

Functioneel Ontwerp – SportData App

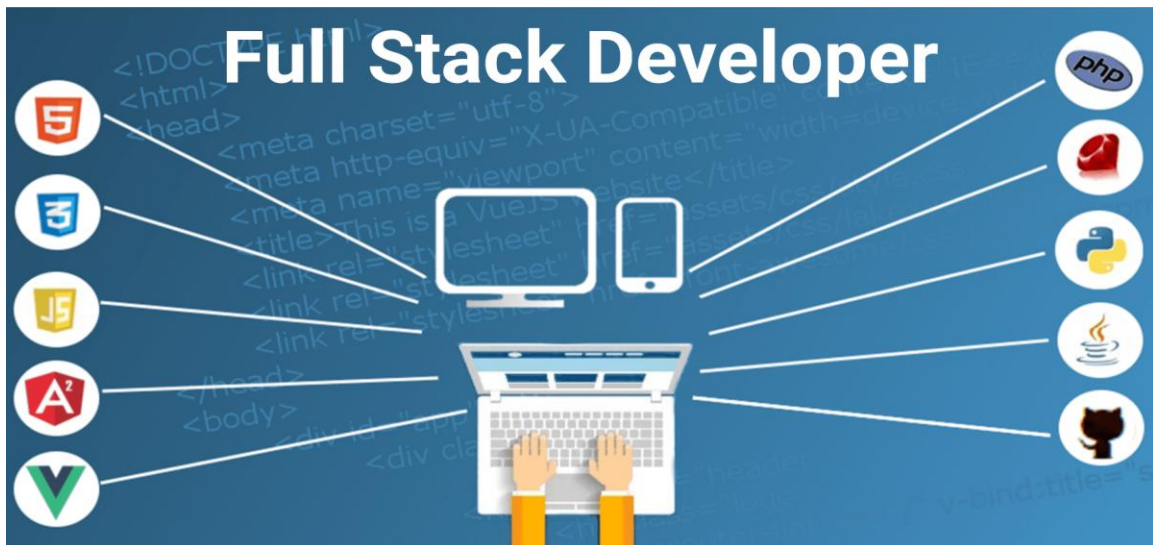
Project: SportData App

Opdracht: Eindopdracht Frontend

Opleiding: Fullstack developer Bootcamp – NOVI Hogeschool

Student: Thomas Oudhoff

Datum: 19-04-2025



Inhoudsopgave

1. Inleiding
2. Ideefase
3. Probleembeschrijving & Oplossing
4. Use Case Tabellen
5. Functionele & Niet-Functionele Eisen
6. Wireframes
7. Schermontwerpen
8. Inspiratiebronnen

1. Inleiding

De SportData App is een frontend applicatie waarmee gebruikers hun favoriete voetbalteams kunnen volgen. De app maakt gebruik van de openbare API van TheSportsDB en biedt functionaliteiten zoals het bekijken van standen, teamdetails, aankomende wedstrijden en het beheren van favoriete teams.

Gebruikers kunnen inloggen of een account aanmaken, waarna zij via een gebruiksvriendelijke interface teams kunnen opzoeken, details bekijken en teams markeren als favoriet. De app is ontwikkeld met React, en maakt gebruik van React Router, CSS modules en JWT-authenticatie.

De API van TheSportsDB is gedeeltelijk gratis beschikbaar. Sommige uitgebreide functies, zoals volledige toegang tot alle teamgegevens of recente wedstrijden, vereisen echter een betaald abonnement. Hierdoor kan het voorkomen dat bij het opvragen van gegevens van een specifiek team altijd dezelfde voorbeelddata wordt teruggegeven, zoals bijvoorbeeld teaminformatie van Arsenal. Ondanks dat de juiste team-ID wordt gebruikt in de API-aanroep, retourneert de gratis versie van de API enkel voorbeeldresultaten. Dit betekent dat de gegevens technisch correct worden opgevraagd, maar de weergave beperkt is tot beschikbare gratis data.

The github link van deze pagina:

[ThomasOudhoff/SportDataApp: Frontend opdracht Novi Hogeschool HBO Software Development](#)

2. Ideefase

Sportdata app

Probleem

Met deze applicatie wil ik het makkelijker maken voor sportliefhebbers om op één plek alles te vinden over hun favoriete teams en hun wedstrijden. Vaak moeten mensen naar meerdere websites om informatie over sportevenementen en teamdetails te vinden. Deze app brengt al die gegevens bij elkaar, zodat gebruikers niet meer hoeven te zoeken.

API

Voor de informatie gebruiken we de TheSportsDB API, een bron die sportdata aanbiedt, zoals gegevens over competities, teams, en spelers.

Vier belangrijkste functies

- ***Inloggen en registreren:***
Gebruikers kunnen een account aanmaken om toegang te krijgen tot de app en hun persoonlijke instellingen op te slaan.
- ***Favoriete teams***
Gebruikers kunnen teams markeren als favorieten, zodat ze sneller updates en informatie over hun favoriete team kunnen vinden.
- ***Wedstrijdschema's en scores:***
Overzicht van aankomende wedstrijden en uitslagen van eerder gespeelde wedstrijden.
- ***Statistieken en details:***
Informatie over de prestaties van teams.

