". There are five small dots at the bottom of this section.

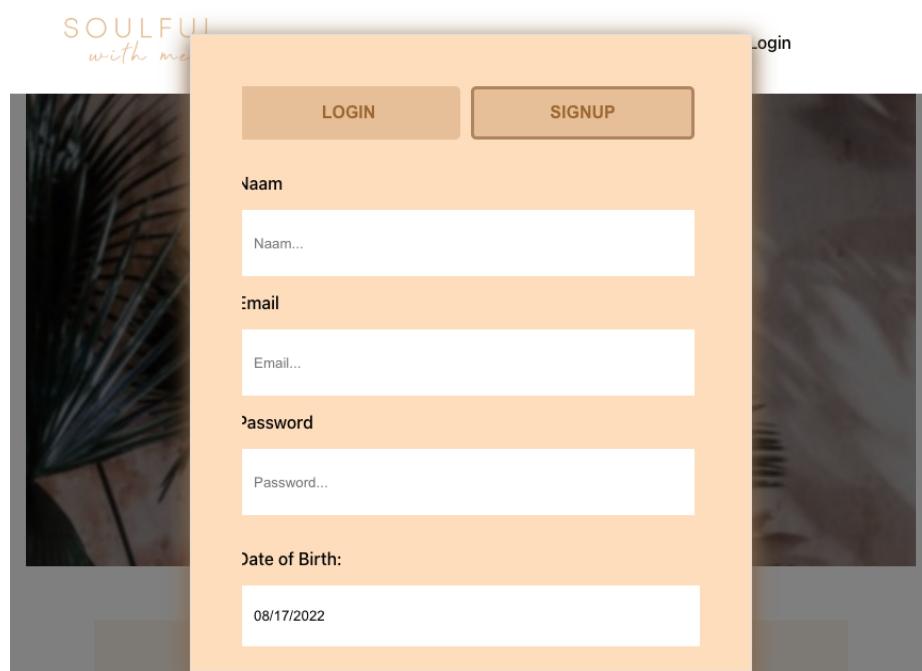
Login

Hier komt de gebruiker terecht zodra hij/zij meer wilt zien van de website/inhoudelijke informatie. Indien de gebruiker een account heeft, kan hij/zij hiermee inloggen.



Aanmelden

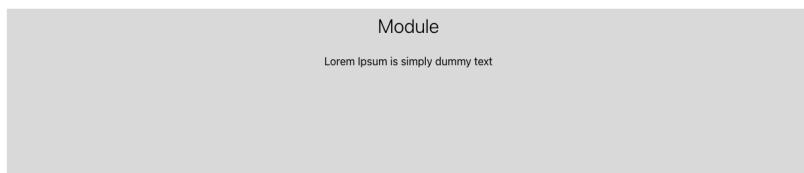
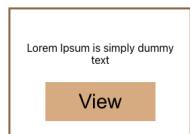
Wanneer de gebruiker nog geen account heeft aangemaakt, kan hij/zij door middel van het invullen van de gegevens een account aanmaken.



Account part 1

In dit gedeelte van de pagina kan de gebruiker inzien welke modules er zijn, waar hij/zij gebleven is, informatie over de module en de gebruiker kan zijn/haar gegevens wijzigen.

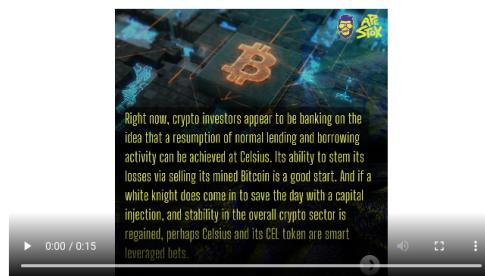
The screenshot shows the SOULFUL website's user profile page. At the top, there is a navigation bar with links for Home, Over mij, Contact, Cursus, and Logout. Below the navigation bar, there is a user profile section featuring a placeholder profile picture icon. To the right of the icon, the text "Welkom en succes met de cursus Michelle Royer!" is displayed. Further to the right, the user's name "Name: Michelle Royer" and email "Email: test@email.com" are shown, along with an "Edit" button. Below this section, a large orange header bar contains the text "Mijn overzicht".



