

# LOGBOEK

Kaori Lauwers - 1GDM1

Prototyping Tools

## Welkom!

In dit logboek toon ik mijn groeiproces binnen het vak **Prototyping Tools**. Doorheen de lessen heb ik gewerkt met low-fi en mid-fi prototypes, UX- en UI-beslissingen, HTML & CSS, Tailwind en het omzetten van ontwerpen naar functionele interfaces.

Elke les bouwt verder op de vorige: van schetsen en structuur, naar visuele hiërarchie, interactie en uiteindelijk een afgewerkt digitaal product. Hieronder vind je een overzicht van alle lessen en mijn evolutie doorheen dit traject.

[Bekijk mijn volledig proces in PDF →](#)

### Les 1

Low-fi Prototypes

### Les 2

Mid-fi Prototypes

### Les 3

Typografie en opmaak

### Les 4

HTML & CSS verdieping

### Les 5

GitHub & Netlify

### Les 6

Werken met Tailwind

### Les 7

Treinschermen → GSM

### Les 8

Tailwind Verder Aanpassen

### Les 9

Tailwind Probleem & Oplossing

## Les 1: Low-fi Prototypes

### Wijzigingen & Redenen

- Klok toegevoegd aan perronscherm, extra taal toegevoegd, pictogrammen toegevoegd, volgorde info aangepast.
- Waarom? Klok helpt tijdsdruk te visualiseren, meertaligheid duidelijker, consistent met NMBS-stijl.

### UX/UI-vragen

- Hoe blijft meertalige info (NL/FR/DE) overzichtelijk?
- Welke info moet bovenaan staan zodat reizigers het snel herkennen?
- Hoeveel tekst kan op één scherm zonder rommelig te worden?
- Welke rol spelen iconen vs. tekst?
- Is een klok of vertraging sneller te begrijpen dan tekst?

### Usability checks

- Test (met zus):
  - Vraag: "Kun je in 5 seconden zien waar de trein naar Turnhout vertrekt?"
  - Resultaat: "Ja, maar ik keek eerst naar de klok."
- Observaties:
  - Gebruikers kijken eerst naar tijd & spoor.
  - Te veel talen maakt het druk → misschien 1 hoofdt taal.

### Reflectie & Next steps

- Wat werkte goed? Duidelijke hiërarchie.
- Wat minder goed? Drie talen tegelijk → druk, iconen te klein.

## Les 2: Mid-fi Prototypes

### Wijzigingen & redenen

- Wat heb ik aangepast?
  - Ontwerp herwerkt naar mid-fidelity
  - Meer bestemmingen toegevoegd
  - Kleurenpalet verfijnd
  - Verschillende tekstdiktes voor hiërarchie
  - Nieuwe iconen toegevoegd
  - Lettertype aangepast voor leesbaarheid
  - Scheidingslijnen toegevoegd
  - Volgorde info herwerkt (tijd → bestemming → spoor)
- Waarom?
  - Betere visuele hiërarchie
  - Duidelijk contrast op drukke perrons
  - Meer leescomfort
  - Structuur door scheidingslijnen
  - Snellere interpretatie door iconen

### UX/UI-vragen

- Hoe toon ik veel info maar behoud ik rust?
- Hoe hou ik meertalige info overzichtelijk?
- Welke info krijgt prioriteit?
- Hoe blijven iconen universeel?
- Hoe zorg ik voor samenhang tussen schermtypes?

### Usability checks

- Test met 2 gebruikers:
  - “Kun je in 5 seconden de wagon met fietsplekken vinden?” → Ja
  - “Wat zie je eerst?” → Tijd & bestemming
- Observaties:
  - Gebruikers kijken eerst naar tijd en spoor
  - Iconen worden sneller begrepen dan tekst

### Reflectie & next steps

- Wat werkte goed: duidelijke structuur
- Wat minder werkte: meerdere talen tegelijk tonen



## Les 3: Typografie en opmaak in HTML & CSS

### Oefening: typografie op schermen

- Experimenteren met kleur, typografie en witruimte via BPUIEditor .
- Aanpassen van titels, paragrafen en lijsten.
- Testen van lettertypes, groottes, spatiëring en kleuren.
- Observeren van leesbaarheid en visuele hiërarchie.
- Screenshots maken van het eindresultaat.

### Export en opmaak in HTML & CSS

- Ontwerp gemaakt voor het scherm van de centrale stationshal.
- Export naar HTML en CSS met behulp van ChatGPT.
- Prompt gebruikt zoals:
  - “*Create 3 HTML files based on these screens that include CSS and the import of the fonts. The font is Roboto Slab.*”
- Bestanden opgeslagen met een hoofdpagina `index.html`.
- Voorbereiding voor publicatie op GitHub en Netlify.