3. Probleembeschrijving & Oplossing

Probleem: Veel sportliefhebbers willen altijd en overal op een centrale plek op de hoogte blijven van de prestaties van hun favoriete voetbalteams. De meeste bestaande apps zijn complex, bieden teveel informatie of vragen om betaalde abonnementen.

Oplossing: De SportData App biedt een toegankelijke en overzichtelijke manier om alleen de belangrijkste informatie over favoriete teams te volgen:

- Standen
- Komende wedstrijden
- Teamdetails met logo's, stadioninformatie en social media
- Favorieten teams opslaan

4. Use Case Tabellen

Use Case 1	
Titel	Registreren/Inloggen
Actor	Nieuwe gebruiker / Terugkerende gebruiker
Trigger	De gebruiker kiest ervoor om de applicatie te gebruiken waarvoor een account vereist is.
Preconditie 1	Gebruiker is op de inlog- of registratiepagina.
Preconditie 2	De gebruiker heeft een internetverbinding.
Main succes scenario	<ol style="list-style-type: none">1. Gebruiker kiest voor registreren of inloggen.2. Gebruiker voert de benodigde gegevens in.3. Het systeem valideert de ingevoerde gegevens.4. Gebruiker wordt succesvol ingelogd en krijgt toegang tot de applicatie.
Postconditie	De gebruiker is ingelogd en krijgt toegang tot alle functies van de applicatie.

Use Case 2	
Titel	Teamoverzicht bekijken
Actor	Gebruiker
Trigger	De gebruiker wil een overzicht zien van alle teams in een competitie.
Preconditie 1	De gebruiker heeft een competitie geselecteerd.
Preconditie 2	De gebruiker heeft een internetverbinding.
Main succes scenario	<ol style="list-style-type: none">1. Gebruiker navigeert naar de teamoverzichtspagina.2. Het systeem haalt alle teams op via de API.3. Teams worden weergegeven met naam en logo.4. Gebruiker kan doorklikken naar details van elk team.
Postconditie	De gebruiker ziet alle teams van de geselecteerde competitie.

Use Case 3	
Titel	Teamdetails bekijken
Actor	Gebruiker
Trigger	De gebruiker wil informatie zien van een specifiek team.
Preconditie 1	De gebruiker bevindt zich op de teamoverzichtspagina.
Preconditie 2	De gebruiker heeft op een team geklikt.
Main succes scenario	1. Het systeem haalt teamdetails op aan de hand van team-ID. 2. Logo, teamnaam en stadion, capaciteit 3. Gebruiker ziet aankomende en afgelopen wedstrijden.
Postconditie	De gebruiker ziet alle informatie van het gekozen team.

Use Case 4	
Titel	Team toevoegen aan favorieten
Actor	Ingelogde gebruiker
Trigger	Gebruiker wil een team markeren als favoriet.
Preconditie 1	Gebruiker is ingelogd.
Preconditie 2	Gebruiker bevindt zich op de teamoverzichtspagina.
Main succes scenario	1. Gebruiker klikt op ster-icoon bij een team. 2. Het team wordt toegevoegd aan een database 3. De ster verandert van kleur om favoriet aan te duiden.
Postconditie	Het team is toegevoegd aan de favorietenlijst van de gebruiker.

5. Functionele & Niet-Functionele Eisen

5.1 Functionele eisen

1. De gebruiker moet kunnen registreren via een formulier.
2. De gebruiker moet kunnen inloggen met gebruikersnaam en wachtwoord.
3. Na succesvolle login ontvangt de gebruiker een JWT-token.
4. De applicatie moet controleren of het JWT-token geldig is.
5. De gebruiker moet uit kunnen loggen.
6. Alleen ingelogde gebruikers kunnen de pagina met favoriete teams bekijken.
7. De gebruiker kan competities zoeken via een zoekbalk.
8. De gebruiker kan een competitie selecteren uit een lijst.
9. De geselecteerde competitie toont een standenoverzicht.
10. De gebruiker kan teams bekijken van een competitie.
11. Teams worden weergegeven met naam en logo.
12. De gebruiker kan klikken op een team voor meer informatie.
13. De detailpagina van een team toont stadion en capaciteit.
14. De detailpagina toont de social media links van een team.
15. De gebruiker kan een team markeren als favoriet.
16. Favoriete teams worden opgeslagen in localStorage.
17. De gebruiker kan favoriete teams verwijderen met bevestiging.
18. De pagina met favoriete teams toont de laatste 5 resultaten per team.
19. Resultaten worden weergegeven als W/V/G/N.
20. De achtergrondkleur van resultaatblokken is afhankelijk van de uitkomst.
21. De gebruiker kan navigeren tussen 'stand' en 'aankomende wedstrijden'.
22. Aankomende wedstrijden worden opgehaald via de API.

23. De gebruiker ziet wedstrijden met teams, score, stadion en tijd.
24. De gebruiker ziet competitie-informatie boven de wedstrijdkaart.
25. De gebruiker ziet steeds de actuele datum en tijd in de navbar.
26. De applicatie onthoudt de laatst geselecteerde competitie.
27. Teams worden dynamisch geladen op basis van league-ID.
28. Bij klikken op een team wordt de juiste team-ID opgevraagd via de API.
29. De applicatie toont een melding als geen data beschikbaar is.
30. Alle pagina's gebruiken een consistente layout met scrollbare vlakken.