Account part 1

Hier kan de gebruiker de gekozen content bekijken en bestuderen. Ik heb hier even (zoals te zien is) een random filmpje toegevoegd enkel om de werkzaamheid van de functie te checken.

A large gray placeholder box labeled "Theory" with placeholder text: "Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum."



Cursusoverzicht

Hier heeft de gebruiker een overzicht van alle modules die er in de cursus zijn en hoeveel percentage al voldaan is. Ik heb hier uiteindelijk het percentage weggehaald, deze werkte steeds niet goed en aangezien het niet nodig is voor de keuring van mijn applicatie heb ik die voor nu even eruit gelaten.

The screenshot shows a user interface for a course overview. At the top, there is a navigation bar with links: Home, Over mij, Contact, Cursus (which is underlined), and Logout. Below the navigation bar is a large orange header bar with the text 'Lesstof'. Underneath this, there is a progress bar indicating '40%' completion. A note below the progress bar says 'Lorem Ipsum is simply dummy text'. A 'View' button is visible at the bottom left of the main content area.

Admin Overview-User Overview

Ik heb deze pagina uiteindelijk een beetje uit bereid, omdat wat ik voorheen had bedacht in Figma nog een aantal dingen ontbrak. Dus voor de gebruiksvriendelijkheid heb ik er nog wat extra features aan toegevoegd. Op deze pagina kan de admin alle studenten bekijken, of hun account nog actief is en gebruikers verwijderen. Ik heb de online/offline functie eruit gehaald, omdat ik hem niet precies performend kreeg zoals ik dat wilde, dus dit is iets waar ik later nog aan wilt werken.

The screenshot shows a 'User Overview' section of the application. On the left, there is a sidebar with navigation links: User Overview (which is highlighted in orange), Content, Subscription, Contacts, and Dashboard. The main content area is titled 'Admin Overview' and lists five user profiles. Each profile includes the user's name, date of birth, email, course progress (represented by a red dot), and a 'Delete Account' button. The user profiles are: Michelle Royer (progress 40%), OZ (progress 10%), Elena Granlich (progress 10%), Romi (progress 10%), and Michelle Royer again (progress 10%).

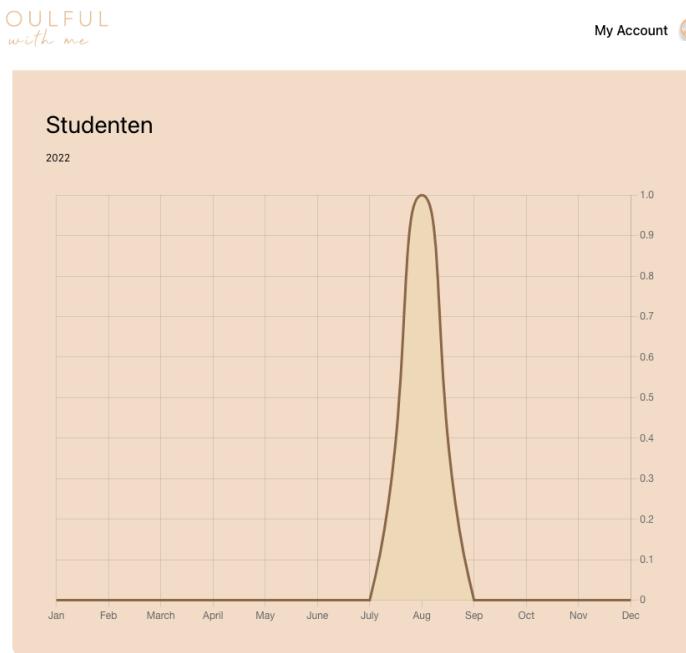
Admin Overview-Content

Hier kan de admin een module toevoegen, aanpassen of verwijderen. De gegevens die hier moeten worden meegegeven zijn: De naam, uitleg over de module (overall) en een stukje theorie.

