

THERMALAI
AFSTUDEER EXPLORATIE
DINAND DAP 2025

01

VOORWOORD

Dit verslag is het resultaat van een persoonlijke fascinatie voor de onzichtbare wereld achter onze technologie. Kunstmatige Intelligentie is in korte tijd een onmisbaar onderdeel van ons leven geworden, een krachtige tool die ons helpt slimmer en sneller te werken. Maar terwijl we ons vergapen aan de mogelijkheden, blijven de ecologische kosten – het immense verbruik van energie, water en grondstoffen – vaak buiten beeld. Deze paradox vormde het startpunt van mijn onderzoek.

Hoe kunnen we iets dat zo abstract en onzichtbaar is, toch begrijpelijk en zelfs voelbaar maken? Deze vraag was de drijvende kracht achter dit project. Het was een zoektocht die verder ging dan traditioneel interfaceontwerp en me leidde naar de wereld van embodied cognition: het idee dat we de wereld niet alleen met ons hoofd begrijpen, maar met ons hele lichaam.

Dit project was een uitdagende, maar ontzettend leerzame reis. Ik wil graag mijn dank uitspreken aan de respondenten van de enquête en het interview, die met hun openhartige antwoorden onmisbare inzichten hebben geleverd. Hun perspectieven waren cruciaal in de vormgeving van het uiteindelijke concept.

Ik hoop dat dit verslag niet alleen een ontwerpproces documenteert, maar ook de lezer inspireert om met een nieuwe blik naar de technologie te kijken die we dagelijks zo vanzelfsprekend gebruiken.

INHOUDSOPGAVE

01 VOORWOORD	PG 02
02 INLEIDING	PG 05
03 POSITIONERING	PG 06
04 HOOFDVRAAG & DEELVRAGEN	PG 07
05 DOELSTELLINGEN & ONTWERPCRITERIA	PG 08
06 DOELGROEP	PG 09
07 DESKRESEARCH	PG 10
08 INZICHTEN DEELVRAGEN	PG 16
09 FIELDRESEARCH	PG 17
10 EXPERIMENTEN	PG 20

INHOUDSOPGAVE

11 VERDIEPING & INSPIRATIE	PG 30
12 CONCEPTEN EMBODIED EXPERIENCE	PG 34
13 STORYTELLING	PG 49
14 VAN CONCEPT NAAR CONCREET PRODUCT	PG 50
15 CONCEPTEN INTERFACE	PG 53
16 DEFINITIEF ONTWERP	PG 61
17 BIJLAGEN	PG 68
18 BRONNENLIJST	PG 70

02

INLEIDING

Kunstmatige Intelligentie (AI) is in korte tijd een onmisbaar hulpmiddel geworden in ons dagelijks leven. Achter de naadloze interfaces van systemen als ChatGPT en Midjourney schuilt echter een realiteit met een aanzienlijke ecologische voetafdruk, van het immense energieverbruik van datacenters tot de groeiende berg elektronisch afval. Het kernprobleem is dat deze impact voor de eindgebruiker volledig abstract en onzichtbaar is, wat bewustwording in de weg staat.

Dit project pakt dat probleem aan en is opgebouwd rond de hoofdvraag: "Hoe kan een embodied experience worden ingezet om gebruikers van KI-toepassingen (20-30 jaar) bewust te maken van de negatieve milieu-impact, met als doel een waardevolle reflectie op hun eigen KI gebruik?"

Om deze vraag te beantwoorden, neemt dit verslag de lezer mee door het volledige ontwerpproces. We starten met een onderbouwing van de positionering, de doelgroep en de onderzoeksopzet. Vervolgens worden de inzichten uit het desk- en fieldresearch gepresenteerd, die de basis vormen voor een reeks praktische experimenten. De leerpunten uit deze fases leiden tot de ontwikkeling van verschillende interface-concepten en monden uiteindelijk uit in de presentatie van het definitieve, onderbouwde prototype.

03

POSITIONERING

Als ontwerper ben ik gefascineerd door de groeiende aanwezigheid van Kunstmatige Intelligentie (KI) in ons dagelijks leven. Hoewel KI veelbelovend is, maak ik me zorgen over de onzichtbare milieu-impact die deze technologie heeft. Mijn afstudeerproject richt zich daarom op het onderzoeken van deze impact en hoe we deze kunnen verminderen zonder innovatie te stoppen.

Mijn werk richt zich specifiek op jonge volwassenen tussen de 20 en 30 jaar die dagelijks KI-tools gebruiken. Deze doelgroep is al digitaal vaardig en heeft een groeiend milieubewustzijn, maar mist vaak concrete kennis over de ecologische voetafdruk van hun eigen digitale gedrag.

Ik geloof dat de meest effectieve manier om dit bewustzijn te creëren is via belichaamde ervaringen (embodied experiences). Geïnspireerd door de theorie van belichaamde cognitie, wil ik de abstracte milieu-impact van KI fysiek voelbaar en tastbaar maken. Door zintuiglijke en lichamelijke elementen te gebruiken, kunnen gebruikers de verborgen kosten van KI niet alleen begrijpen, maar ook echt voelen. Dit is een krachtigere benadering dan alleen het geven van informatie, omdat het een diepere, emotionele connectie creëert.

Met mijn project wil ik deze jonge KI-gebruikers niet alleen informeren, maar hen ook aanzetten tot nadenken over hun eigen gedrag. Het doel is dat zij niet alleen weten, maar ook voelen wat de impact van KI op onze planeet is, en zo bijdragen aan een duurzamere technologische toekomst.

HOODVRAAG & DEELVRAGEN

"Hoe kan een embodied experience worden ingezet om gebruikers van KI-toepassingen (20-30 jaar) bewust te maken van de negatieve milieu-impact, met als doel een waardevolle reflectie op hun eigen KI gebruik?"

Deelvragen:

Wat zijn de milieu-effecten van KI?

Hoe maak ik gebruikers bewust van de milieugevolgen van KI?

Wie zijn de mensen die KI gebruiken maar geen idee hebben van de gevolgen?

Hoe gebruik ik embodied cognition om de gebruiker lichamelijk te laten leren?

Hoe creeer ik een meetbaar prototype voor mijn doelgroep?

DOELSTELLINGEN & ONTWERPCRITERIA

Doelen:

- Verhoogd Bewustzijn: De gebruiker moet een significant verhoogd bewustzijn ontwikkelen van de onzichtbare energie-, water- en grondstofkosten van KI-gebruik.
- Gedragsreflectie & -aanpassing: Het product moet de gebruiker aanzetten tot reflectie over hun eigen KI-gebruik en, idealiter, leiden tot het overwegen of toepassen van duurzamere digitale gewoonten.
- Betrokkenheid & Memorabiliteit: Het product moet een memorabele en emotioneel betrokken ervaring bieden die langer blijft hangen dan traditionele informatieoverdracht.

Ontwerpcriteria:

- Fysieke Feedback (Embodied): De milieu-impact moet direct en onmiddellijk fysiek voelbaar zijn.
- Visuele Feedback: Visuele signalen moeten de fysieke feedback ondersteunen en de status van de KI-impact intuïtief communiceren.
- Stabiel Systeem: Het systeem moet stabiel functioneren en consistent de beoogde feedback geven.
- Eenvoudig te Begrijpen: De relatie tussen de actie van de gebruiker, de KI-output en de fysieke/visuele feedback moet direct duidelijk zijn zonder uitgebreide uitleg.
- Aansprekend: Het concept en de uitvoering moeten aansluiten bij de interesses, dagelijks leven en digitale levensstijl van de doelgroep.

DOELGROEP

Voor dit project is een bewuste en strategische keuze gemaakt voor de doelgroep: jongvolwassenen in de leeftijdscategorie van 20 tot 30 jaar. Deze generatie is een intensieve gebruiker van Kunstmatige Intelligentie (AI) en integreert deze technologie vanzelfsprekend in het dagelijks leven, of het nu voor studie, werk of entertainment is. Denk hierbij aan het gebruik van ChatGPT voor het opstellen van teksten, DALL-E voor het genereren van afbeeldingen, of de aanbevelingsalgoritmes van streamingdiensten als Spotify en Netflix.

De keuze voor deze specifieke groep is cruciaal voor het slagen van het project. Omdat zij de grootste gebruikers zijn, kan het stimuleren van een impactvolle reflectie op hun AI-gebruik al leiden tot een significant verschil. Bovendien staat deze generatie, die is opgegroeid met snelle technologische ontwikkelingen, open voor nieuwe ideeën en interactieve ervaringen. Dit maakt hen de ideale doelgroep voor een prototype dat de impact van AI laat voelen in plaats van er alleen over te lezen.

Het centrale knelpunt waar deze doelgroep mee worstelt, is de onzichtbaarheid van de milieu-impact van AI. Het is voor hen onmogelijk om te weten hoeveel energie en water een simpele zoekopdracht of het genereren van een afbeelding kost. Hoewel er een algemene interesse in duurzaamheid is, ontbreekt het hen aan concreet inzicht en praktische handvatten om hun digitale gedrag te verduurzamen. Ze weten niet hoe groot hun impact is, wat ze kunnen doen om deze te verkleinen, en missen de transparantie van techbedrijven hierover.

Dit project wil die kloof overbruggen. Door jongvolwassenen bewust te maken, worden zij niet alleen duurzamer, maar ook een voorbeeld voor hun vrienden. Het doel is dus een prototype dat aanzet tot diepe reflectie en duurzamer gedrag rondom AI.

DESKRESEARCH

Wat zijn de milieu-effecten van KI?

Energieverbruik en koolstofvoetafdruk

De snelle ontwikkeling en brede inzet van kunstmatige intelligentie (KI) hebben veel nieuwe mogelijkheden gebracht voor verschillende sectoren. Tegelijkertijd zorgt deze groei voor aanzienlijke milieubelasting, iets waar veel gebruikers zich niet van bewust zijn. Naarmate KI-systeem groter en complexer worden, stijgt ook hun energieverbruik, wat vragen oproept over hun duurzaamheid in tijden van klimaatverandering.

Het trainen van grote KI-modellen vraagt veel rekenkracht en dus veel elektriciteit. Een bekend voorbeeld is het trainen van GPT-3, wat ongeveer 1.287 megawattuur aan stroom verbruikte. Dit zorgde voor de uitstoot van zo'n 502 ton CO₂ – vergelijkbaar met wat 112 auto's in een jaar uitstoten (AG Connect, 2023; Undetectable AI, 2025). En dat is alleen voor de training van het model. Het dagelijks gebruik, zoals het beantwoorden van vragen, verbruikt daarna ook nog veel energie.