## Les 4: HTML & CSS – Typografie, Kleur en Ruimte

### Inhoud van de les

- Introductie tot HTML en CSS.
- Typografie: lettertypes, groottes en hiërarchie.
- Kleurgebruik en contrast voor leesbaarheid.
- Ruimte, marges en witruimte.
- Praktische oefening met HTML-pagina's.

### Observaties tijdens de les

- Consistente typografie zorgt voor duidelijkheid.
- Goed contrast is essentieel voor leesbaarheid.
- Witruimte brengt rust in een layout.
- Oefeningen maken de theorie begrijpelijker.

### Besproken tools & bronnen

- Google Fonts voor typografie.
- Visual Studio Code voor HTML en CSS.
- Online editors om snel te testen.

## Les 5: GitHub & Netlify

### Wat we hebben gedaan

In deze les leerden we hoe we onze website online konden zetten. We maakten een GitHub repository aan en koppelden deze aan Netlify, zodat de website automatisch gedeployed werd.

### Belangrijke stappen

- Een nieuwe GitHub repository aangemaakt.
- Projectbestanden gecommit en gepusht.
- Netlify gekoppeld aan GitHub.
- Website automatisch online gezet.

### Wat ik onthoud

- Een repository is de online versie van je project.
- Elke update op GitHub verschijnt automatisch op Netlify.
- De juiste mappenstructuur is belangrijk.

## Les 6: Werken met Tailwind

### Wat we hebben geleerd

In deze les leerden we hoe Tailwind werkt en hoe we onze bestaande CSS konden vervangen door utility classes. We werkten verder aan onze logboekwebsite in Visual Studio Code.

### Wat we hebben gedaan

- Tailwind toegevoegd aan het project.
- Bestaande CSS omgezet naar Tailwind-classes.
- Layout en spacing verbeterd met utility classes.
- De logboekpagina's verder afgewerkt.

### Wat ik onthoud

- Tailwind werkt volledig via classes in HTML.
- Styling aanpassen gaat snel en overzichtelijk.
- Minder aparte CSS-bestanden nodig.

## Les 7: Treinschermen → GSM Schermen

### Wat ik heb gedaan

- Mijn treinschermen omgezet naar een mobiele gsm-versie.
- De layout aangepast voor leesbaarheid op een klein scherm.
- Elementen herplaatst zoals tijd, iconen en tekstblokken.
- Informatie compacter gemaakt zonder belangrijke info te verliezen.