The screenshot shows a web interface titled "Admin Overview". At the top, there are navigation links: "Home", "Over mij", "Contact", "Cursus", "Oefeningen overzicht", and "Logout". Below the navigation is a search bar with the placeholder "Back Pain Relie" and dropdown arrows, and two buttons: "Verwijderen" and "Aanpassen". The main content area is titled "Nieuwe cursus toevoegen" and contains three input fields: "Name:" (placeholder "Cursus titel..."), "Theorie:" (placeholder "Theorie van de cursus."), and "Cursus beschrijving:" (placeholder "Cursus beschrijving..."). On the left side, there is a sidebar with a "User Overview" section containing "Content", "Subscription", "Contacts", and "Dashboard" options. The overall design has a light orange and white color scheme.

Admin Overview-Dashboard

Hier heeft de admin een overzicht van het aantal ingeschreven studenten.



Verder is er nog "Subscription" en "Contacts"

Bij subscription heeft de admin een overzicht van alle mensen die zich hebben ingeschreven voor de nieuwsbrief. Bij contacts kan de admin inzien wie een aanvraag tot contact heeft gedaan via de contactpagina.

Verantwoordingsdocument

In dit verantwoordingsdocument heb ik een aantal pagina's beschreven waarbij ik wat lastigere code had, ik hooks heb geïmplementeerd die buiten de opleiding vielen (inclusief mijn reden hiervoor) en zal ik uitleggen als slot mijn limitaties tijdens het creëren van mijn applicatie. Omdat ik mijn applicatie ook daadwerkelijk heb gecreëerd zodat deze ook daadwerkelijk gebruikt kan worden is hij iets uitgebreider geworden dan gepland. Hopelijk is dit niet erg. En omdat mijn project zo groot is heb ik 10 willekeurige bestanden genomen om te beschrijven, waarbij ik .

In de CSS heb ik voornamelijk gebruik gemaakt van Flexbox, maar ook grid, omdat dit meer vrijheid biedt voor de layout.

Home

Aangezien dit de allereerste pagina was waar ik mee was begonnen, zal ik deze ook als eerst beschrijven. In deze pagina heb ik de useAuth hook gebruikt, omdat ik dit de makkelijkste manier vond om authenticatie aan mijn applicatie toe te voegen. Via de "useEffect" hook heb ik een if-else statement toegevoegd waarin wordt beschreven dat wanneer de "currentUser" inlogt met de admin-email hij de backend variant van de applicatie te zien krijgt zo niet (iedere andere gebruiker) wordt de gebruiker gepushed naar de 'normale' account pagina. Hierbij heb ik ook gebruik gemaakt van de useHistory hook, waarbij de juiste gebruiker naar de juiste pagina gepushed kan worden.

Navbar

Bij de "Navbar" heb ik ervoor gekozen om mijn logo aan de linkerkant te plaatsen, omdat, uit meerdere marketing onderzoeken, is gebleken dat gebruikers een website altijd bekijken van links naar rechts. Op deze manier weet de gebruiker dus direct op welke website ze zijn en de naam van het platform onthouden. Voor de gebruikers experience heb ik in de CSS de ".active-item" classname toegevoegd zodat de gebruiker duidelijk kan zien op welke pagina zij op dit moment zijn.

Toen ik mijn navbar document had gemaakt, kwam ik er later achter dat deze nog niet was aangepast naar mobiele/tablet view, dus moest ik dit later nog toevoegen en in de CSS hier aparte styling aan geven. Hier heb ik uiteindelijk in toegevoegd dat zodra het scherm

kleiner wordt, de schermindeling ook wordt aangepast en in plaats van dat het menu naast elkaar wordt weergegeven het nu in een "hamburger" veranderd en het een "drop-down" functie heeft.

Ik heb om deze reden in mijn CSS veel gebruik gemaakt van 'rem', omdat deze schaalbaar is (computerscherm, mobiel beeldscherm of tablet).