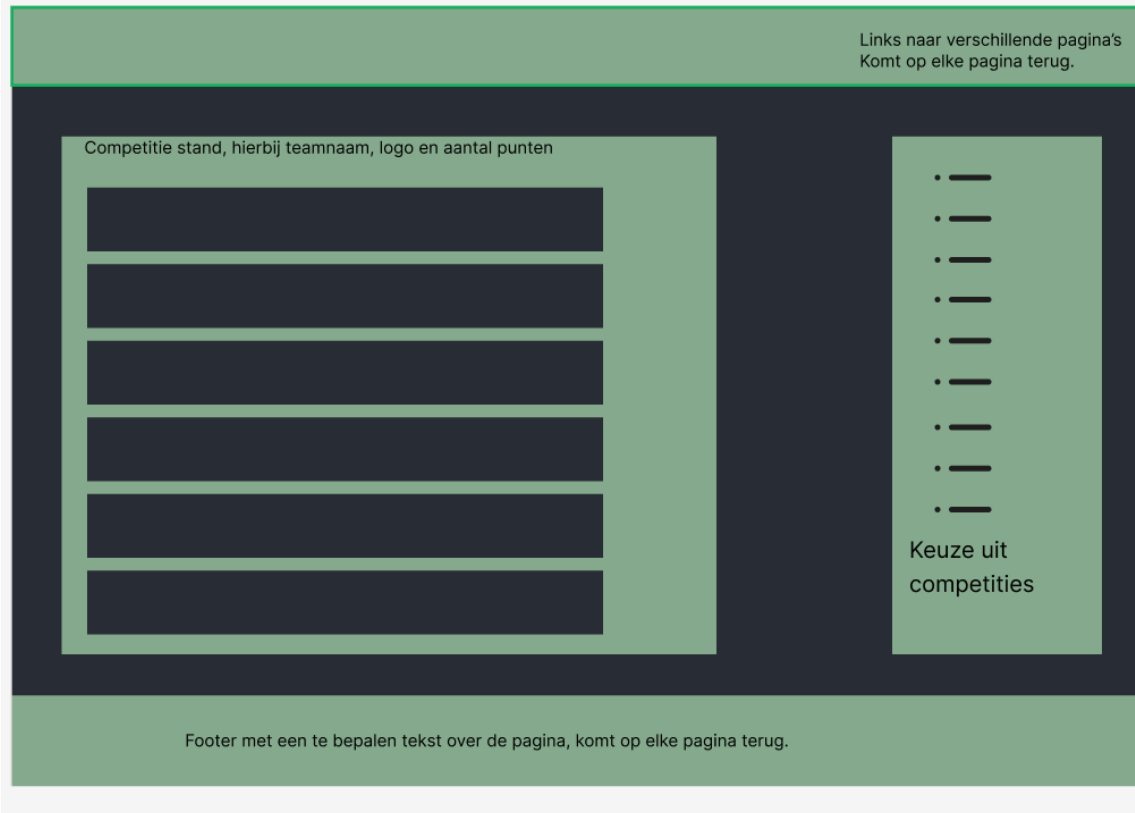
5.2 Niet-functionele eisen

1. De applicatie moet binnen 2 seconden opstarten in de browser.
2. De gebruikersinterface moet responsive zijn op mobiel en desktop.
3. De gebruikte API moet publiek toegankelijk zijn (TheSportsDB).
4. Gegevensuitwisseling gebeurt via HTTPS.
5. Er mag geen gevoelige gebruikersdata in de frontend zichtbaar zijn.
6. Gebruikerswachtwoorden worden niet opgeslagen in de frontend.
7. De applicatie maakt gebruik van JWT voor beveiligde routes.
8. De favorieten worden lokaal opgeslagen per gebruiker.
9. De gebruiker ontvangt duidelijke foutmeldingen bij mislukte login.
10. De code is gestructureerd in componenten per functionaliteit.
11. CSS wordt modulaair opgebouwd met duidelijke klassenamen.
12. De applicatie gebruikt React Router voor navigatie.
13. Data wordt asynchroon opgehaald via fetch.
14. Er wordt gebruikgemaakt van useState en useEffect hooks.
15. De applicatie is getest met minimaal 2 testaccounts.
16. De broncode is vrij van console errors bij oplevering.
17. De applicatie bevat een README met installatie-instructies.
18. API-sleutels zijn afgeschermd via een .env bestand.
19. Het project is versiebeheer-vriendelijk opgezet via Git.
20. De app is gebruiksvriendelijk en intuïtief te gebruiken zonder uitleg.

