

Functioneel Ontwerpen Hooikoorts Stadwijzer Friesland



Auteur: Michael Barak

Datum: 28-07-2025

Doel: Eindopdracht Front-end

Inhoudsopgave

A.	Inleiding.....	3
B.	Op te leveren producten.....	3
C.	Probleembeschrijving	4
a.	Kernfunctionaliteiten.....	4
D.	Gebruikte Case Tabellen	5
a.	Registratie en Inloggen	5
b.	Voorkeuren Instellen	5
c.	Locatiekeuze en Top 5	6
d.	Hooikoortsadvies.....	6
E.	Functionele Eisen	7
F.	Niet-Functionele Eisen	8
G.	Technische Architectuur	9
H.	Handgetekende Wireframes.....	10
I.	Schermontwerpen	18
J.	Broncode React Applicatie (Projectstructuur)	18
K.	Link naar Figma	19
L.	Conclusie	20

Inleiding

De Hooikoorts Stadwijzer Friesland is een webapplicatie, ontwikkeld in React, die specifiek is ontworpen om hooikoortspatiënten te ondersteunen in hun zoektocht naar de meest geschikte locaties binnen Friesland. De applicatie formuleert haar aanbevelingen op basis van actuele pollenconcentraties, weersomstandigheden en persoonlijke voorkeuren van de gebruiker. Door gegevens te combineren uit zowel de Ambee Pollen API als de OpenWeather API, ontstaat een aanzienlijk completer en nauwkeuriger beeld van de omgevingsomstandigheden, samen met praktisch onderbouwde gezondheidsadviezen. Authenticatie verloopt via de NOVI-backend, terwijl voorkeuren worden opgeslagen in LocalStorage. Er wordt geen conventionele database toegepast. De app slaat voorkeuren niet op via een server-side database, zoals MySQL, PostgreSQL, MongoDB of vergelijkbare systemen. Deze technologieën worden volledig buiten beschouwing gelaten. Deze applicatie voldoet aan de eisen van de eindopdracht Frontend en richt zich op een gebruiksvriendelijke, toegankelijke oplossing.

Op te leveren producten

Alle producten moeten worden ingeleverd als één ZIP-bestand (maximaal 50 MB), zonder `node_modules`, en voldoen aan de volgende specificaties:

- **Functioneel Ontwerp Document**
 - Bestandsformaat: PDF
 - Het document bevat alle vereiste onderdelen. Onder andere handmatig opgestelde wireframes (geleverd als afbeeldingen) en daarnaast een link naar de schermontwerpen in Figma. Alle onderdelen zijn zorgvuldig opgenomen, zodat aan de gestelde eisen wordt voldaan.
- **Broncode React Applicatie**
 - Volledige mappenstructuur, inclusief `public`, `src`, en configuratiebestanden (zoals `.env.example`, `package.json`, `vite.config.js`).
 - Geen `node_modules` of build-output (bijv. `dist` of `build`).
 - Die wordt als een zip-code toegestuurd
- **Installatiehandleiding**
 - Het projectbestand dient te worden opgeslagen als `README.md` in de root-map van het project.
 - Duidelijke instructies voor installatie, benodigdheden en gebruik.
- **Verantwoordingsdocument**
 - Geïntegreerd in het Functioneel Ontwerp Document.
 - Bevat technische keuzes, reflectie, limieten en doorontwikkelingen.
- **Figma-project met schermontwerpen**
 - De gedeelde Figma-link is beschikbaar en te vinden in het bestand `schermontwerpen-link.txt`, dat zich in de projectmap bevindt.

- **Handgetekende Wireframes**
 - Als afbeeldingen (bijv. PNG/JPG) ingevoegd in het PDF-document

Probleembeschrijving

Hooikoortspatiënten in Friesland ondervinden variabele klachten afhankelijk van lokale pollenconcentraties en weersfactoren zoals temperatuur, wind, luchtvochtigheid en UV-index. Huidige toepassingen beperken zich vaak tot losse informatie, bijvoorbeeld over pollen of het weer, en houden daarbij geen rekening met persoonlijke voorkeuren of advies dat specifiek is afgestemd op de locatie van de gebruiker. De Hooikoorts Stadwijzer Friesland lost dit op door real-time data te combineren met een dynamisch scoring-systeem en gepersonaliseerde tips, specifiek gericht op Friese steden. De Hooikoorts Stadwijzer Friesland is een innovatieve webapplicatie die specifiek is ontworpen om hooikoortspatiënten te ondersteunen bij het identificeren van de meest geschikte locaties in Friesland, rekening houdend met actuele pollen- en weersgegevens op basis van:

- Real-time pollenconcentraties (Ambee API)
- Weersomstandigheden (OpenWeather API)
- Gepersonaliseerde gebruikersvoorkeuren