Modal

Het 'Modal' (ook wel overlay) bestand, deze heeft de naam omdat het over het beeld heen komt en ik dit hier wel passend bij vond. Hier zit de logica voor de pop-ups (login/register en edit). Ik heb hiervoor gekozen om de useState hook te gebruiken waardoor de waarde verandert zodra je op de "Button" drukt.

Login

De login pagina heb ik vrij simpel gehouden. Ik wilde hier eigenlijk ook facebook inloggen en etc. aan toevoegen, maar ik kwam erachter dat hier meer bij kwam kijken en het implementeren hiervan meer werk was dus heb ik dit voor nu even achterwege gelaten. In de login pagina heb ik ervoor gekozen om via react toastify meldingen te weergeven, zodat de gebruiker weet wanneer er iets misgaat of wat er is misgegaan.

ContactForm

De 'ContactForm' pagina heb ik als aparte bestand gemaakt zodat ik deze kan hergebruiken door mijn code en dus niet telkens opnieuw hoef te herschrijven. Ik heb hier ook weer gebruik gemaakt van toastify voor de meldingen en de database is hier ook geïmplementeerd zodat iedereen die contact opneemt dit wordt opgeslagen/genoteerd in de backend (firebase database). Deze heb ik vervolgens verwikkeld in een Fragment waardoor de functie ook daadwerkelijk verschijnt in de applicatie. Door middel van de onChange handler wordt de informatie die wordt ingevoerd in de invoervelden gerenderd en verwerkt.

Dashboard

Ik wilde dat de admin een overzicht had van alle maandelijkse inkomsten (en dit kon vergelijken met de voorgaande maand) en ook van alle studenten. De studenten van zowel het hele jaar als maandelijks. Wanneer de admin op de dashboard terecht komt krijgt

hij als eerst de ‘testimonials’ te zien. Dit geeft de admin een positieve blik over hoe de studenten zich voelen over de cursus. Voor het overzicht heb ik ervoor gekozen om de statistieken in een visuele chart weer te geven in plaats van tabel, omdat dit makkelijker leesbaar is en te vergelijken per maand. In de dashboard heb ik via de useEffect een hook geschreven waardoor deze pagina alleen zichtbaar is zodra de admin inlogt met de “admin@soulful.com” email adres. Wanneer dit niet zo is, wordt hij/zij door middel van de history.push naar de gebruikers account pagina gepushed.

Studenten

In het begin toen ik mijn figma beeld bekeek, wist ik nog niet precies hoe ik ook daadwerkelijk de grafiek moest weergeven. Uiteindelijk heb ik hier hulp bij gehad van een kennis en dit door middel van de <graph> class vormgegeven.

Graph

Hier moest ik chart.js importeren om zo gebruik te maken van de chart-functie en moest ik hier waardes aan geven om zo een zichtbaar/visueel voorbeeld te geven van het overzicht.

Account

Voor de account pagina wilde ik dat alles zo makkelijk mogelijk te gebruiken was voor de gebruiker en de gebruiksvriendelijkheid optimaal was. Hierbij vond ik het leuk/belangrijk dat iedere ingelogde gebruiker die naar de account pagina gaat wordt begroet met zijn/haar naam.

Ik heb de logica van deze pagina beschreven in de “Course Detail”, waarin ik via de useEffect hook de functie heb toegevoegd dat de pagina aanpasbaar is, zodra er op een van de modules wordt gedrukt (de pagina blijft hetzelfde, enkel de content veranderd). Verder heb ik doormiddel van ... de optie toegevoegd dat elke module die al is afgehandeld verder naar een lichtgrijze kleur. Dit helpt de gebruiker zodat duidelijk is waar hij/zij gebleven is.

Contact

Ook weer de contact pagina heb ik verdeeld in: de contact pagina (die de opbouw logica bevat), contact detail, contactInfo (waarbij ook de foto is ingevoerd) en de contactForm.

Ik heb op deze pagina door middel van “placeholders” aangegeven wat er in de invoervelden ingevoerd moet worden. De info die de klant hier invoert wordt automatisch toegevoegd aan de contact overview in de admin overview pagina.