6. Wireframes

[SportsData App – Figma](#)

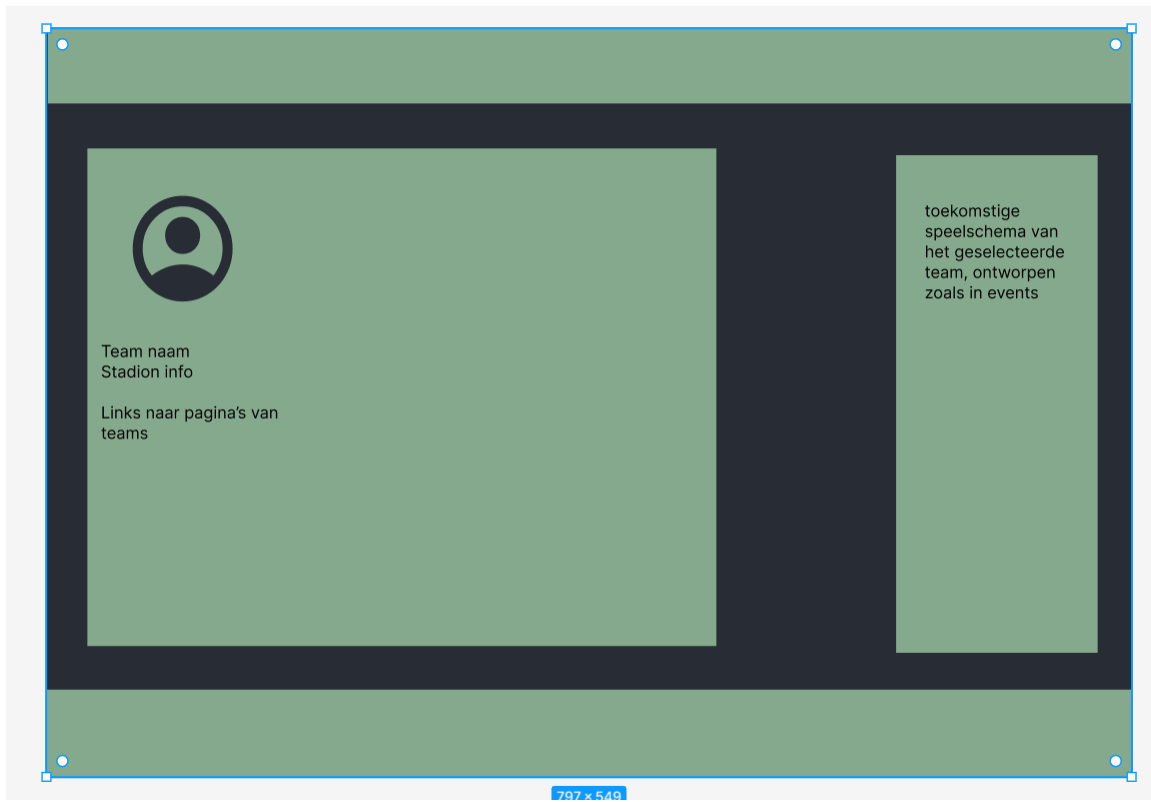
1. Thuis pagina



2. Favorieten pagina



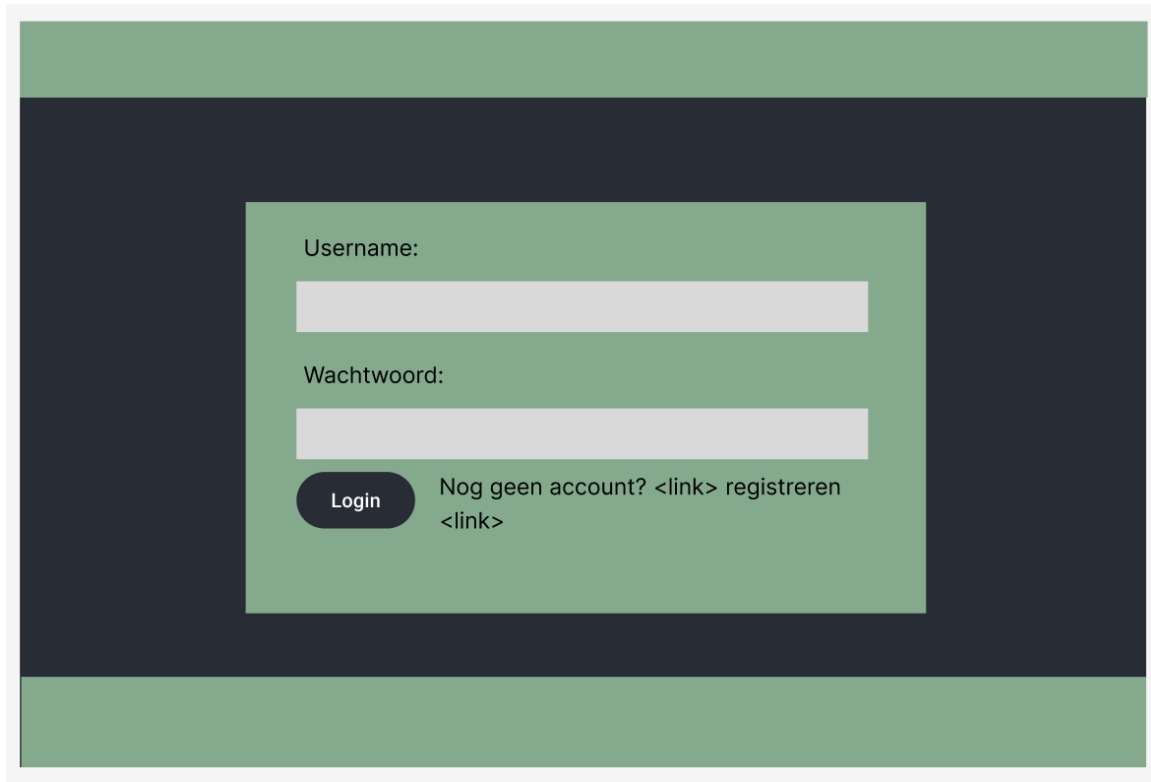
3. Teamdetails



4. Teamslist