Kernfunctionaliteiten

1. **Registratie en Inloggen:** Gebruikers registreren en loggen in via de NOVI-backend om voorkeuren veilig op te slaan.
2. **Voorkeuren Instellen:** Gebruikers configureren criteria (bijv. lage pollen, temperatuur tussen 15-25°C) en wijzen prioriteiten toe (1-5).
3. **Locatiekeuze en Top 5:** Gebruikers selecteren maximaal 10 Friese steden; de app berekent en toont de top 5 meest gunstige locaties.
4. **Hooikoortsadvies:** Per stad worden minimaal drie dynamische adviezen geleverd (bijv. medicatie, activiteiten, voeding) gebaseerd op API-data.

Gebruikte Case Tabellen

Registratie en Inloggen

- Veld: Beschrijving
- Use Case ID: **UC1**
- Naam: Registratie en Inloggen
- Beschrijving: Gebruiker registreert of logt in via NOVI-backend.
- Actoren: Gebruiker, NOVI-backend
- Voorwaarden: Internetverbinding, geldige inloggegevens

Happy Flow:

1. Gebruiker opent registratiepagina.
2. Vult e-mail, naam, wachtwoord in.
3. Ontvangt verificatielink per e-mail.
4. Klikkt op link en logt in.
5. Dashboard wordt getoond.

Alternatieve Scenario's:

- A1: Ongeldig e-mailformaat → Foutmelding "Voer een geldig e-mailadres in".
- A2: Zwak wachtwoord – Toon de foutmelding: "Het wachtwoord moet minimaal 8 tekens lang zijn."
- A3: E-mail bestaat al → Er wordt een voorstel gedaan om nog een keer in te loggen

Voorkeuren Instellen

- Veld: Beschrijving
- Use Case ID: **UC2**
- Naam: Voorkeuren Instellen
- Beschrijving: Gebruiker stelt weer- en gezondheidsvoorkeuren in.
- Actoren: Ingelogde gebruiker
- Voorwaarden: Gebruiker is ingelogd, internet (optioneel)

Happy Flow

1. gebruiker opent voorkeurenpagina.
2. Selecteert criteria (bijv. lage pollen < 2).
3. Wijst prioriteiten toe (1-5 via sliders).

4. Systeem slaat op in LocalStorage.

Alternatieve Scenario's

- B1: Geen criteria geselecteerd → Foutmelding "Selecteer minimaal één criterium".
- B2: Onrealistische combinatie (bijvoorbeeld 100% luchtvochtigheid zonder wind)
- B3: Offline → Opslag in LocalStorage, synchronisatie bij online. (Voorkeuren worden tijdelijk opgeslagen in localStorage onder 'In afwachting van voorkeuren' en wanneer u weer online bent, worden de voorkeuren in afwachting gesynchroniseerd)

Locatiekeuze en Top 5

- Veld: Beschrijving
- Use Case ID: **UC3**
- Naam: Locatiekeuze en Top 5
- Beschrijving: Gebruiker kiest steden en krijgt top 5.
- Actoren: Gebruiker, Ambee API, OpenWeather API
- Voorwaarden: Voorkeuren ingesteld, API's beschikbaar

Happy Flow

1. Gebruiker selecteert 3-10 steden.
2. Systeem haalt data op via API's.
3. Berekent scores met scoreCalculator.jsx.
4. Toont top 5 met scores.

Alternatieve Scenario's

- C1: Geen steden geselecteerd → Foutmelding "Kies minimaal 1 stad".
- C2: API-fout → Gebruik gecachte data.
- C3: Gelijke scores → Sorteer op luchtkwaliteit/UV-index.

Hooikoortsadvies

- Veld: Beschrijving
- Use Case ID: **UC4**
- Naam: Hooikoortsadvies
- Beschrijving: Gebruiker bekijkt adviezen per stad.
- Actoren: Gebruiker
- Voorwaarden: Top 5 beschikbaar

Happy Flow:

1. Gebruiker kiest stad uit top 5.
2. Systeem toont pollen-/weergegevens en 3+ tips.
3. Gebruiker slaat op of deelt via e-mail/WhatsApp.

Alternatieve Scenario's:

- D1: Geen voorkeuren → Standaardadvies gebaseerd op algemene richtlijnen.
- D2: Extreem weer (bijv. storm) → Specifiek advies "Blijf binnen".
- D3: Delen → Genereer samenvatting en kopieer naar klembord.