De zogenoemde inferentiefase – waarin het model gebruikt wordt om reacties te geven – kan zelfs tot 60% van het totale energieverbruik van een model beslaan (KIA Community, 2024). Bijvoorbeeld: één vraag aan ChatGPT verbruikt ongeveer 100 keer meer energie dan een normale Google-zoekopdracht (Tunity, 2024; Undetectable AI, 2025). Aangezien miljarden van dit soort interacties elke dag plaatsvinden, is het totale energieverbruik enorm.

De milieubelasting komt niet alleen door het energieverbruik. Ook de hardware die nodig is voor KI—zoals chips, servers en koelsystemen—heeft impact. Voor de productie van deze onderdelen zijn zeldzame grondstoffen, veel water en energie nodig. En aan het einde van hun levensduur moeten deze materialen ook nog verwerkt of weggegooid worden.

Datacenters, waar KI-systeem draaien, zijn wereldwijd grote energieverbruikers geworden. Volgens het Internationaal Energieagentschap (IEA) zal hun verbruik tegen 2026 mogelijk 1.000 terawattuur bereiken, ongeveer evenveel als heel Japan nu gebruikt (IEA, 2024; IO+, 2024). Hoewel dit niet alleen voor KI is, neemt het deel voor KI wel snel toe. Google laat bijvoorbeeld weten dat 15% van de energie in hun datacenters naar machine learning gaat – en dat groeit (Hardware Info, 2023).

De klimaatimpact van KI hangt ook sterk af van waar de datacenters staan en welke energiebron ze gebruiken. In Finland draait een datacenter van Google op 97% schone energie (ESG News, 2024; Google, n.d.). Maar in delen van Azië gebruiken datacenters vooral fossiele brandstoffen, wat leidt tot meer CO₂-uitstoot en vervuiling (InkopersCafe.nl, 2025).

Tot slot: steeds grotere KI-modellen vragen meer data, meer parameters en dus ook meer energie. Dit roept de vraag op of deze groei wel houdbaar is, en of de extra prestaties van nieuwe modellen opwegen tegen de toenemende milieukosten.

Waterverbruik & Grondstofeffecten

Naast het hoge energieverbruik en de CO₂-uitstoot hebben KI-systeem ook een andere, minder zichtbare milieu-impact: waterverbruik. Datacenters gebruiken namelijk veel water om hun servers te koelen, vooral wanneer ze zware KI-taken uitvoeren. Gemiddeld gebruikt een datacenter ongeveer 1,7 liter water per kilowattuur aan energie (AG Connect, 2023). Dit betekent dat grote technologiebedrijven jaarlijks miljoenen liters zoet water nodig hebben om hun KI-systeem draaiende te houden.

Uit duurzaamheidsrapporten blijkt hoe groot deze impact is. Microsoft meldde een toename van 34% in hun wereldwijde waterverbruik tussen 2021 en 2022 (Microsoft, 2023). Bij Google steeg het verbruik in diezelfde periode met 20% (Google, 2023). Onderzoekers, zoals Shaolei Ren van UC Riverside, wijzen vooral naar het toenemende gebruik van KI als de oorzaak van deze stijging (Ren et al., 2021). Zelfs gewone KI-gebruiksituaties kunnen meer water verbruiken dan je zou verwachten. Een eenvoudige sessie van 10 tot 50 vragen aan GPT-3 verbruikt ongeveer een halve liter water. Een e-mail van 100 woorden die wordt gegenereerd door GPT-4 kan zelfs meer dan 500 milliliter kosten – bijna een volle fles water (AG Connect, 2023; Undetectable AI, 2025).

Dit wordt een groter probleem als datacenters zich bevinden in regio's met watertekorten of droogte. Daar kan het extra waterverbruik van KI de bestaande schaarste verergeren. Inwoners van deze gebieden zijn vaak niet de mensen die het meeste baat hebben bij KI-technologie, wat leidt tot zorgen over milieurechtvaardigheid (Stichting Internet Domeinregistratie Nederland, 2024).

De milieubelasting van KI gaat ook verder dan alleen energie en water. Voor het bouwen van de benodigde hardware worden zeldzame aardmetalen en edelmetalen gebruikt. De winning van deze materialen heeft grote gevolgen voor het milieu, zoals vervuiling, vernietiging van natuurgebieden, en hoog energieverbruik (EY, 2024; InkopersCafe.nl, 2025). Daarnaast veroudert KI-hardware snel, waardoor veel elektronisch afval ontstaat dat moeilijk te recyclen is (EY, 2024).

De totale milieu-impact van KI omvat alles van onderzoek en ontwikkeling tot het dagelijks gebruik en onderhoud. Hierbij horen bijvoorbeeld ook de energie die onderzoekers gebruiken tijdens het testen van modellen, de materialen die nodig zijn om grote datasets te verzamelen en onderhouden, en de infrastructuur voor het beheren en updaten van KI-systeem.

Geografische verdeling van milieueffecten

De milieueffecten van KI zijn niet gelijk verdeeld over de wereld. Sommige regio's ervaren vooral de nadelen, terwijl andere vooral profiteren van de voordelen. Dit roept belangrijke vragen op over milieurechtvaardigheid en de verantwoordelijkheid van landen en bedrijven wereldwijd.

Grote technologiebedrijven bouwen hun datacenters vaak op locaties waar energie goedkoop is, waar de regelgeving gunstig is, of waar ze belastingvoordelen krijgen. Hierdoor ontstaan er gebieden met veel datacenters en dus ook veel milieubelasting. Een bekend voorbeeld is Noord-Virginia, dat inmiddels de grootste concentratie datacenters ter wereld heeft. Deze centra gebruiken grote hoeveelheden elektriciteit en water, wat druk zet op de lokale omgeving (Stichting Internet Domeinregistratie Nederland, 2024).

De situatie wordt nog complexer door de wereldwijde aard van KI-productieketens. Zo worden KI-modellen meestal ontwikkeld en gebruikt in rijke landen, terwijl de grondstoffen voor de hardware vaak worden gewonnen in armere landen. Daar zijn de milieuregels vaak minder streng. Bovendien komt het elektronisch afval van oude apparatuur vaak terecht in lage-inkomenslanden, waar recycling minder veilig gebeurt en gezondheidsrisico's ontstaan voor de lokale bevolking (EY, 2024; InkopersCafe.nl, 2025).

Ook klimaatverandering speelt een rol. Regio's die nu al kwetsbaar zijn voor droogte, overstromingen of extreme weersomstandigheden kunnen extra onder druk komen te staan door de groei van KI-infrastructuur. Ironisch genoeg zijn deze gemeenschappen vaak het minst verantwoordelijk voor klimaatverandering en profiteren ze ook het minst van de voordelen van KI.

En die voordelen zijn er wel degelijk. KI kan bijvoorbeeld helpen bij milieubescherming, zoals het voorspellen van overstromingen of het efficiënter gebruiken van grondstoffen. Maar ook deze toepassingen zijn niet gelijk toegankelijk. Rijke landen hebben vaker de benodigde infrastructuur en kennis, terwijl armere landen dat meestal niet hebben.

Deze ongelijke verdeling van lasten en baten maakt de ontwikkeling van KI tot een ethisch vraagstuk. Het stelt ons voor belangrijke vragen: Wie is verantwoordelijk voor de schade? Moeten getroffen regio's gecompenseerd worden? En vooral: hoe zorgen we voor een eerlijker gebruik van KI wereldwijd, met aandacht voor zowel mensen als het milieu?

Onbekende en verborgen kosten van KI-energieverbruik

Hoewel steeds meer mensen zich zorgen maken over de milieu-impact van kunstmatige intelligentie (KI), blijven veel aspecten van het energie- en grondstofverbruik onduidelijk of zelfs bewust verborgen. Deze gebrekke transparantie maakt het moeilijk om de werkelijke milieukosten van KI goed te begrijpen of aan te pakken.

Een belangrijk probleem is dat er geen standaard manier is voor bedrijven om te rapporteren over hun milieubelasting.

In tegenstelling tot andere sectoren zijn techbedrijven niet verplicht om informatie te delen over het energieverbruik, de CO₂-uitstoot of het watergebruik van hun KI-systemen. Daardoor is het lastig voor gebruikers, beleidsmakers en bedrijven zelf om verantwoorde keuzes te maken.

Onderzoekers, zoals Jesse Dodge van het Allen Institute for AI, geven aan dat ontwikkelaars geen gemakkelijke of betrouwbare toegang hebben tot gegevens over de klimaateffecten van hun systemen. Zelfs ontwikkelaars die hun systemen milieuvriendelijker willen maken, hebben te weinig informatie om dat te doen (Dodge et al., 2022).

Bovendien beschouwen veel bedrijven gegevens over hun KI-infrastructuur als bedrijfsgeheim. Ze delen bijvoorbeeld niet hoeveel energie of grondstoffen hun systemen gebruiken. Hierdoor kunnen onafhankelijke onderzoekers de milieuclaims van deze bedrijven niet controleren, en blijft het echte effect op het milieu onduidelijk.

Er zijn ook indirekte milieukosten die vaak over het hoofd worden gezien. Denk aan de milieuvervuiling door het winnen van grondstoffen voor hardware, de energie die wordt gebruikt in internationale productieketens, en de impact van elektronisch afval. Deze factoren worden zelden meegenomen in berekeningen van de totale milieuoefactuur van KI, maar ze zijn wel degelijk belangrijk.

Daarnaast is er sprake van toekomstige, moeilijk voorspelbare kosten. Omdat KI-modellen steeds groter en complexer worden, kan hun behoefte aan energie en grondstoffen snel blijven toenemen. Als hier geen maatregelen worden genomen, kan de milieudruk van KI in de toekomst flink uit de hand lopen.

Wat misschien nog het meest zorgwekkend is: gebruikers zijn zich vaak helemaal niet bewust van de impact van hun interacties met KI. Of je nu een chatbot gebruikt, een afbeelding genereert of een KI-aanbeveling volgt, je ziet nergens hoeveel energie dat kost of hoeveel grondstoffen daarvoor nodig zijn. Zoals de Amerikaanse senator Edward Markey zei:

"Het is nu niet mogelijk om te weten hoe jouw KI-verzoek de koolstofemissies of zoetwatervoorraden beïnvloedt" (Markey, 2024).

Omdat deze kosten onzichtbaar zijn voor gebruikers, voelen mensen ook geen motivatie om hun KI-gebruik te beperken of te kiezen voor duurzamere alternatieven. Dit vergroot het risico op een vicieuze cirkel van overmatig verbruik.

Tot slot maakt dit gebrek aan transparantie het moeilijk voor overheden om effectief beleid te maken. Zolang er geen duidelijke informatie is over de milieubelasting van KI, kunnen beleidsmakers geen goede normen en regels opstellen om de technologie duurzamer te maken.

Hoe maak ik gebruikers bewust van de milieugevolgen?