### Waarom deze aanpassingen

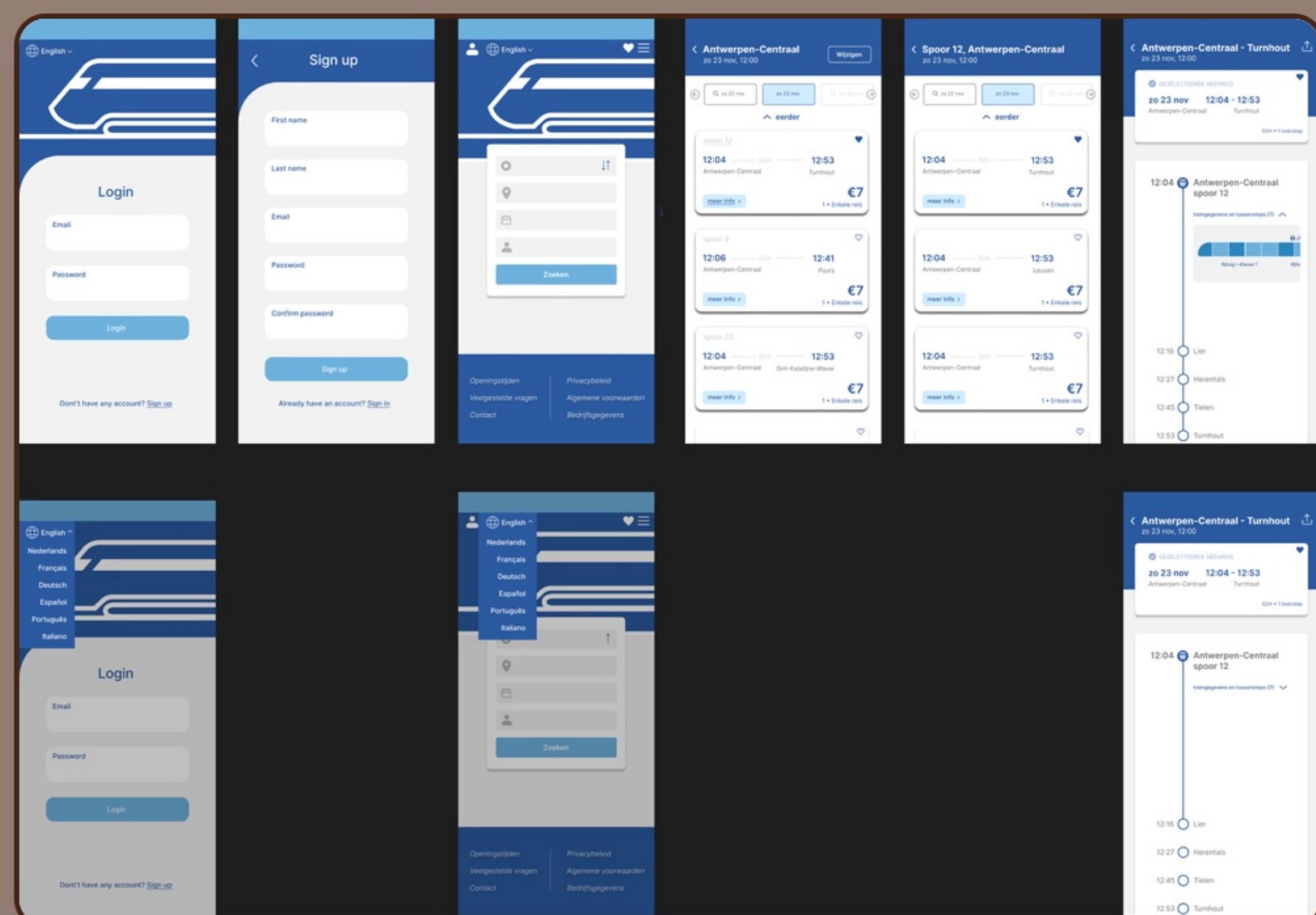
- Op een gsm moet informatie snel scannbaar zijn.
- Er is minder ruimte dan op grote treinschermen.
- Iconen moesten groter en eenvoudiger blijven.
- Een gsm wordt vaak gebruikt terwijl mensen onderweg zijn.

### Mijn proces

- Eerst bepaald welke info echt nodig is voor reizigers.
- De content opgesplitst in verticale blokken.
- Verschillende groottes getest voor tekst en iconen.
- De stijl afgestemd op mijn mid-fi ontwerp.

### Reflectie

- Het ontwerp is overzichtelijk en goed leesbaar op gsm.
- Te veel informatie maakt het snel onduidelijk.
- Volgende keer zou ik extra varianten testen zoals dark mode.



## Les 8: Tailwind Verder Aanpassen

### Wat we hebben gedaan

- Verder gewerkt met Tailwind om de layout consistenter te maken.
- Spacing, kleuren en schaduwen verfijnd.
- Kleine hover-effecten toegevoegd aan knoppen en navigatie.

### Waarom deze aanpassingen

- Om de website visueel rustiger en duidelijker te maken.
- Zodat belangrijke elementen sneller opvallen.
- Voor een betere en aangenamere gebruikerservaring.

### Experimenten

- Geëxperimenteerd met verschillende kleurencombinaties.
- Getest hoe kleine animaties de site levendiger maken.

### Reflectie

- Ik kreeg meer inzicht in hoe Tailwind efficiënt gebruikt kan worden.
- Kleine aanpassingen kunnen een groot visueel verschil maken.

## Les 9: Tailwind Probleem & Oplossing

### Wat ik heb gemerkt

- Ik merkte dat ik Tailwind verkeerd had toegepast in mijn project.
- De structuur van mijn bestanden klopte niet met hoe Tailwind hoort te werken.
- Daardoor werkte styling niet altijd zoals verwacht.

### Wat ik heb aangepast

- Ik heb alle nodige bestanden verplaatst naar **HTML\_09\_TWND**.
- Daar ben ik opnieuw begonnen met een correcte Tailwind-setup.
- De HTML-structuur is opgeschoond en overzichtelijker gemaakt.

### Wat ik hieruit heb geleerd

- Dat een juiste mappenstructuur essentieel is voor Tailwind.
- Dat fouten soms pas duidelijk worden wanneer alles samenkomt.
- Dat opnieuw beginnen soms sneller is dan blijven proberen te fixen.

### Reflectie

- Deze les hielp me beter begrijpen hoe Tailwind echt werkt.
- Ik weet nu waar ik beter moet op letten bij nieuwe projecten.
- Het eindresultaat is properder en logischer opgebouwd.