Functionele Eisen

1. Gebruikers kunnen registreren en inloggen via NOVI-backend (endpoints: /auth/register, /auth/login).
2. Gebruikers stellen voorkeuren in voor minimaal vijf criteria (pollenconcentratie, temperatuur, UV-index, windkracht, luchtvochtigheid) met prioriteiten (1-5).
3. Applicatie berekent scores voor 3-10 Friese steden en toont de top 5, gesorteerd op score.
4. Per stad worden minimaal drie dynamische gezondheidsadviezen getoond,
5. gebaseerd op API-data (bijv. medicatie, activiteiten).
6. De applicatie verzamelt actuele gegevens door gebruik te maken van twee bronnen: weersinformatie wordt verkregen via de OpenWeather API, terwijl pollengegevens worden opgehaald uit de Ambee Pollen API.
7. Ondersteuning voor delen van adviezen via e-mail, WhatsApp of klembord.
8. Gebruikers kunnen voorkeuren wijzigen en opslaan na inloggen.
9. Applicatie valideert e-mailadressen bij registratie (bijv. "@-teken, domein").
10. Wachtwoorden moeten minimaal 6 karakters bevatten, inclusief een cijfer en hoofdletter.
11. Applicatie toont een foutmelding bij ongeldige inloggegevens.
12. Gebruikers kunnen maximaal 10 steden selecteren voor scoreberekening.
13. Adviezen worden automatisch aangepast bij nieuwe API-data.
14. Gebruikers kunnen een stad uit de top 5 selecteren voor gedetailleerde weergegevens.
15. Gebruikers kunnen advies bij elke stad kopen.
16. Applicatie biedt een zoekfunctie voor Friese steden.
17. De gegevens worden elk uur automatisch vernieuwd.
18. Gebruikers krijgen een waarschuwing bij het verlaten van de voorkeurenpagina zonder opslaan.
19. Applicatie ondersteunt het wissen van voorkeuren

20. Door middel van deze Gebruikers (users) kunnen eigen profiel bekijken en ook bewerken nadat ze ingelogd zijn.
21. Applicatie toont een laadscherm tijdens API-opgravingen.
22. Foutmeldingen zijn specifiek en bevatten herstel instructies.
23. API-responsen worden gelogd voor debugdoeleinden.
24. Biedt de applicatie een functie waarmee gebruikers hun voorkeuren kunnen resetten naar de standaardinstellingen.
25. Gebruikers kunnen een waarschuwing ontvangen dat alle data wordt gereset.
26. Met behulp van CSV-bestand kan dit applicatie het importeren van steden ondersteunen
27. Deze applicatie ondersteunt ook het exporteren van geselecteerde steden (b.v. Groningen of Dokkum...ect.) via een CSV-bestand.
28. De applicatie maakt het mogelijk om een sjabloon met verschillende steden te downloaden

Niet-Functionele Eisen

1. Responsief design, compatibel met desktop (min. 768px) en mobiel (min. 320px).
2. De pagina moet binnen 2 seconden laden wanneer caching actief is; zonder caching mag de laadtijd niet langer zijn dan 4 seconden
3. Veilige opslag van voorkeuren in LocalStorage.
4. Intuïtieve navigatie (maximaal drie kliks naar elke functie).
5. De applicatie komt overeen met “WCAG 2.1 AA-toegankelijkheidsrichtlijnen” (b.v. contrast).
6. Ondersteunt offline gebruik met gecachte data.
7. Applicatie verbruikt maximaal 50 MB aan lokale opslag.
8. het maximum aantal API-calls bedraagt circa 50 per dag voor Ambee en 60 per minuut voor OpenWeather.
9. gebruikersinterface minstens 95% dekking of compatibiliteit biedt (bijvoorbeeld qua functies of ondersteunde gebruikersscenario's).
10. Foutpercentages bij API-opvragingen mogen niet hoger zijn dan 5
11. Gebruikerservaring is consistent over verschillende schermgroottes.
12. De applicatie ondersteunt tot 100 gelijktijdige gebruikers, zonder merkbaar verlies van prestaties.
13. Alle informatie is beschikbaar in het Nederlands en er wordt gebruikgemaakt van een duidelijk leesbaar lettertype van ten minste 16px.
14. Knoppen en interactieve elementen hebben een hover-effect voor betere zichtbaarheid.
15. De applicatie moet volledig bedienbaar zijn via het toetsenbord, waarbij gebruikers door alle interactieve onderdelen kunnen navigeren. Een voorbeeld hiervan is dat d.m.v.

knoppen kunnen bepaalde dingen zoals spatiebalk of de Enter plekjes geactiveerd worden.