Huidige bewustzijnsniveau en kenniskloven

Ondanks de toenemende milieu-impact van kunstmatige intelligentie (KI), is het bewustzijn hierover bij het brede publiek nog steeds erg laag. De meeste mensen die KI gebruiken, weten weinig tot niets over het energieverbruik, de uitstoot van broeikasgassen of het watergebruik die met deze technologie gepaard gaan. Dit gebrek aan kennis vormt een belangrijke belemmering voor duurzamer gebruik van KI en voor het stimuleren van milieuvriendelijke technologische ontwikkelingen.

Er zijn verschillende redenen voor dit gebrek aan bewustzijn. Ten eerste zijn de milieueffecten van digitale technologieën, en KI in het bijzonder, voor gebruikers niet zichtbaar. Waar traditionele industrieën vaak duidelijk herkenbare milieuschade veroorzaken, gebeuren de milieueffecten van KI vooral in datacenters die ver weg zijn van de gebruiker. Als iemand bijvoorbeeld ChatGPT gebruikt om een tekst te schrijven of een afbeelding maakt met DALL-E, is er niets te zien van de middelen die daarvoor nodig zijn.

Ten tweede geven technologiebedrijven zelden duidelijke informatie over de milieu-impact van hun KI-diensten. Hoewel sommige bedrijven duurzaamheidsrapporten publiceren, bevatten deze meestal alleen algemene cijfers. Ze zeggen bijvoorbeeld niets over het energieverbruik van een specifieke toepassing. Veel gebruikers weten daarom niet dat één ChatGPT-interactie ongeveer 100 keer meer energie verbruikt dan een zoekopdracht op Google, of dat het schrijven van een korte e-mail met GPT-4 meer dan een halve liter water kan kosten (AG Connect, 2023; Tunity, 2024; Undetectable AI, 2025).

Ten derde richten media en publieke discussies zich vooral op de voordelen en risico's van KI voor economie en maatschappij, en minder op de ecologische gevolgen. Als het milieu ter sprake komt, gaat het vaak over hoe KI kan helpen bij duurzaamheid, in plaats van over de milieubelasting van KI zelf.

Ook in het onderwijs wordt weinig aandacht besteed aan de milieukant van digitale technologie. Leerlingen leren hoe ze technologie kunnen gebruiken, maar krijgen zelden uitleg over de grondstoffen en milieueffecten die ermee gepaard gaan. Zelfs in informatica- en datawetenschapsopleidingen komt duurzaamheid meestal slechts zijdelings aan bod.

Onderzoek bevestigt deze bewustzijnskloof. In een enquête uit 2024 onder ruim 2.200 middelbare scholieren bleek dat bijna iedereen KI gebruikte, bijvoorbeeld via YouTube of Spotify, maar dat slechts enkelen wisten dat dit milieukosten met zich meebrengt. Ook onder volwassenen blijkt dat hoewel velen zich zorgen maken over het milieu, bijna niemand denkt aan de ecologische impact van digitale technologie (Kennisnet, 2024).

Zonder dit bewustzijn kunnen mensen geen duurzamere keuzes maken of bedrijven aanzetten tot milieuvriendelijker praktijken. Het verkleinen van deze kenniskloof is daarom een belangrijke eerste stap naar een duurzamere ontwikkeling en toepassing van KI.

Effectieve communicatiestrategieën

Om mensen bewust te maken van de milieu-impact van kunstmatige intelligentie (KI), zijn duidelijke en effectieve communicatiestrategieën nodig. Deze strategieën moeten de onzichtbare gevolgen van KI begrijpelijk en relevant maken voor gebruikers. Verschillende aanpakken bieden goede mogelijkheden om het bewustzijn te vergroten en duurzamer KI gebruik te stimuleren.

Een effectieve methode is om het verbruik van grondstoffen te vertalen naar herkenbare voorbeelden. In plaats van abstracte cijfers te gebruiken, helpt het om de impact van KI uit te leggen met vergelijkingen uit het dagelijks leven. Bijvoorbeeld: het trainen van een groot taalmodel kost evenveel elektriciteit als 112 auto's die een jaar rijden, of één gesprek met ChatGPT verbruikt ongeveer evenveel water als een halve literfles. Zulke beelden maken de gevolgen beter voorstelbaar (AG Connect, 2023; Undetectable AI, 2025).

Beelden en visualisaties kunnen ook krachtig zijn. Infografieken, animaties of interactieve schema's laten op een eenvoudige manier zien hoeveel energie of water een KI-systeem gebruikt. Een realtime visualisatie van waterverbruik tijdens een interactie met KI zou bijvoorbeeld direct effect kunnen hebben.

Ook gepersonaliseerde feedback kan helpen. Toepassingen zouden gebruikers kunnen laten zien wat hun eigen digitale voetafdruk is, net zoals een fitness-app beweging bijhoudt. Dit kan mensen aanzetten tot bewuster gedrag en hen het gevoel geven invloed te hebben.

Daarnaast kunnen vergelijkingen tussen KI-diensten gebruikers helpen om duurzamere keuzes te maken. Zoals energielabels op apparaten, zouden er milieulabels kunnen komen voor KI-diensten. Dit stimuleert ook bedrijven om hun systemen zuiniger te maken.

Als informatie over energie- of waterverbruik direct bij het gebruik van een KI-toepassing wordt getoond, is de kans groter dat gebruikers er aandacht aan besteden. Een kleine melding tijdens een interactie, bijvoorbeeld over hoeveel water of stroom iets kost, kan het bewustzijn verhogen zonder extra moeite voor de gebruiker.

Verhalen die de milieu-impact van KI koppelen aan thema's als klimaatverandering of watergebruik in de regio, kunnen het onderwerp tastbaarder maken. Dergelijke verhalen verbinden technologische ontwikkelingen met maatschappelijke en persoonlijke gevolgen.

Onderwijs speelt ook een belangrijke rol. Door dit onderwerp op te nemen in het lesprogramma op scholen, universiteiten en in media, kunnen jongeren beter begrijpen welke effecten technologie op het milieu heeft en waarom duurzame keuzes belangrijk zijn.

Tot slot is samenwerking nodig tussen technologiebedrijven, milieuorganisaties, overheden en het onderwijs. Door samen te werken en op elkaar afgestemde boodschappen te verspreiden, kan een breed publiek worden bereikt en wordt voorkomen dat de verantwoordelijkheid bij slechts één partij terechtkomt.

Deze communicatie is het meest effectief wanneer ze niet alleen informeert, maar gebruikers ook aanzet tot actie. Daarom moeten boodschappen over milieu-impact gekoppeld worden aan praktische tips en manieren om te pleiten voor duurzamere ontwikkeling van KI.

Transparantie en rapportagekaders

Effectieve bewustwording over de milieu-impact van kunstmatige intelligentie (KI) vereist betrouwbare en vergelijkbare informatie. Daarom is het belangrijk om transparante en gestandaardiseerde systemen te ontwikkelen die gebruikers inzicht geven in de ecologische gevolgen van KI-gebruik.

Een eerste stap is het opstellen van duidelijke meetmethoden voor de ecologische voetafdruk van KI. Denk aan standaarden die niet alleen het energieverbruik meten, maar ook de uitstoot van broeikasgassen, het watergebruik en het gebruik van grondstoffen. Deze metingen moeten betrekking hebben op de hele levenscyclus van een KI-systeem – van ontwikkeling en training tot dagelijks gebruik. Door standaardisatie kunnen gegevens uit verschillende bronnen eerlijk worden vergeleken.

Internationaal zijn al verschillende initiatieven gestart. Zo werkt de Internationale Organisatie voor Standaardisatie (ISO) aan criteria voor 'duurzame KI' (ISO, n.d.). De Green Software Foundation ontwikkelt de Software Carbon Intensity (SCI) standaard, die de CO₂-uitstoot van softwaretoepassingen meet, inclusief KI (Green Software Foundation, n.d.). Deze initiatieven helpen bij het opbouwen van een gedeelde taal rond milieu-impact.

Ook wetgeving begint een rol te spelen. In Europa verplicht de AI Act rapportage over energie- en grondstofgebruik van risicovolle KI-systemen (Europese Commissie, 2024). In de VS wordt wetgeving overwogen die de overheid verplicht om de milieu-impact van KI in kaart te brengen en standaarden te ontwikkelen voor toekomstige rapportage. Dergelijke regels zorgen ervoor dat milieugegevens niet enkel afhankelijk zijn van vrijwillige bedrijfsrapportages.

De betrouwbaarheid van rapportages neemt toe door controle van derden. Onafhankelijke audits garanderen dat bedrijven correcte informatie delen. Organisaties zoals de Carbon Trust doen dit al in andere sectoren; vergelijkbare modellen kunnen ook voor KI worden gebruikt.

Milieulabels op producten kunnen gebruikers helpen bij duurzame keuzes. Net als energielabels op koelkasten, zouden duidelijke labels voor KI-diensten inzicht kunnen geven in hun ecologische impact – zowel in absolute cijfers als in vergelijking met andere opties.

Realtime tools die tijdens het gebruik informatie tonen over energie- of waterverbruik, maken de impact van KI direct zichtbaar. Een dashboard dat het stroom- of watergebruik van een sessie laat zien, helpt gebruikers om de optelsom van kleine handelingen te begrijpen.

Open toegankelijke databanken met milieu-informatie over KI maken het mogelijk voor onderzoekers, journalisten en maatschappelijke organisaties om deze data te analyseren en gebruiken in hun werk. Zo wordt transparantie vergroot en kan er kennis gedeeld worden.

Tot slot bieden duurzaamheidsrapporten van bedrijven waardevol inzicht, vooral als ze specifiek ingaan op KI. Ze moeten meer doen dan cijfers rapporteren: ze moeten ook laten zien welke concrete stappen worden genomen om milieuschade te beperken, welke doelen zijn gesteld en wat de voortgang is.

Samen vormen deze transparantie- en rapportagesystemen een belangrijke basis. Ze geven gebruikers houvast voor bewuste keuzes, helpen overheden bij het maken van beleid en zorgen voor verantwoordelijkheid bij technologiebedrijven. Door de milieu-impact van KI zichtbaar te maken, wordt een duurzamer gebruik mogelijk.

Gebruikersempowerment en besluitvormingstools

Naast bewustwording en transparantie is het belangrijk dat gebruikers ook de juiste hulpmiddelen en ondersteuning krijgen om daadwerkelijk duurzamere keuzes te maken in hun gebruik van kunstmatige intelligentie (KI). Er bestaan verschillende veelbelovende strategieën die gebruikers in staat stellen om hun impact te verminderen.