Eerst had ik de persoonlijke email en telefoonnummer van de admin ook toegevoegd op de pagina, maar vanwege veiligheids redenen heb ik dit uiteindelijk toch verwijderd en de functie toegevoegd dat alles op de website te doen en te zien is. Naast het feit dat dit veiliger is, is dit ook makkelijker qua gebruiksvriendelijkheid omdat alles op de website te zien is en je dus niet naar een aparte applicatie hoeft.

Testimonials

De testimonials zijn zichtbaar op de homepagina en de admin dashboard. Dit heb ik op de homepagina aan het begin toegevoegd zodat andere gebruikers meteen een indruk krijgen van de cursus/website via andere/voorgaande gebruikers.

Meestal plaatsen websites de testimonials onderaan de pagina, maar omdat binnen de yoga community het vaker voorkomt dat mensen pas een cursus starten door middel van mond op mond reclame/voorgaande ervaringen dacht ik dat het handig was dit bovenaan de pagina te plaatsen. Ik heb hier door middel van de “simple slider” de functie toegevoegd dat meerdere testimonials zichtbaar zijn en zich automatisch afspelen/verschuiven.

Limitations:

Testimonials: Ik wilde hier nog een pagina toevoegen waarbij mensen hun testimonial kunnen schrijven of dat de admin dit automatisch kan pushen vanuit een email bestand, maar ik kwam hier verder niet meer uit en dit zou ook teveel tijd kosten.

Contact: Ik wilde hier nog een functie toevoegen dat de admin kan zien of de gebruiker die contact opneemt al een bestaande gebruiker is of een nieuwe gebruiker.

Ook wilde ik hier nog een functie toevoegen dat alle mails automatisch worden verzonden naar de daadwerkelijke e mail, maar had uiteindelijk geen tijd meer om uit te zoeken hoe ik dit daadwerkelijk moest doen. Ik twijfel hierbij nog of ik dit uiteindelijk wel ga doen aangezien het ook wel makkelijk is dat alles op de informatie op de website blijft.

Dashboard

Ik wilde hier ook nog de optie dat je de omzet kon bekijken en dit in vergelijking met vorig jaar, maar omdat ik nu geen betalingssysteem kon toevoegen moest ik dit weglaten.

Email Authenticatie: Hier zou ik later nog de functie willen toevoegen dat mensen ook kunnen inloggen via gmail, facebook etc.

Payment

Ik zou later nog een optie willen toevoegen dat mensen door middel van paypal, ideal of andere betaal mogelijkheden de cursus via de website kunnen kopen. Op dit moment kan namelijk iedereen een account aanmaken en dat is natuurlijk niet de bedoeling.

Progressie Bar

Ik heb de progressie bar even moeten weglaten omdat ik hem niet werkend kreeg op basis van hoeveel percentage de student al heeft voldaan van de cursus.

Signup

Voor nu heb ik de sign up knop wel erin gelaten, zodat je wel een account kan aanmaken en voor de user experience. Alleen moet ik hier nog wat waardes aan toevoegen, alleen dit kon ik pas doen als ik ook de betalingsoptie heb toegevoegd aan de website.

Inkomsten overzicht

Ik wilde in het dashboard voor de admin nog de functie dat de admin kan zien hoeveel geld er die maand is binnengekomen en dat ook in vergelijking van de voorgaande maand.

Figma

<https://www.figma.com/file/Qmh8sPWIETMEnuR7qYxQEp/SoulfulApp?node-id=9%3A240>

Via deze link kun je het figma bestand bekijken.

Firebase

Om gebruik te maken van de database, ga je naar deze website:

<https://console.firebaseio.google.com/>

Inloggegevens:

E-mail: michelle.reclame2@gmail.com

Wachtwoord: W3lkom-1234!

Hier krijg je toegang tot de backend. Hier kun je gebruikers verwijderen en toevoegen en content aanpassen en verwijderen.

Admin login

Email: admin@soulful.com

Wachtwoord: Admin123