16. Bij het delen van gegevens via e-mail of WhatsApp wordt gegevensversleuteling toegepast ten behoeve van de privacybescherming
 17. Applicatie biedt een helpsectie met veelgestelde vragen.
 18. Caching-mechanisme ververs data niet vaker dan elke 60 minuten.
 19. Applicatie toont een offline-melding bij internetuitval.
 20. De applicatie is vormgegeven met handmatig geschreven CSS, zonder gebruik te maken van bestaande frameworks. Dit benadrukt een directe en gecontroleerde benadering van het ontwerp.
 21. React-componenten zijn herbruikbaar en modulair ontworpen.
 22. Deze applicatie ondersteunt een donkere modus (er zijn twee verschillende modus) als toegankelijke optie .
 23. Foutmeldingen worden maximaal 5 seconden getoond voordat ze automatisch verdwijnen.
 24. Logbestanden worden voor een periode van maximaal 30 dagen opgeslagen om analyses mogelijk te maken.
 25. Applicatie ondersteunt een minimale schermresolutie van 320x480 pixels.
 26. Gebruikersinterface laadt binnen 1 seconde bij herhaalde bezoeken met caching.
 27. Applicatie biedt visuele feedback bij het klikken op knoppen (bijv. animatie).
 28. Bij het optreden van fouten worden foutmeldingen gepresenteerd in een modaal venster, waardoor de zichtbaarheid en gebruikerservaring worden verbeterd.
-

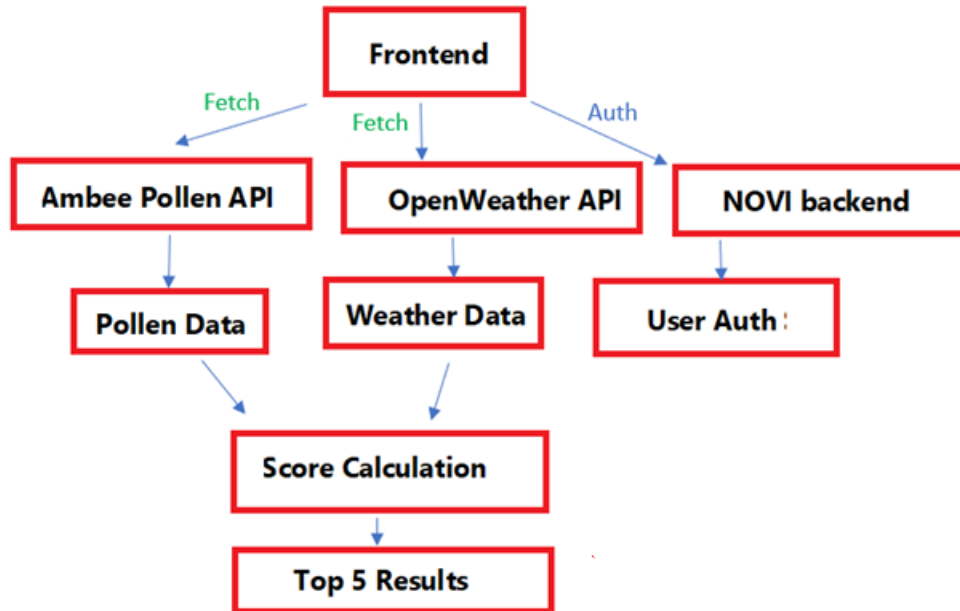
Technische Architectuur

- **De frontend** :Dit is opgebouwd met twee verschillende code methoden. Eerste is *“React”* en 2de is *“JavaScript”*. het bevat ook een *“handmatig geschreven CSS”*.
- **API Integratie**: De applicatie voert asynchrone *“fetch”*-verzoeken uit naar zowel de Ambee Pollen API als de OpenWeather API. Indien een verzoek mislukt, wordt automatisch retry-logica toegepast om de kans op succesvolle gegevensoverdracht te vergroten. Daarnaast wordt caching met een *“Time To Live (TTL)”* van één uur gehanteerd, zodat gegevens gedurende deze periode opnieuw gebruikt kunnen worden zonder dat herhaalde API-calls noodzakelijk zijn.
- **Voor de backend** :Er wordt gebruikgemaakt van de NOVI-backend voor gebruikersauthenticatie, met onder andere de endpoints POST *“/auth/register”* en POST *“/auth/login”*.
- **Dataopslag**: de bedoeling is dat via dat van LocalStorage gebruik maken voor voorkeuren. Daarnaast wordt hier van cache via localStorage gebruik gemaakt voor API-responsen.
- **Offline**: Indien de externe API's tijdelijk niet bereikbaar zijn, schakelt de applicatie automatisch over op gecachte gegevens of zogeheten *“mock-data”*. Dit waarborgt de

continuïteit van de functionaliteit, ook bij een tijdelijke onderbreking van de gegevensstroom.