Een eerste aanpak is het aanbieden van instelbare duurzaamheidsopties binnen KI-toepassingen. Gebruikers zouden bijvoorbeeld kunnen kiezen tussen modellen met verschillende complexiteit: een lichter, energiezuiniger model voor eenvoudige taken, en een krachtiger model wanneer dat echt nodig is. Denk aan een "eco-modus" in een KI-schrijfassistent die standaard een efficiënt model gebruikt, maar de optie biedt voor zwaardere functionaliteit als gewenst.

Daarnaast kunnen geïntegreerde koolstofcalculators gebruikers inzicht geven in de milieueffecten van hun acties. Deze tools zouden realtime informatie kunnen tonen over energieverbruik, uitstoot en waterverbruik, samen met tips om deze impact te beperken. Door dit direct zichtbaar te maken tijdens het gebruik, kunnen gebruikers beter geïnformeerde beslissingen nemen.

Een ander waardevol hulpmiddel is een persoonlijke tracker die de totale milieuvoetafdruk van een gebruiker bijhoudt. Net als bij een gezondheids- of budgetapp kunnen gebruikers hun gebruik monitoren, doelen stellen voor reductie, en voortgang volgen. Deze trackers kunnen ook een vergelijking maken met gemiddelden, wat motivatie kan geven om bewuster om te gaan met KI-gebruik.

Ook kunnen keuzehulpmiddelen gebruikers ondersteunen bij het afwegen van prestaties, functionaliteit en milieu-impact bij het kiezen van KI-diensten. Door deze informatie overzichtelijk te presenteren, kunnen gebruikers beter bepalen welke optie het beste past bij hun behoeften en hun waarden.

Platformen voor collectieve actie bieden de mogelijkheid om als groep een verschil te maken. Door gezamenlijke doelen te stellen en collectieve besparingen zichtbaar te maken, ontstaat een gevoel van gemeenschap en gezamenlijke verantwoordelijkheid. Dit kan individuele betrokkenheid versterken.

Educatieve middelen en praktische handleidingen zijn essentieel om gebruikers te helpen duurzamer om te gaan met KI. Deze materialen kunnen tips bevatten zoals het combineren van meerdere verzoeken, hergebruik van outputs, het kiezen van kleinere modellen voor eenvoudige taken, of het plannen van KI-taken tijdens pieken in hernieuwbare energie.

Tot slot zijn feedbackkanalen belangrijk waarmee gebruikers hun zorgen of ideeën kunnen delen met technologiebedrijven. Denk aan de mogelijkheid om duurzame functies te beoordelen, suggesties te doen of deel te nemen aan panels. Zo kunnen bedrijven hun producten verbeteren op basis van gebruikersinput.

Deze strategieën erkennen dat bewustzijn alleen niet genoeg is. Door praktische tools, sociale ondersteuning en concrete voordelen aan te bieden, kunnen gebruikers werkelijk bijdragen aan duurzamer KI-gebruik – zonder dat de verantwoordelijkheid volledig bij hen wordt gelegd. Ze maken ruimte voor individueel initiatief binnen een bredere beweging richting structurele verandering.

INZICHTEN DEELVRAGEN

Voorafgaand aan het ontwerpproces is er deskresearch uitgevoerd om de verborgen impact van AI-gebruik te doorgronden. Het doel was om te begrijpen hoe een ontwerp kan bijdragen aan bewustwording rondom de ecologische voetafdruk van AI. Uit dit onderzoek kwamen drie fundamentele inzichten naar voren die als leidraad dienen voor het prototype. Deze inzichten behandelen de onzichtbare milieukosten, de kracht van fysieke feedback en de effectiviteit van een metaforisch ontwerp.

Belangrijkste inzichten:

De Onzichtbare Kosten van KI:

Energie & Water: KI-systeem, met name de datacenters die ze draaien, verbruiken enorme hoeveelheden energie en water.

Grondstoffen & E-waste: De productie en snelle veroudering van KI-hardware leidt tot uitputting van grondstoffen en een groeiende afvalberg.

Bewustwording door Fysieke Ervaring:

Embodied Cognition: Abstracte data over milieu-impact kan begrijpelijk en voelbaar worden gemaakt door fysieke sensaties zoals warmte.

Directe Feedback: Een directe, fysieke reactie op een KI-actie creëert een duidelijke oorzaak-gevolg relatie en verhoogt het bewustzijn.

Effectieve Ontwerprincipes:

Metafoor van Overbelasting: Het gevoel van beklemming door warmte en/of pijnprikkels is een krachtige metafoor voor de overbelasting van het milieu.

FIELDRESEARCH

Naast het deskresearch was het essentieel om directe inzichten te verzamelen van de doelgroep zelf. Om een beter begrip te krijgen van hun gedrag, de context van hun AI-gebruik en hun houding ten opzichte van duurzaamheid, is er een onderzoek uitgevoerd. Dit bestond uit een interview en een bredere enquête. De analyse van deze data levert een scherp beeld op van de beleidingswereld van de gebruiker en legt de kern van het ontwerpprobleem bloot.

Hieronder worden de belangrijkste inzichten uit dit onderzoek gepresenteerd. Een volledige transcriptie van het interview is te vinden in de bijlage.

Belangrijkste Inzichten uit het Interview

- AI-gebruik is functioneel en vanzelfsprekend: De doelgroep ziet AI als een efficiënt hulpmiddel dat tijd bespaart en frictie wegneemt. Het gebruik is frequent en geïntegreerd in zowel werk- als privéroutines.
- Duurzaamheid is een thema, maar losgekoppeld van digitaal gedrag: Er is een duidelijke bereidheid om duurzame keuzes te maken in de fysieke wereld (voeding, transport, kleding), maar deze mentaliteit wordt nog niet toegepast op de digitale wereld.
- De impact van AI is volledig abstract en onzichtbaar: Het kernprobleem is de perceptie. AI "gebeurt in de cloud" en wordt niet geassocieerd met fysiek energie- of waterverbruik. Dit is de grootste barrière voor bewustwording.
- Concrete vergelijkingen zijn een eyeopener: Het kwantificeren van de impact (bv. "een AI-vraag = een uur een lamp aan") zou direct leiden tot een schokeffect en een intentie om het gedrag aan te passen.
- Behoefte aan inzicht, niet aan schuldgevoel: De doelgroep zoekt naar tools die inzicht geven en helpen bij het maken van bewuste keuzes. Ze willen niet bestraft worden, maar geholpen worden om het beter te doen. De vraag is niet om minder AI te gebruiken, maar om het efficiënter en bewuster te doen.
- Transparantie van bedrijven is gewenst: Er is een duidelijke behoefte aan informatie over de duurzaamheidsprestaties van verschillende techbedrijven, wat kan sturen in de keuze voor een bepaalde tool.

De resultaten van de enquête, waaraan de respondenten deelnamen, bevestigen en versterken de inzichten uit het deskresearch en het interview. De data toont een duidelijk beeld van het gedrag en de houding van de doelgroep ten aanzien van AI en duurzaamheid.

De belangrijkste conclusies zijn als volgt:

AI is diep geïntegreerd in het dagelijks leven. Een overweldigende meerderheid van de respondenten gebruikt AI-tools dagelijks (35,3%) of meerdere keren per week (35,3%). Dit gebruik is primair gericht op functionele doeleinden zoals studie/werk (76,5%), creatieve projecten (52,9%) en persoonlijke productiviteit (52,9%). Dit bevestigt dat AI een onmisbaar en efficiënt hulpmiddel is geworden in de routine van de doelgroep.

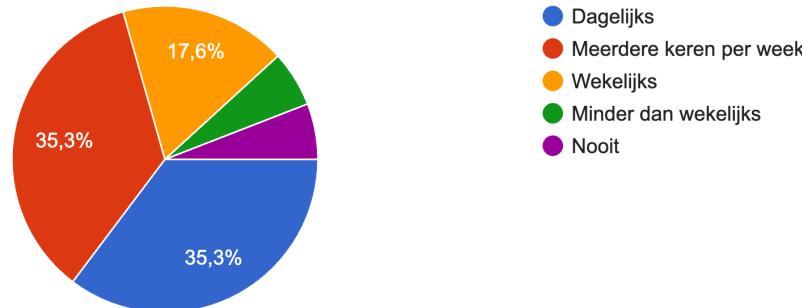
Er is een significante ‘awareness gap’ (bewustzijnsgat). Ondanks het frequente gebruik, was een grote meerderheid van de respondenten (64,7%) niet op de hoogte van het hoge energie- en waterverbruik van AI-systemen. Dit onderstreept de abstracte en onzichtbare perceptie van de milieu-impact van digitale technologie. Hoewel een groot deel zich redelijk tot heel verantwoordelijk voelt voor hun technologiegebruik (samen 82,4%), wordt deze verantwoordelijkheid nog niet gekoppeld aan de concrete impact van AI.

Er is een sterke behoefte aan inzicht en feedback. De doelgroep staat zeer positief tegenover het ontvangen van informatie. Maar liefst 64,7% geeft aan dat realtime feedback over de milieu-impact hen bewuster zou maken. De wens voor meer kennis over de impact (70,6%) en inzicht in het eigen verbruik (70,6%) is groot.

Fysieke en visuele feedback is de meest gewenste oplossing. De meest opvallende uitkomst is de vraag “Wat zou jou helpen om bewuster om te gaan met AI-gebruik?”. Hierop antwoordde een overweldigende 94,1% dat fysieke of visuele feedback hen zou helpen. Dit sluit naadloos aan bij de hypothese uit het deskresearch dat embodied cognition (het voelbaar maken van data) een krachtig middel is voor bewustwording. De openheid voor een interactieve ervaring die de impact laat voelen (52,9% ‘interessant’, 35,3% ‘misschien’) versterkt deze conclusie.

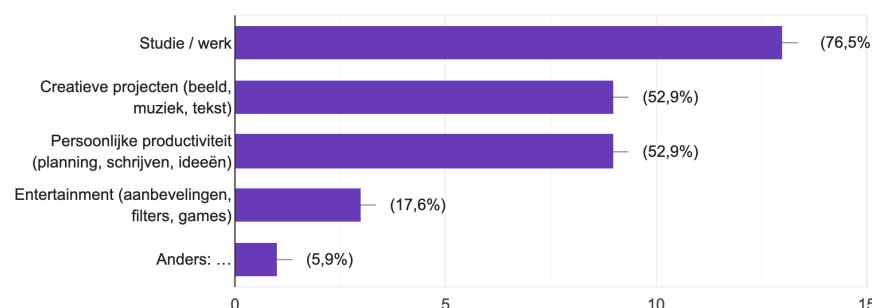
Samenvattend toont de enquête aan dat er een duidelijke kans ligt voor een ontwerp dat de onzichtbare impact van AI zichtbaar en voelbaar maakt. De doelgroep is zich nog onbewust van het probleem, maar staat tegelijkertijd open voor oplossingen die hen op een intuïtieve, niet-opdringerige manier inzicht geven. De voorkeur voor fysieke en visuele feedback biedt een concreet en gevalideerd startpunt voor het te ontwikkelen prototype.