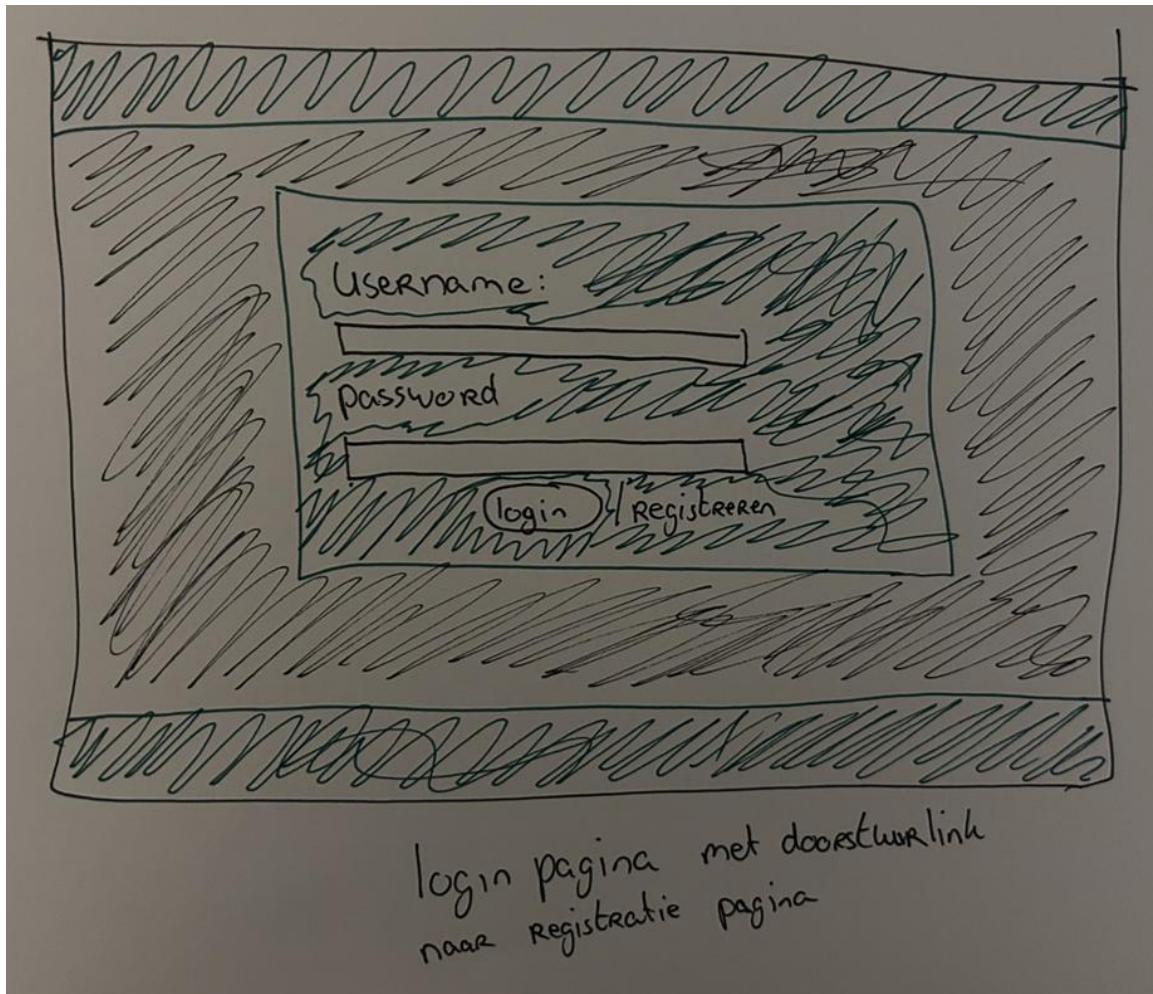


5. Login

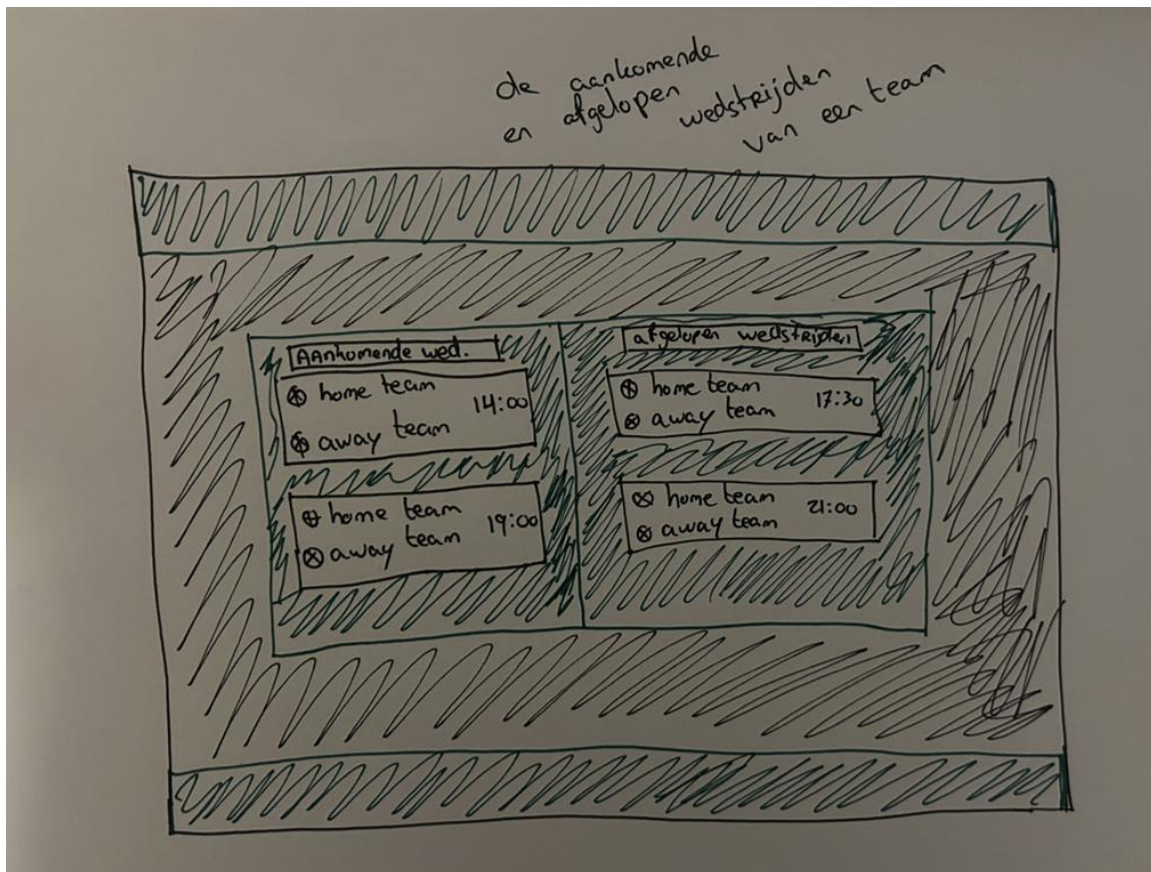


7. Schermontwerpen

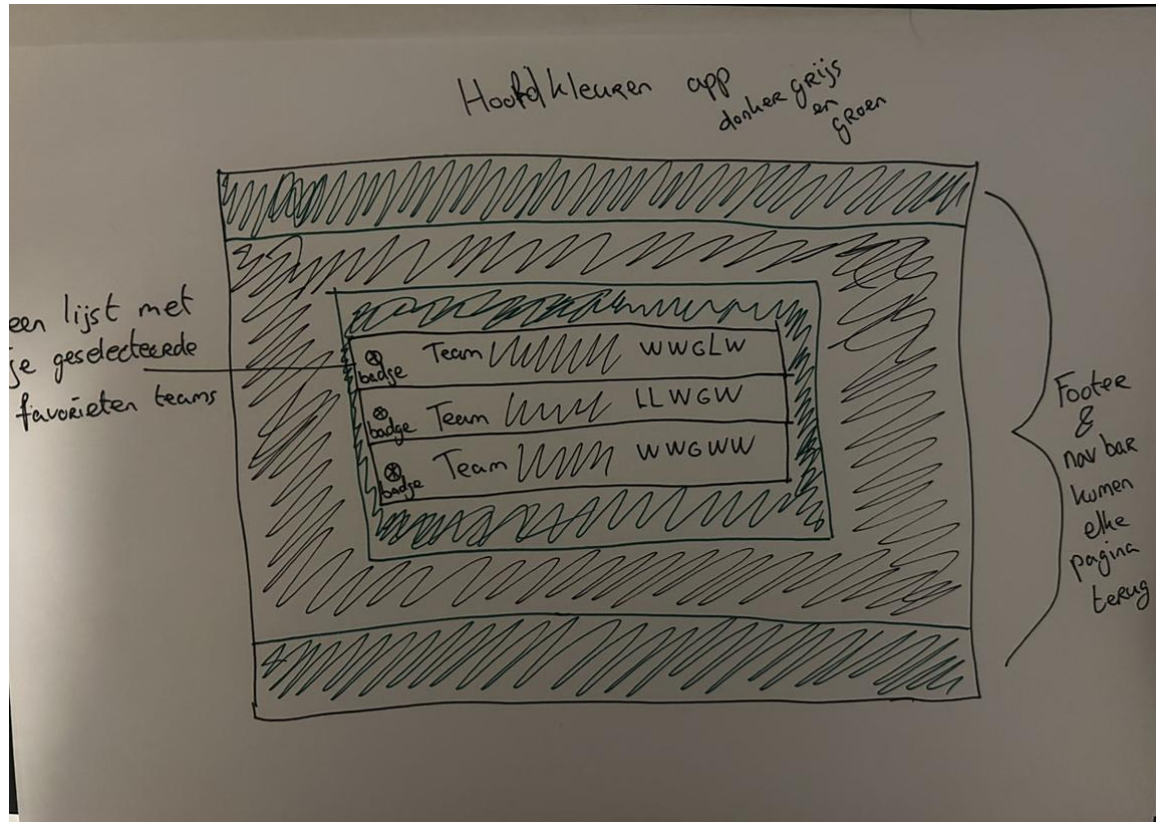
Login pagina



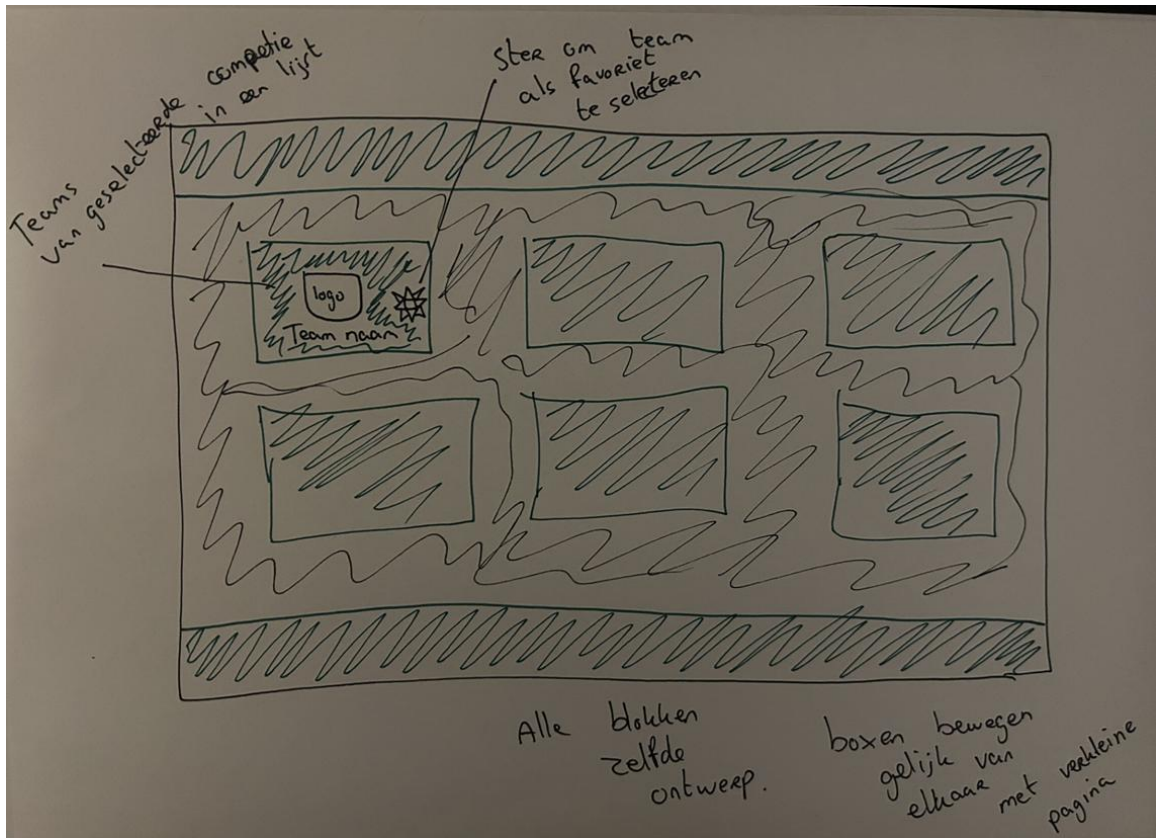