- **Scoring:** Gewogen gemiddelde:
$$\text{Score} = (\text{Pollen} \times W_{\text{pollen}}) + (\text{Temperatuur} \times W_{\text{temp}}) + (\text{UV} \times W_{\text{uv}}) + (\text{Wind} \times W_{\text{wind}}) + (\text{Vocht} \times W_{\text{vocht}})$$

Waarbij W_x de gebruikersprioriteit is (1-5, genormaliseerd naar 0-1).



Handgetekende Wireframes

Wireframes zijn handgetekend en omvatten:

- **Login/Registratie:** Formulier met e-mail, wachtwoord, en knoppen.
- **Voorkeuren:** Sliders voor criteria (temp, pollen), prioriteitselectie.
- **Stedenoverzicht:** Het stedenoverzicht bestaat uit een lijst van tien steden, waarbij de vijf steden met de hoogste scores extra worden gemarkeerd.
- **Detailpagina:** Weer-/pollendata, drie adviezen, deeloptyes.
- **Inlevering:** Als geplakte afbeeldingen in de deze Pdf. file met beschrijvingen en bepaalde aanwijzingen.

Inloggen **Registreren**

E-mailadres: Registreren

voorbeeld@email.com

Gebruikersnaam:

Minimaal 3 tekens

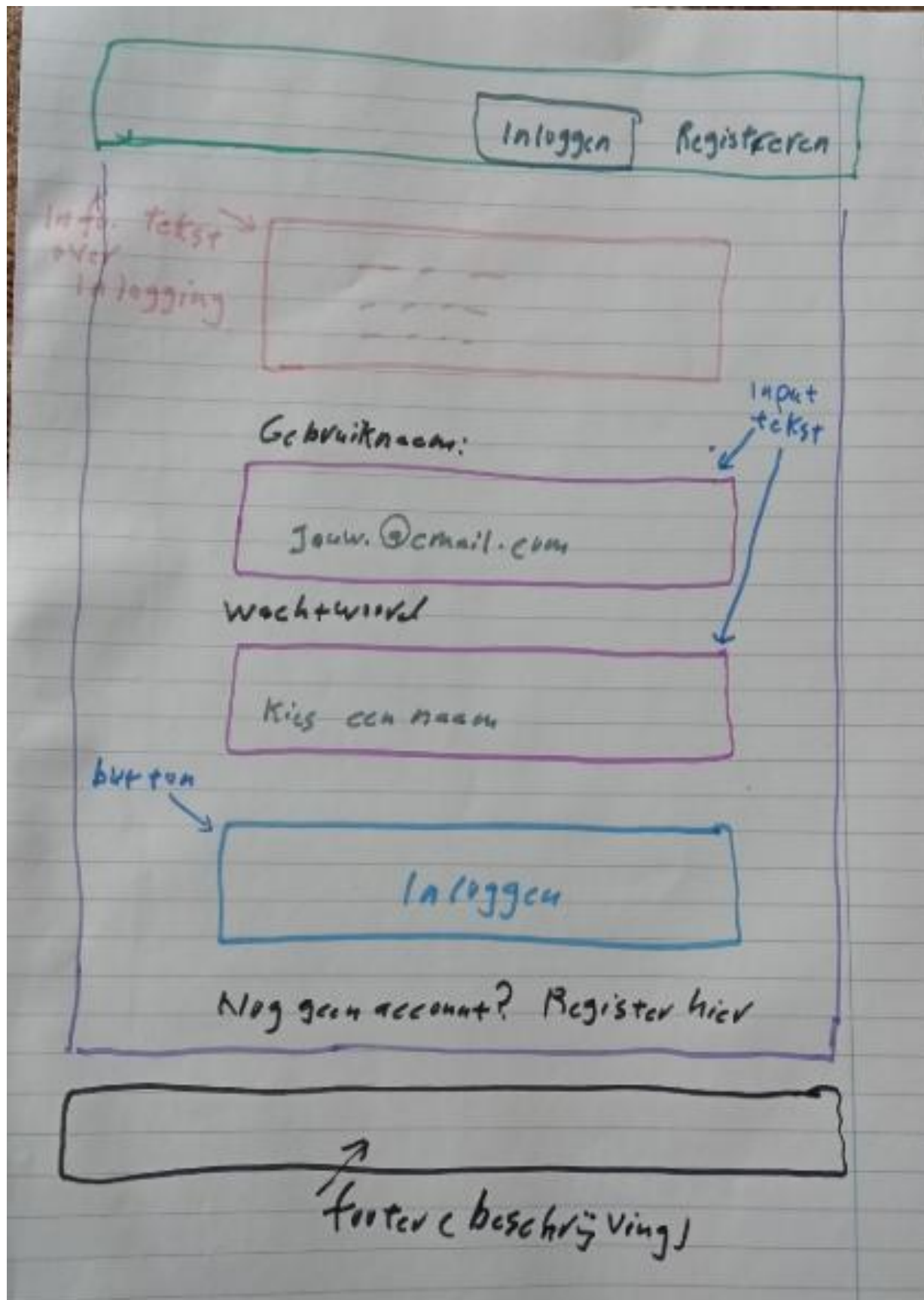
Wachtwoord:

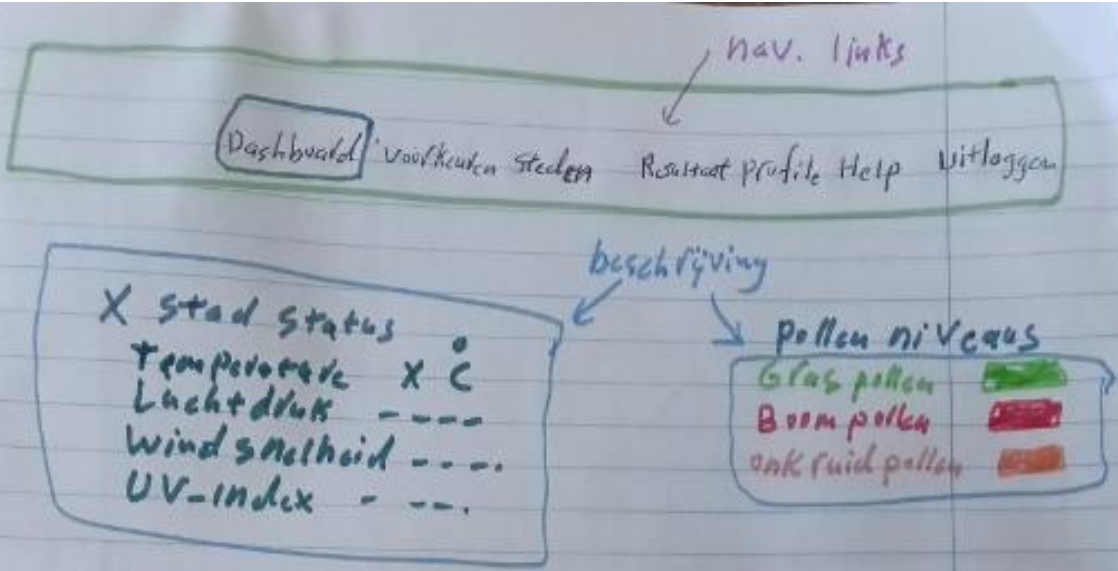
Minimaal 6 tekens

Registreren

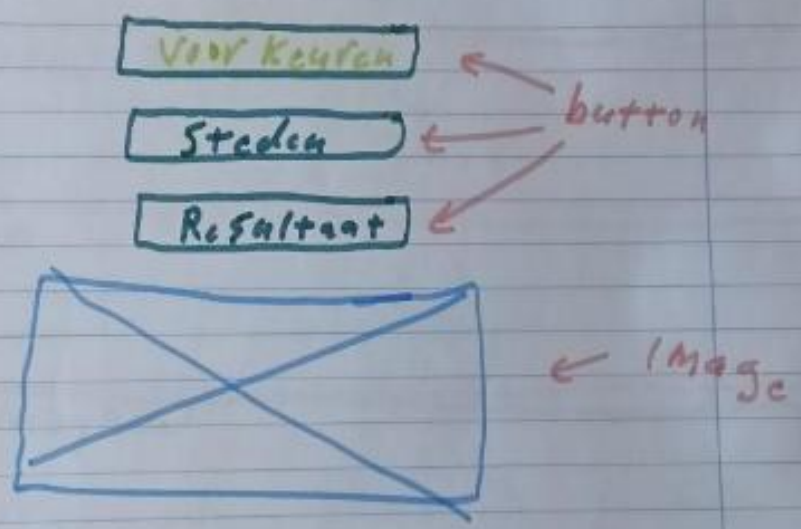
Heb je al een account? log hier in

footer beschrijving





Snelle actie



Dashbord Voorkeuren Steden Resultaat profiel Help Uitloggen

Stel je Voorkeuren in
temperatuur (°C)

box
↙

min ———— ● ————
max ———— ● ————

luchtvochtigheid (%)

min ———— ● ————
max ———— ● ————

Windsnelheid (km/u)