Hoe vaak gebruik je AI tools?



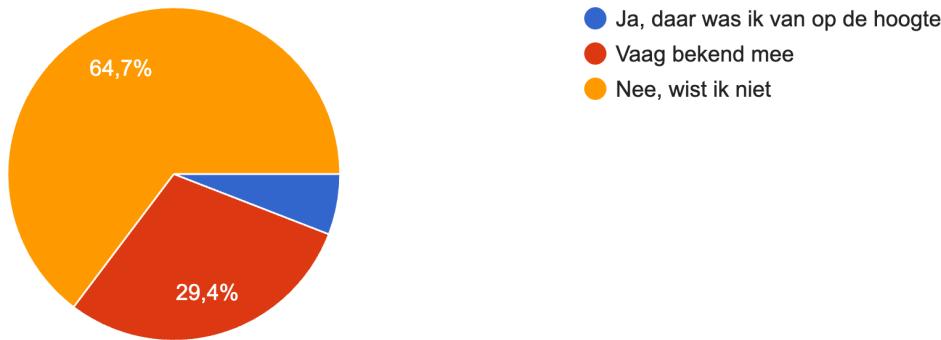
Figuur 1: Resultaat uit de enquête op de vraag “Hoe vaak gebruik je AI tools?”

Voor welke doeleinden gebruik je AI het meest?

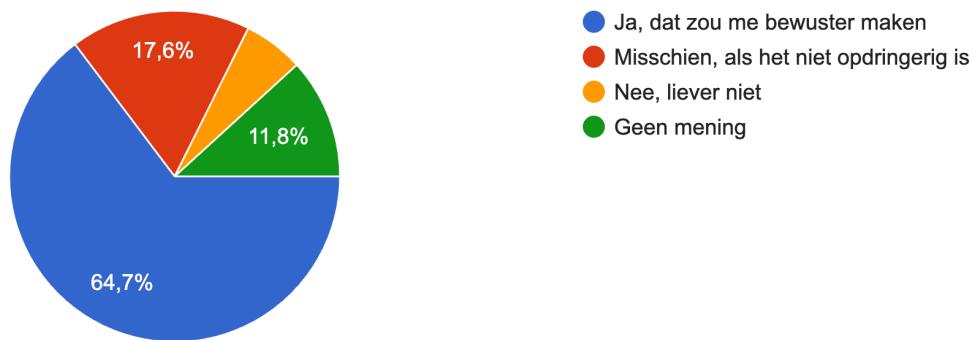


Figuur 2: Resultaat uit de enquête op de vraag “Voor welke doeleinden gebruik je AI het meest?”

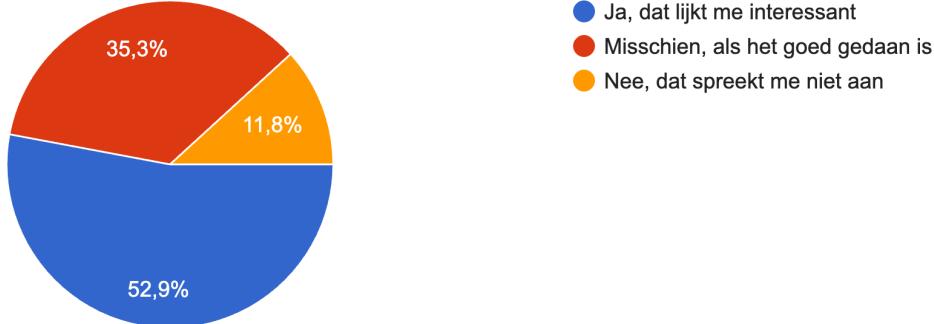
Voor je deze enquête invulde, wist je dat AI-systeem veel energie en water verbruiken?



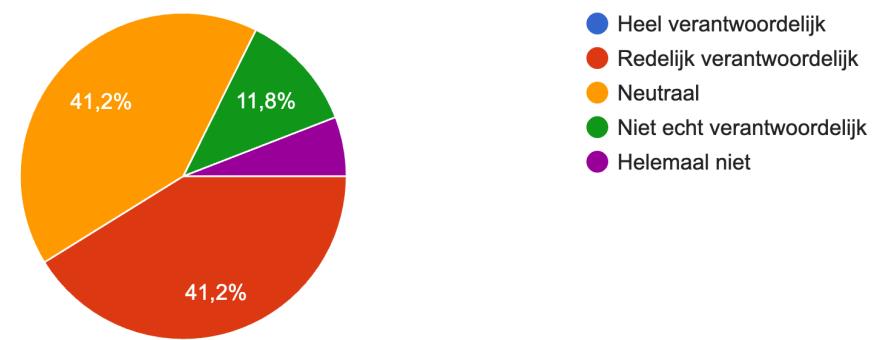
Zou je het waardevol vinden om realtime feedback te krijgen over de milieu-impact van jouw AI-gebruik?



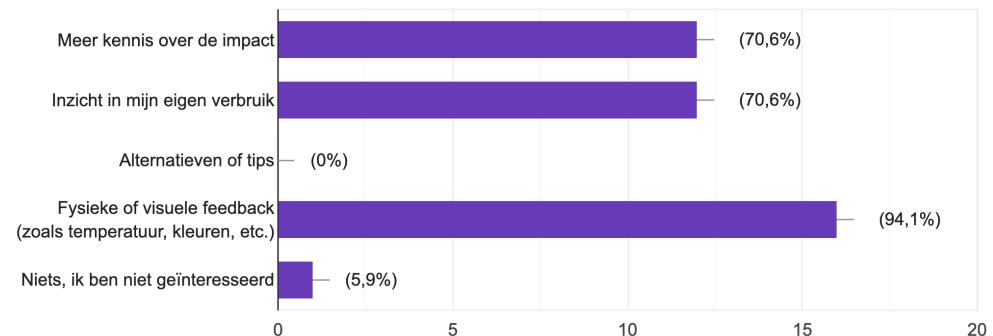
Sta je open voor een interactieve ervaring (bijv. een installatie of wearable die je laat voelen wat de impact van AI is?)



In hoeverre voel jij je verantwoordelijk voor de milieu-effecten van jouw technologiegebruik?



Wat zou jou helpen om bewuster om te gaan met AI-gebruik?



Figuur 3,4,5,6,7: Resultaten uit de enquête

EXPERIMENTEN

Aan de hand van de inzichten en interviews ben ik begonnen met het opzetten de eerste low fi experimenten en deze te testen onder de doelgroep. Deze manier van onderzoeken heb ik uitgewerkt tot de volgende 3 experimenten.

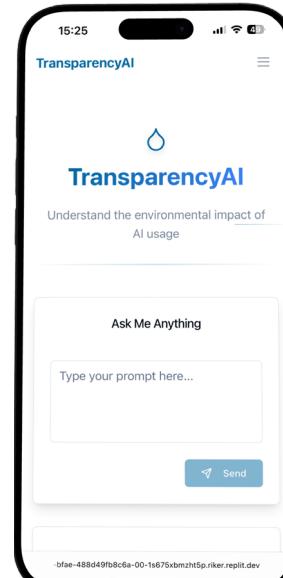
Experiment 1: Bewustwording van waterverbruik door AI

In dit eerste experiment is een AI-omgeving ontwikkeld met als doel gebruikers bewust te maken van de milieu-impact van hun AI-gebruik, in het bijzonder het waterverbruik dat gepaard gaat met het genereren van AI-antwoorden.

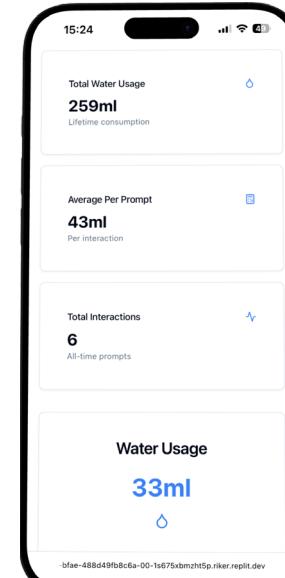
De ontwikkelde omgeving is zo ingericht dat elke ingevoerde prompt door de gebruiker een directe terugkoppeling geeft over het geschatte aantal milliliter water dat nodig is om het antwoord te genereren. Deze inschatting is gebaseerd op bestaande wetenschappelijke literatuur over het waterverbruik van datacenters bij het draaien van AI-modellen.

Wanneer het waterverbruik van een gebruiker boven een vooraf bepaalde drempel uitkomt, verschijnt er een melding. In deze melding wordt de gebruiker erop gewezen dat het wellicht mogelijk is om het antwoord op een andere manier te vinden—bijvoorbeeld door de prompt aan te passen of zelf verder te zoeken op het internet—en zo bij te dragen aan het verminderen van onnodig waterverbruik.

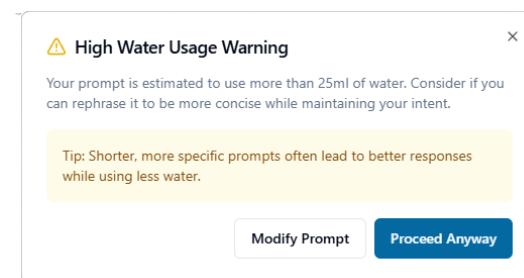
Ter aanvulling is er een dashboard geïntegreerd waarin gebruikers hun totale waterverbruik kunnen inzien gedurende hun sessies. Dit dashboard visualiseert het cumulatieve verbruik en is ontworpen om gebruikers te confronteren met de ecologische voetafdruk van hun interacties met AI. Het uiteindelijke doel is om bewustwording te creëren en gebruikers aan te zetten tot meer verantwoord AI-gebruik.



Figuur 8: Interface TransparencyAI



Figuur 9: TransparencyAI verbruikersgegevens



Figuur 10: De waarschuwingsmelding bij een hoog waterverbruik na een prompt.