Event pagina



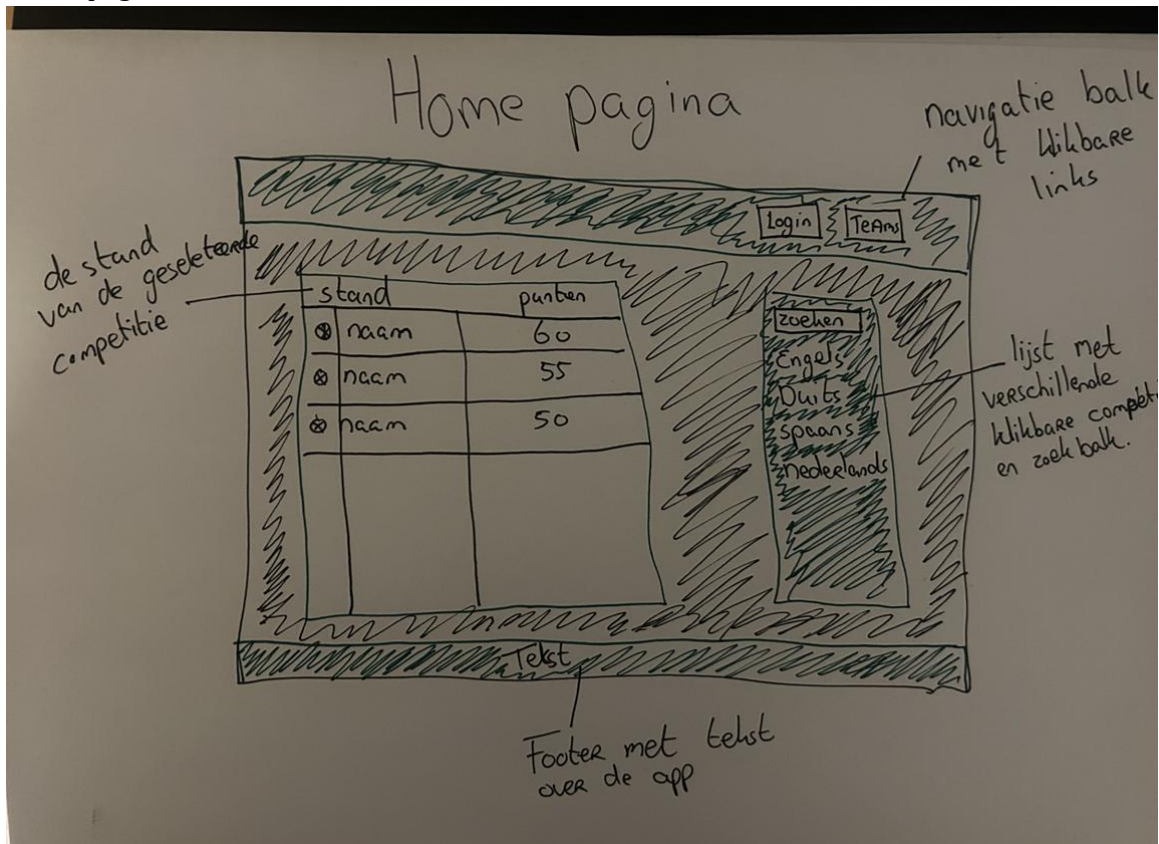
Favorieten team pagina



Teams pagina



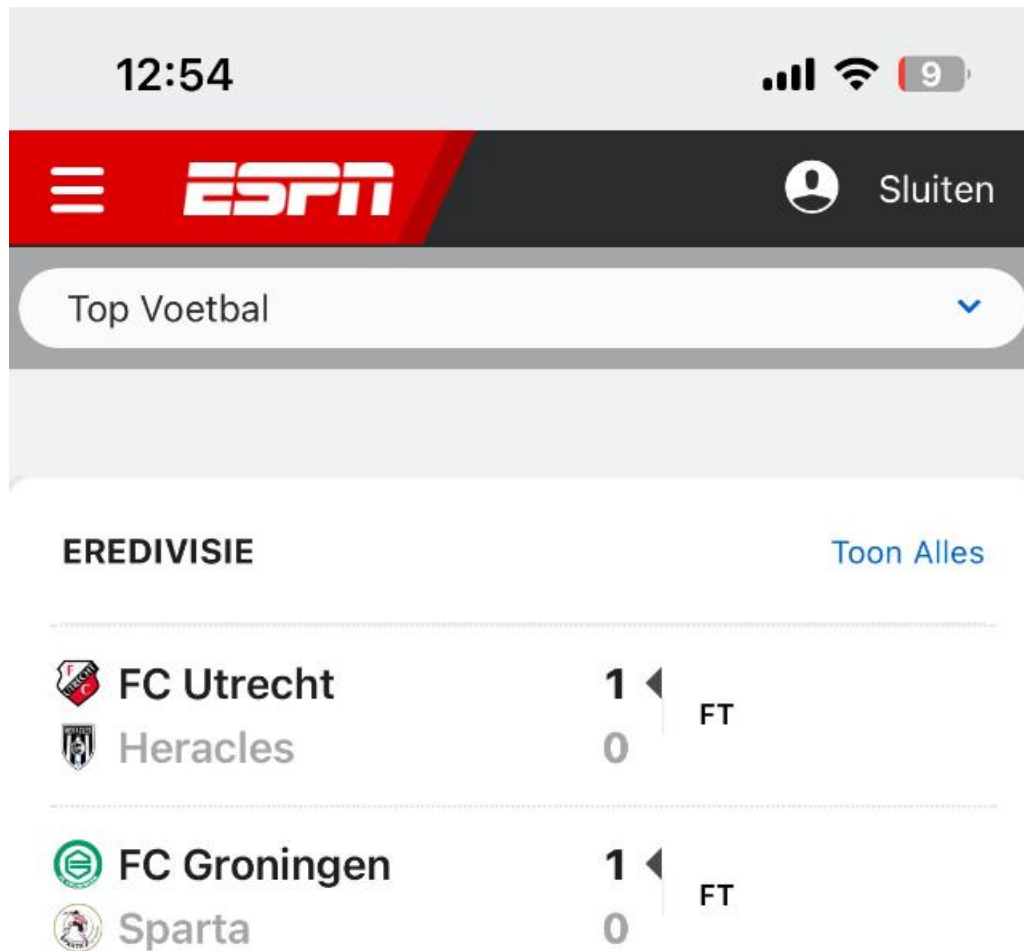
Home pagina



6. Inspiratiebronnen voor Visuele Stijl

1. ESPN App:

De overzichtelijke indeling en visuele accenten geven snel inzicht in scores dit onderwerp gebruik ik om de informatie op de app eenvoudig en overzichtelijk te maken.



2. SofaScore:
Strakke lijnen, een team is te herkennen aan badge en naam.



3. FlashScore:

Dit is de inspiratiebron voor de log in pagina, waarin je tijdens het inloggen al de styling van mijn applicatie kunt zien.