min ———— ● ————
max ———— ● ————

UV-index

min ———— ● ————
max ———— ● ————

pollengevoeligheid

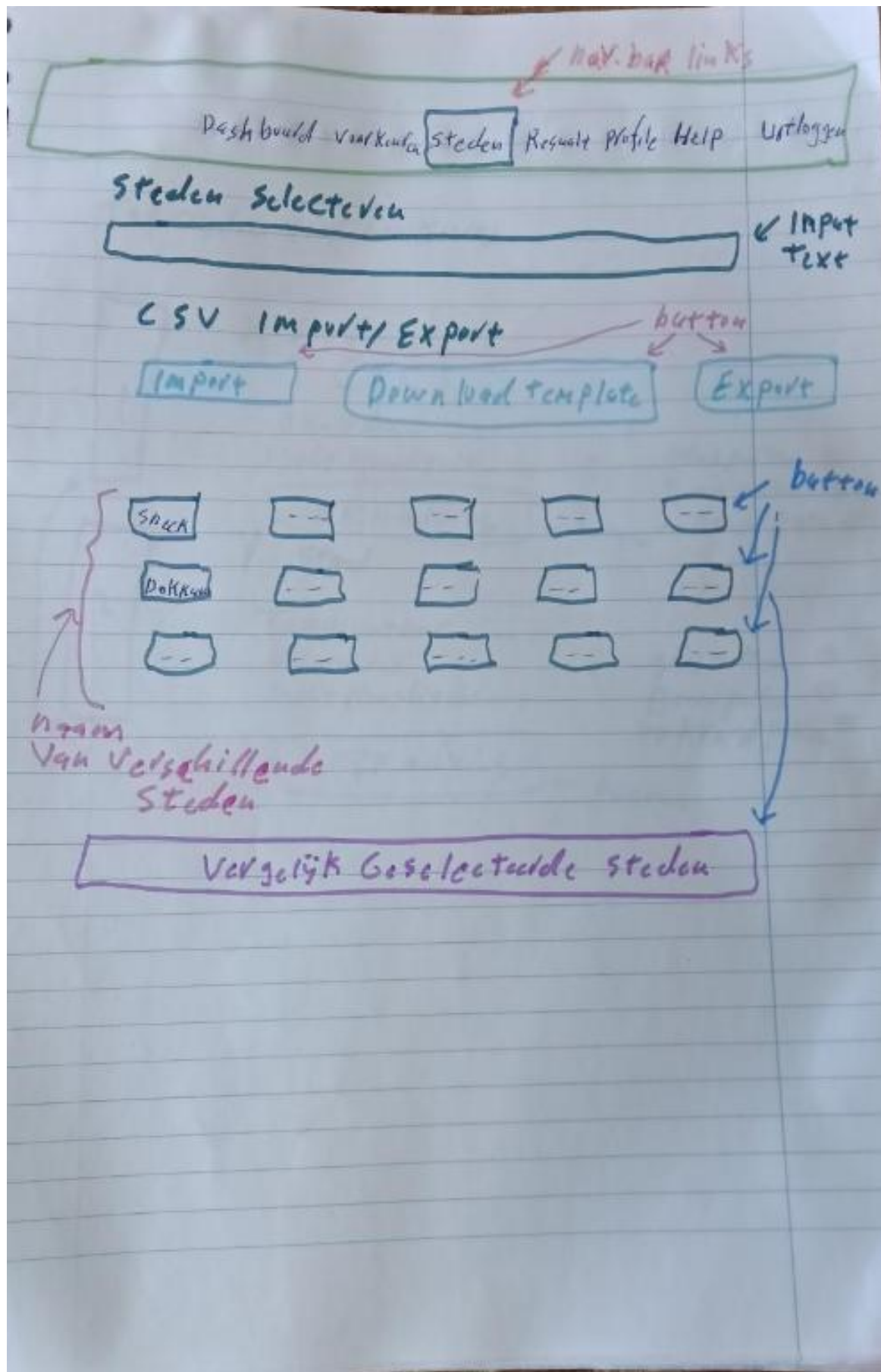
Gras pollen	1	2	3	4	5
Bes pollen	1	2	3	4	5
onkruid pollen	1	2	3	4	5