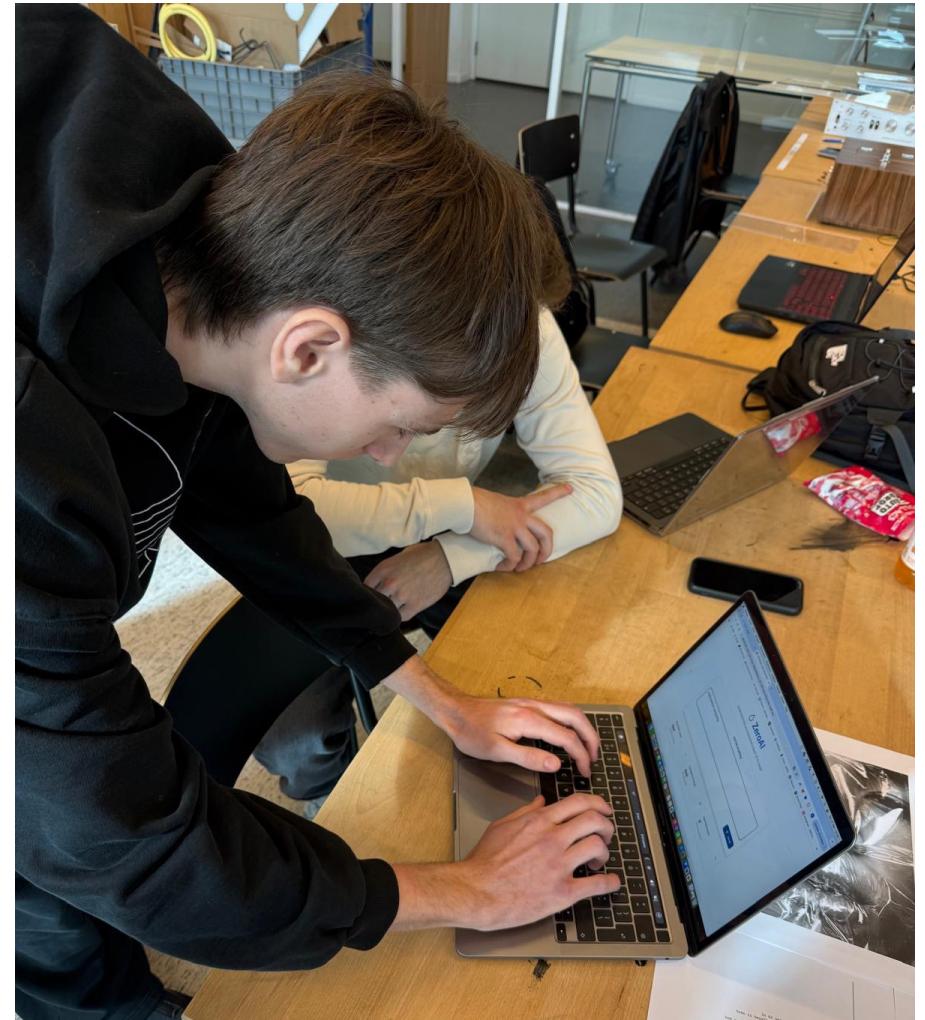
Technische aspecten

De applicatie is opgebouwd als een webgebaseerde omgeving, bestaande uit een moderne frontend en een robuuste backend. De frontend is ontwikkeld met React in combinatie met TypeScript voor een dynamische en schaalbare gebruikersinterface, en gestyled met Tailwind CSS om consistente en toegankelijke vormgeving te garanderen. De backend is gebouwd met Node.js en Express.js (eveneens in TypeScript), waarin de serverlogica, API-functionaliteiten en de integratie met de OpenAI API zijn ondergebracht.

De kernfunctionaliteit van het systeem draait om het analyseren en visualiseren van het waterverbruik dat gepaard gaat met AI-gebruik. De belangrijkste features zijn onder meer:

- Prompt input en analyse: gebruikers kunnen prompts invoeren en direct het bijbehorende waterverbruik zien.
- Waterverbruik vergelijkingen: biedt context door het verbruik te vergelijken met alledaagse watergebruiken.
- Educatieve content: aanvullende informatie helpt gebruikers de impact van AI op het milieu beter te begrijpen.
- Gebruikersniveausysteem: beloont efficiënter en bewuster gebruik met een opbouwend feedbackmechanisme.
- Leaderboard-functionaliteit: laat gebruikers hun prestaties vergelijken, wat bewustwording stimuleert via onderlinge vergelijking.

Deze technische opzet zorgt niet alleen voor een gebruiksvriendelijke ervaring, maar maakt het ook mogelijk om op een schaalbare manier inzicht te geven in de verborgen ecologische kosten van AI-interactie.



Figuur 11: Gebruiker test en interacteert met TransparencyAI

Resultaten/Terugkoppeling op het experiment: Bewustwording van waterverbruik door AI

Tijdens het testen van de AI-omgeving, die gebruikers bewust moest maken van het waterverbruik bij AI-interacties, zijn er meerdere inzichten verzameld. Deze feedback is te verdelen in positieve punten en onderdelen waar verbetering mogelijk is.

Wat goed werkte volgens de testgroep

De testgebruikers vonden het doel van de applicatie erg vernieuwend en relevant. De koppeling tussen het gebruik van AI en de ecologische gevolgen – specifiek het waterverbruik – werd als interessant en belangrijk ervaren. Vooral het direct tonen van het geschatte waterverbruik per prompt hielp om de impact zichtbaar en begrijpelijk te maken.

Ook de vormgeving en de gebruikerservaring kregen positieve reacties. Dankzij de moderne opzet met React, TypeScript en Tailwind CSS was de applicatie overzichtelijk en prettig in gebruik. De extra informatie en vergelijkingen met alledaagse situaties maakten het makkelijker om de cijfers in context te plaatsen.

De spelementen, zoals het beloningssysteem met niveaus en de leaderboard, motiveerden gebruikers om bewustere keuzes te maken. Dit zorgde voor extra betrokkenheid bij het onderwerp.

Wat beter kan volgens de testgroep

Een aantal testers vroeg zich af hoe betrouwbaar de getoonde waterverbruikscijfers precies waren. Meer uitleg over hoe deze berekend worden en waar de data vandaan komt, zou het vertrouwen in de tool vergroten.

Verder werd de melding bij een te hoog verbruik soms als iets te streng of direct ervaren. Een vriendelijker of minder confronterende manier van waarschuwen zou mogelijk beter werken.

Gebruikers gaven ook aan dat ze graag meer controle zouden willen, bijvoorbeeld door zelf in te stellen wanneer meldingen verschijnen of hoe streng de waarschuwingen zijn. Daarnaast bleek dat de applicatie op mobiele apparaten nog niet altijd even goed functioneerde.

Tot slot wilden sommige gebruikers graag meer informatie over de bredere milieu-impact van AI, zoals energiegebruik of CO₂-uitstoot, naast het waterverbruik.

Conclusie

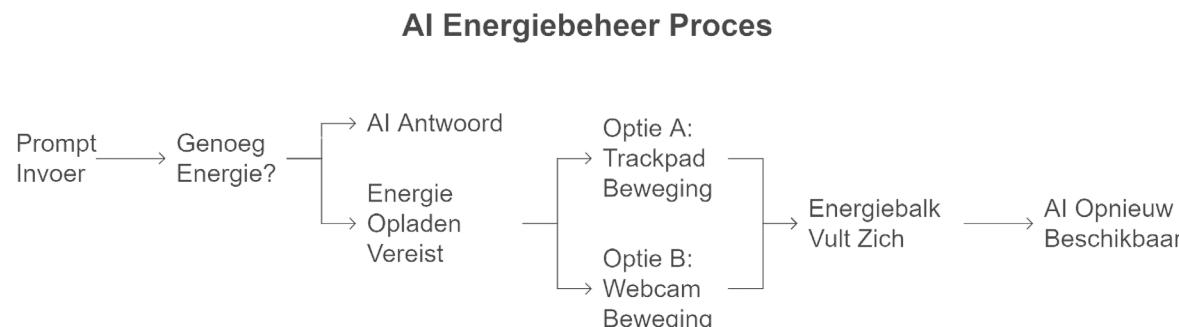
De testgroep vond het experiment waardevol en leerzaam. Het is op een duidelijke en creatieve manier gelukt om aandacht te vragen voor een minder bekend milieuprobleem. Tegelijkertijd zijn er ook enkele verbeterpunten, vooral op het gebied van datatransparantie, gebruiksgemak op mobiele apparaten en uitbreidingen van de inhoud. Deze feedback biedt een goede basis voor verdere ontwikkeling van de applicatie.

Experiment 2: Powered By You

In dit tweede experiment is een aangepaste AI-omgeving ontwikkeld waarin het gebruik van het AI-model afhankelijk is van een visuele energiebalk. De energiebalk representeert de beschikbare energie die nodig is om de AI-functies te activeren. Wanneer de energiebalk vol is, kunnen gebruikers prompts invoeren en ontvangen zij hierop antwoorden van het AI-model. Zodra de energiebalk leeg is, wordt het systeem geblokkeerd en kunnen gebruikers geen nieuwe prompts meer verwerken. Deze opzet is bedoeld om gebruikers bewust te maken van het feit dat de energie waarop AI-modellen draaien niet oneindig beschikbaar is.

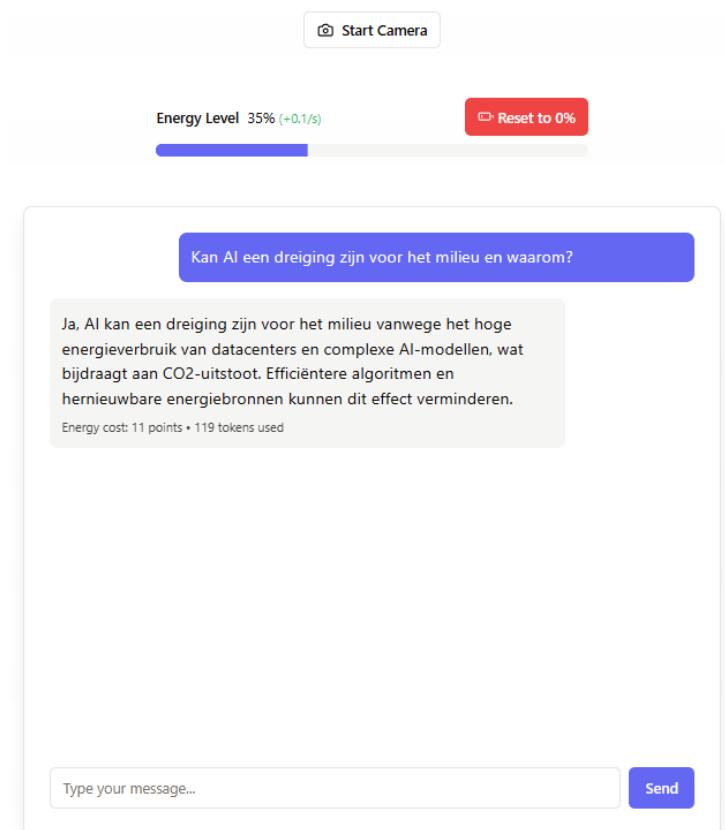
Om het AI-systeem opnieuw 'op te laden', zijn er twee methoden geïmplementeerd, die beide fysieke inspanning vereisen van de gebruiker:

1. Trackpadbeweging: door langdurig of intensief met de vinger over het trackpad van de laptop te bewegen, wordt geleidelijk energie teruggewonnen.
2. Camera-gebaseerde beweging: gebruikers kunnen hun webcam inschakelen en fysieke bewegingen uitvoeren, zoals zwaaien, dansen of springen. De applicatie detecteert beweging via de camera en zet deze om in hernieuwde energie voor het AI-systeem.



Figuur 14: Schematische weergave van het energiebeheerproces in de applicatie.

Powered By You

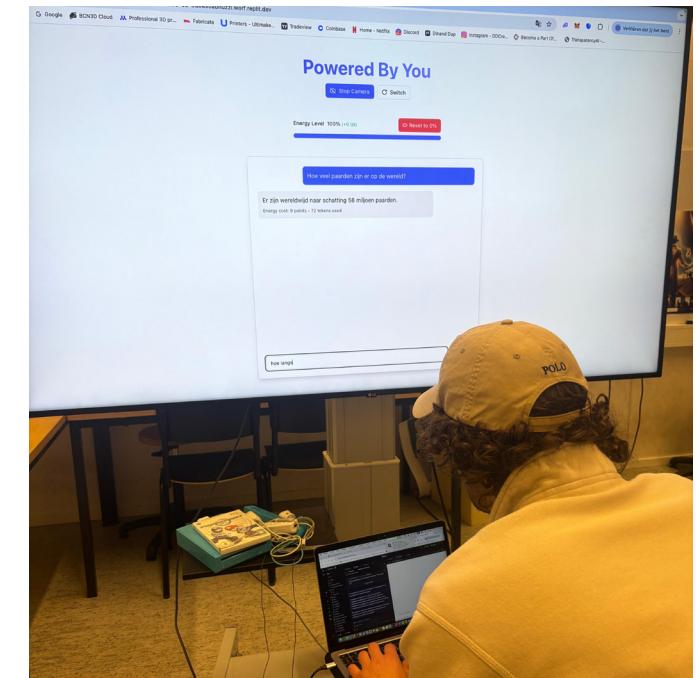
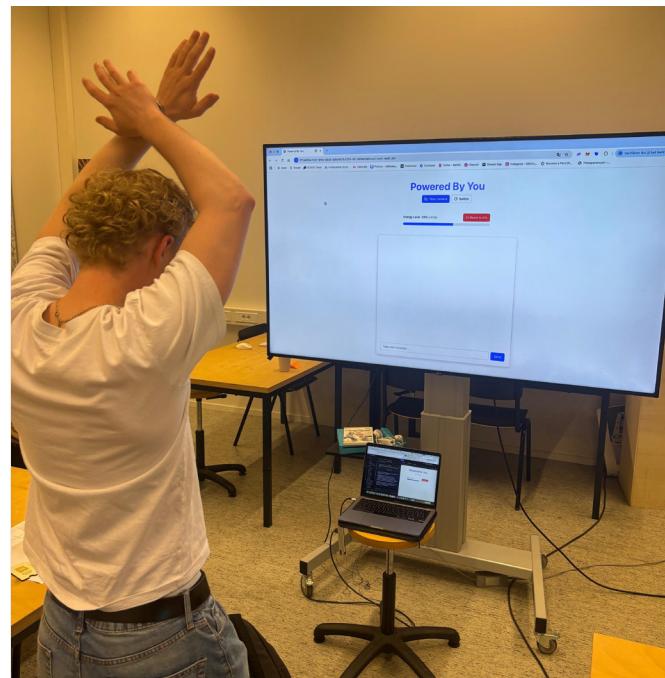


Figuur 12: Powered By You interface



Figuur 13: De foutmelding die verschijnt wanneer de gebruiker onvoldoende energie heeft.

Door deze actieve deelname vereist te maken, ontstaat er opzettelijk een frictie in het gebruik van AI. Hiermee wordt de gebruiker geconfronteerd met de fysieke realiteit achter digitale processen: toegang tot AI is niet vanzelfsprekend of eindeloos. Het experiment streeft ernaar om gebruikers te laten ervaren dat zij zelf energie moeten investeren om AI-technologie te kunnen benutten.



Figuur 15,16,17: Gebruikers interacteren fysiek met de "Powered By You" applicatie om energie op te wekken via de webcam.

Technische aspecten

De applicatie is gebouwd als een interactieve webomgeving, waarbij de frontend opnieuw is ontwikkeld met React en TypeScript, en vormgegeven met Tailwind CSS voor een intuïtieve en responsieve gebruikersinterface. De backend is gerealiseerd met Node.js en Express.js (TypeScript), met integraties voor het AI-model via de OpenAI API.

De technische architectuur ondersteunt de volgende kernfunctionaliteiten:

- Energiebalk-interface: dynamische weergave van het beschikbare energieniveau voor AI-gebruik.
- Prompt input afhankelijk van energie: AI-reacties zijn alleen beschikbaar bij voldoende energie.
- Trackpad input tracking: registreren van continue beweging op het trackpad om energie op te wekken.
- Webcam-gebaseerde bewegingsherkenning: detecteert fysieke activiteit van de gebruiker en zet dit om in energie.
- Feedback en interactie: visuele en tekstuele feedback tijdens het opladen en gebruik van het systeem.

Deze opzet combineert technologische innovatie met gedragspsychologie, om op een speelse maar confronterende manier bewustwording te creëren over de fysieke en energetische impact van AI-gebruik.

Resultaten/Terugkoppeling op het experiment: Powered By You

In dit tweede experiment werd een interactieve AI-omgeving ontworpen waarbij gebruikers alleen toegang kregen tot AI-functionaliteit wanneer zij zelf energie genereerden. Dit kon via trackpadbewegingen of fysieke bewegingen die door een webcam werden geregistreerd. Het doel hiervan was om gebruikers bewust te maken van het feit dat AI-technologie afhankelijk is van echte, fysieke energie en dus niet onbeperkt of vanzelfsprekend is.

Wat goed werkte volgens de testgroep

De testgroep vond het onderliggende idee van dit experiment erg origineel en prikkelend. Het principe dat je eerst zelf moeite moet doen om AI te kunnen gebruiken – door letterlijk energie op te wekken – werd als effectief ervaren om bewustwording te creëren over het energieverbruik van technologie.

Ook de technische uitvoering van de energiebalk en de responsieve gebruikersinterface kreeg positieve reacties. Door het gebruik van herkenbare technieken zoals React, TypeScript en Tailwind CSS oogde de omgeving modern en gebruiksvriendelijk. De visuele feedback tijdens het 'opladen' maakte het systeem begrijpelijk en gaf een speels, bijna game-achtig karakter aan de ervaring.

Veel gebruikers waardeerden de creativiteit van de energierugwinning via het trackpad en de webcam. Vooral de camerafunctie werd als verrassend en leuk ervaren. Het zette gebruikers letterlijk in beweging en zorgde voor een fysieke betrokkenheid die je normaal niet associeert met AI-gebruik.

Wat beter kan volgens de testgroep

Hoewel de insteek van het experiment duidelijk was, vonden sommige gebruikers de vereiste fysieke inspanning op momenten wat overdreven of zelfs frustrerend. Met name bij langer gebruik werd het als vermoedend ervaren om telkens opnieuw energie te moeten opwekken, zeker als het systeem snel weer 'leeg' raakte.

Daarnaast werkte de bewegingsdetectie via de webcam niet altijd even nauwkeurig. Gebruikers merkten op dat niet alle bewegingen goed werden herkend, wat soms leidde tot verwarring of herhaalde pogingen om energie op te wekken. Dit kan het gebruiksplezier verminderen.

Verder werd voorgesteld om het systeem iets flexibeler te maken, bijvoorbeeld door gebruikers zelf de gevoeligheid of snelheid van energieverbruik te laten instellen. Een 'pauzestand' of tijdelijke uitschakeling van de energiebeperking zou het experiment toegankelijker maken voor een bredere groep gebruikers.

Tot slot vroegen sommige testers zich af hoe deze simulatie zich verhoudt tot het daadwerkelijke energieverbruik van AI in de praktijk. Hoewel het speelse karakter van het experiment werkte, zou aanvullende informatie of context helpen om het verband met de echte ecologische impact van AI duidelijker te maken.

Conclusie

Het experiment "Powered By You" werd als origineel, uitdagend en inspirerend ervaren. Het slaagde erin om gebruikers stil te laten staan bij het idee dat AI afhankelijk is van fysieke middelen. Tegelijkertijd zijn er verbeterpunten op het gebied van gebruiksgemak, nauwkeurigheid van bewegingsdetectie en het bieden van context. Door deze feedback mee te nemen, kan het concept verder worden aangescherpt en mogelijk breder ingezet worden als educatief of bewustmakend middel.

Experiment 3: Mens als model

In dit derde experiment is een diepgaande analyse uitgevoerd van het menselijke energiemodel, met als uitgangspunt het eigen lichaam van de onderzoeker. Gedurende een periode van 24 uur is gedetailleerd in kaart gebracht op welke manieren het lichaam energie verbruikt en opneemt. Dit onderzoek strekte zich uit over zowel actieve als passieve momenten van de dag, inclusief slaap, maaltijden, beweging, mentale inspanning en rust.

Alle waarnemingen zijn vastgelegd in een zorgvuldig samengestelde dataset/tabel waarin de energie-inname (voeding, slaap, rustmomenten) en het energieverbruik (lichamelijke activiteit, hersenactiviteit, spijsvertering, warmteproductie, etc.) op micro-niveau zijn gemeten. De tabel omvat een nauwkeurige tijdlijn van energieprocessen die per activiteitstype, tijdsblok en fysiologisch systeem zijn uitgesplitst.

Het doel van dit experiment is om deze biologische gegevens te vertalen naar een functioneel AI-model dat zich gedraagt volgens een "menselijk energiemodel". In plaats van AI-modellen die voortdurend op volle kracht opereren op een abstracte of eindeloze energietoeroer, streeft dit experiment ernaar om een AI-systeem te ontwikkelen dat werkt volgens dezelfde principes als een menselijk lichaam: beperkt, cyclisch, afhankelijk van energie-inname, herstelmomenten, en wisselende intensiteit.

Hiermee wordt onderzocht hoe menselijke beperkingen en ritmes geïntegreerd kunnen worden in de werking van kunstmatige intelligentie, met als doel een meer duurzame, begrijpelijke en empathische vorm van AI te ontwerpen.