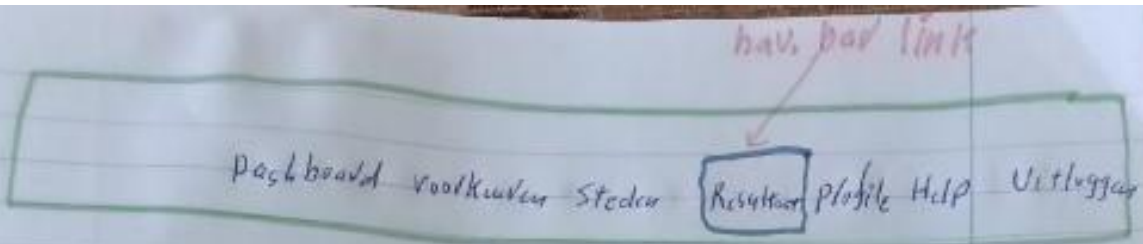
Reset naar standaard

↳ button

opslaan en doorgaan

↙





Top steden voor huiltroorts

← beschrijving



X stad
 $T(X^0)$

UV-index ---

lucht kwaliteit ---

bekijk advies

Score ---

gros pollen ●
 Boom pollen ●
 onkruid pollen ●

Y stad

Temperatuur ---

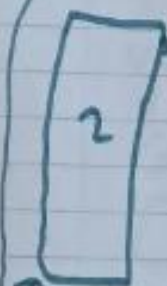
UV-index ---

lucht kwaliteit ---

bekijk advies

Score ---

gros pollen ●
 Boom pollen ●
 onkruid pollen ●



box

← button

hov. hov link

Dashboard Voorkeuren Steden Resectest profiel **Help** Uitloggen

Help en Veel gestelde vragen

Algemeen

pollen

Voorkeuren

Klachten

button

Wat is de Hovikorts App?

Hoe werkt de App?

beschrijving

Nog vragen?

stuur e-mail

button

Schermontwerpen

- Ontwikkeld in Figma met een groen- grijs palet voor toegankelijkheid.
 - Responsieve lay-outs: mobiel (320px), tablet (768px), desktop (1920px).
 - Interactieve elementen: knoppen, sliders, zoekfunctie.
 - Binnen het document is een Figma-link toegevoegd, waarmee illustraties van het ontwerp inzichtelijk worden gemaakt.⁶
-

Broncode React Applicatie (Projectstructuur)

De broncode van de Hooikoorts Stadwijzer is opgebouwd als een modulaair React-project. De structuur is logisch ingedeeld met duidelijke mappen voor componenten, context, services, assets en hulpfuncties. Alle logica is toegepast in “*JavaScript (ES6)*” in combinatie met “*React 18+*”. De projectstructuur bevat een “*README.md-bestand*” met duidelijke installatie-instructies, en de broncode is publiekelijk beschikbaar via GitHub. Hieronder volgt een overzicht van de projectstructuur.

public/

Bevat het algemene HTML-bestand (index.html) waarmee de app wordt opgestart.

src/

Hier bevindt zich de volledige broncode van mijn API's de applicatie:

assets/

– Afbeeldingen en andere statische bestanden.

components/

– Herbruikbare React-componenten, (b.v. knoppen) zijn altijd beschikbaar om herhaaldelijk gebruik worden binnen verschillende plekken van de API's.

context/

– Context-bestanden, die is bedoelt voor state management, bijvoorbeeld gebruikersvoorkeuren of thema's.

services/

- Daarnaast bevat het systeem functies die geschikt zijn voor het uitvoeren van applicatie-aanroepen, bijvoorbeeld voor diensten van Ambee.

utils/

- Hulpfuncties zoals berekeningen (bijv. “*scoreCalculator*”).

App.jsx/

- De hoofdsamenstelling van de interface.

main.jsx/

- Dit deel fungeert als het startpunt van de applicatie, waar “*React*” wordt geïnitieerd..

.env.example/

Voorbeeldbestand voor de benodigde omgevingsvariabelen zoals API-sleutels.

package.json/

Bevat projectgegevens en een overzicht van gebruikte dependencies en scripts.

README.md/

Document met installatie-instructies, uitleg en projectinformatie.

vite.config.js/

Configuratiebestand voor de ontwikkelserver en buildproces via Vite.

Link Naar Figma Ontwerpen

<https://www.figma.com/design/I4KMhOfEwsVucf4HR50NXI/Untitled?node-id=0-1&p=f&t=mqjblE6Xte2HfI2-0>

Conclusie

De Hooikoorts Stadwijzer Friesland is in essentie een toegankelijke webapplicatie die zich richt op het ondersteunen van mensen met hooikoorts bij het selecteren van de meest geschikte Friese steden. Hierbij wordt gebruikgemaakt van actuele pollendata en weersinformatie, waarmee gebruikers hun keuzes kunnen baseren op feitelijke en persoonlijke voorkeuren. De applicatie hanteert een geavanceerd scoringsmechanisme dat realtime data verwerkt, afkomstig van zowel de Ambee Pollen API als de OpenWeather API. Hierdoor biedt de applicatie gebruikers accurate en relevante aanbevelingen. Het voldoet volledig aan de eisen van de eindopdracht Front-end en biedt een praktische, toegankelijke oplossing voor hooikoortspatiënten. Door de combinatie van technische robuustheid, intuïtief design en innovatieve data-integratie onderscheidt de applicatie zich als een waardevol hulpmiddel. Met verdere uitbreiding van de backend en schaalbare API-toegang is de applicatie geschikt te maken voor bredere uitrol, ook buiten Friesland.