De eerste stap in dit experiment bestond uit het creëren van een database waarin het energieverbruik van het menselijk lichaam per categorie werd vastgelegd. Voor de dataverzameling is gebruikgemaakt van handmatige logging in combinatie met schattingen op basis van bestaande wetenschappelijke gegevens over metabolisme, calorisch verbruik per activiteit, slaapfisiologie en cognitieve belasting. Deze data zijn vertaald naar een gestructureerde tabel met de volgende hoofdcategorieën (figuur 11):

Energie-inname:

- Voedsel (type, tijdstip, hoeveelheid, geschatte energie inhoud)
- Slaap (duur, kwaliteit, herstellende waarde)
- Rustmomenten (zoals meditatie of passieve ontspanning)

Energieverbruik:

- Fysieke activiteit (lopen, fietsen, huishoudelijke taken, sport)
- Mentale activiteit (concentratie, probleemoplossing, creatieve processen)
- Automatische lichaamsfuncties (hartslag, ademhaling, spijsvertering, temperatuurregulatie)
- Emotionele responsen en stressbelasting

De volledige dataset vormt het basismodel voor een nog te ontwikkelen AI-systeem dat op een vergelijkbare manier energie moet managen. Het AI-model zal enkel taken uitvoeren wanneer het voldoende "energie" heeft, en heeft mogelijk ook momenten van 'rust', 'slaap' of herstelfases nodig. De architectuur wordt daarmee een hybride tussen menselijke fisiologie en machine learning-logica.

24 uurs tijdlijn van activiteiten en energieverbruik

Starttijd	Eindtijd	Activiteit	Duur (min)	Calorieverbruik (kcal)
08:30	09:00	Zitten	30	48.5
09:00	09:10	Lopen naar bushalte	10	47.2
09:10	10:30	Zitten	80	129.3
10:30	10:45	Rustmoment	15	22.4
10:45	10:48	Traplopen	3	29.8
10:48	14:00	Zitten	192	310.3
14:00	14:30	Rustmoment	30	44.8
14:30	14:50	Lopen naar bushalte	20	94.5
14:50	15:30	Zitten	40	64.7
15:30	16:55	Intensief studeren	85	158.5
16:55	17:00	Rustmoment	5	7.5
17:00	18:00	Zitten	60	97.0
18:00	18:30	Eten + sociale interactie	30	55.9
18:30	19:30	Serie kijken (rust)	60	89.5
19:30	19:45	Aankleden voor voetbal	15	24.2
19:45	19:50	Fietsen naar voetbal	5	28.0
20:00	21:30	Voetbaltraining	90	895.2
21:30	22:45	Zitten	75	121.2
22:45	08:30	Slapen	585	691.0

24 uurs tijdlijn van energie-inname

Tijd	Item	Geschatte kcal
10:30	1 suikerwafel	180
14:00	2 bruine boterhammen met kaas	420
18:00	Rijst, snijbonen en schnitzel	600

Figuur 18: Voorbeeld van de 24-uurs dataset, met links het energieverbruik en rechts de energie-inname.

Totaal verbruik over de dag: ± 2.960 kcal
 Totaal energie-inname via voeding: 1.200 kcal
 Netto energiebalans: -1.760 kcal

Mensenlijke energie omzetten in elektrische energie

Nu de dataset compleet is, kunnen de verzamelde gegevens worden omgezet naar energie-eenheden die passen bij het functioneren van een AI-model. Door het menselijke energieverbruik op deze manier te vertalen, ontstaat er een vergelijkmiddel waarmee zichtbaar wordt hoeveel energie een AI-model nodig heeft in verhouding tot de energie die een mens verbruikt. Dit maakt het mogelijk om een beter begrip te krijgen van de impact en het verbruik van kunstmatige intelligentie, op basis van iets wat we allemaal kennen: ons eigen lichaam.

Stap 1: Energie omzetten
 1 kilocalorie (kcal) = 1.163 Wh (wattuur)
 (= 4184 joule ÷ 3600 s)

Formule:

Energie in Wh=Energie in kcal×1.163

Stap 2: Vergelijken met AI-verbruik
 Gemiddeld AI-verbruik hangt af van het type model. Hier enkele richtgetallen

AI-systeem	Verbruik per 1000 tokens (Wh)
GPT-3 / ChatGPT-3.5	± 0.35 Wh
GPT-4	± 2.9 Wh
Simpele ML-model (lokaal)	± 0.05 Wh

Figuur 19: Vergelijkingstabel van het energieverbruik van verschillende AI-modellen per 1000 tokens.

Voorbeeld:

Mijn totale energieverbruik op deze dag = 2.960 kcal

Omgezet naar elektriciteit =

$$2960 \times 1.163 = \approx 3442 \text{ Wh} = 3.44 \text{ kWh}$$

Stap 3: Hoeveel prompts zou de AI kunnen draaien met mijn energie?

Stel je gebruikt GPT-4, dat 2,9 Wh per prompt verbruikt:

$$3442 \text{ Wh} \div 2,9 \text{ Wh} = \approx 1186,89 = 1187 \text{ prompts.}$$

Hoewel uit de berekeningen blijkt dat mijn totale dagelijkse energieverbruik (2960 kcal, omgerekend circa 3442 Wh) voldoende zou zijn om bijvoorbeeld 1187 prompts aan GPT-4 te versturen, is dit slechts een theoretisch maximum. In deze berekening is namelijk uitsluitend uitgegaan van het totale energieverbruik over de dag, zonder rekening te houden met de daadwerkelijke energie-inname.

Wanneer we de energie-inname (in de vorm van voeding) wél meenemen in de analyse, wordt duidelijk dat het lichaam op deze dag onvoldoende energie heeft binnengekregen om het volledige energieverbruik te compenseren. Dit betekent dat het netto beschikbare energieniveau lager ligt dan het theoretisch berekende maximum.

In de context van het experiment waarin het menselijk energiemodel wordt vertaald naar een AI-model, is dit onderscheid van groot belang. Een AI-systeem dat gebaseerd is op menselijke energie zou, net als het menselijk lichaam, slechts optimaal kunnen functioneren zolang er voldoende energie beschikbaar is om te verbranden. Zodra de energie-inname onvoldoende is ten opzichte van het verbruik, zou dit AI-model in capaciteit moeten afnemen of tijdelijk stoppen met functioneren.

Het doel is dan ook om het gedrag van dit AI-model dynamisch aan te passen op basis van een tijdlijn die zowel de energie-inname als het verbruik in kaart brengt. Op die manier ontstaat er een systeem dat, net als een menselijk lichaam, energiebeperkingen erkent en functioneert binnen realistische fysiologische grenzen.

Resultaten/Terugkoppeling op het experiment: Mens als model

Het derde experiment richtte zich op het ontwikkelen van een AI-systeem gebaseerd op de energieprincipes van het menselijke lichaam. De feedback uit de testgroep bevat zowel positieve bevindingen als enkele suggesties voor verbetering.

Positieve bevindingen:

De gedetailleerde dataset en tijdlijn, die het energieverbruik en de energie-inname over de dag vastlegden, werden als zeer waardevol beschouwd. Dit gaf een helder inzicht in de energiebalans van het lichaam, zowel tijdens actieve als passieve momenten. Het gebruik van wetenschappelijke gegevens om metabolisme en calorisch verbruik te schatten werd gewaardeerd, wat de betrouwbaarheid van de data versterkte.

De integratie van menselijke fysiologie in het ontwerp van AI werd als innovatief gezien, met de mogelijkheid om duurzamere en empathische AI-systemen te ontwikkelen die rekening houden met menselijke energiebeperkingen en ritmes.

Verbeterpunten:

De testgroep suggereerde een verfijning van de meetmethoden, vooral voor hersenactiviteit en emotionele reacties, aangezien handmatige logging en schattingen onnauwkeurigheden kunnen introduceren. Daarnaast werd het experiment, dat slechts op één proefpersoon is gebaseerd, als te beperkt gezien. Het uitbreiden naar meerdere proefpersonen zou de resultaten betrouwbaarder maken.

De vertaling van biologische principes naar een werkend AI-systeem werd als technisch uitdagend beoordeeld. Er werd geopperd om meer gedetailleerde technische specificaties op te nemen voor de implementatie van het AI-model. Tenslotte werd de meetperiode van 24 uur als te kort beschouwd, met de suggestie om het experiment over een langere periode uit te voeren voor consistenter en representatiever data.

VERDIEPING & INSPIRATIE

Na de uitgevoerde low fi experimenten is het tijd om vorm te geven aan het eindontwerp. Hiervoor zijn de belangrijke inzichten vanuit het deskresearch, fieldresearch en de experimenten opgenomen. Verder heb ik mezelf verdiept in de theorie van embodied cognition om beter begrip te krijgen in embodied experiences.

Hoe gebruik ik embodied cognition om de gebruiker lichamelijk te laten "leren"?

In dit project staat het concept van 'embodied experience' centraal, wat nauw verbonden is met de theorie van Embodied Cognition (belichaamde cognitie). Deze theorie stelt dat ons denken en begrijpen niet alleen in onze hersenen plaatsvindt, maar diep verweven is met ons lichaam, onze zintuigen en onze interacties met de omgeving. Met andere woorden: we denken niet alleen met ons brein, maar ook met ons lichaam.

Wat is Embodied Cognition?

Traditioneel werd gedacht dat cognitie (denken, waarnemen, begrijpen) een puur mentaal proces is, los van het lichaam. Embodied Cognition daagt dit uit door te stellen dat:

1. Het lichaam essentieel is voor cognitie: Onze lichamelijke ervaringen, bewegingen en zintuiglijke waarnemingen vormen de basis van hoe we de wereld begrijpen en hoe we denken. Abstracte concepten, zoals 'tijd' of 'macht', worden vaak begrepen via lichamelijke metaforen (bijvoorbeeld 'vooruitkijken' voor de toekomst, of 'opkijken' naar iemand met macht).
2. Cognitie is interactief: Denken is geen geïsoleerd proces, maar ontstaat uit de voortdurende interactie tussen ons lichaam en de omgeving. We leren en begrijpen door te doen en te ervaren.
3. Zintuiglijke ervaringen zijn cruciaal: Informatie die we via onze zintuigen (zien, horen, voelen, ruiken, proeven) ontvangen, is niet zomaar input voor het brein, maar een integraal onderdeel van het cognitieve proces.

In de context van mijn project betekent dit dat ik abstracte concepten, zoals de milieu-impact van AI, tastbaar en voelbaar wil maken. Door mensen fysiek te laten ervaren wat de impact is, wordt het begrip dieper en persoonlijker dan wanneer het alleen via cijfers of tekst wordt gepresenteerd. Dit is verder te onderbouwen aan de hand van somaesthetics waardevolle inzichten. Somaesthetics, een term geïntroduceerd door filosoof Richard Shusterman, richt zich op de systematische, kritische en constructieve studie van de menselijke ervaring van het lichaam (soma) als een bron van kennis, perceptie en expressie. Het gaat verder dan esthetiek en omvat de bewuste ontwikkeling en verbetering van onze lichamelijke sensaties en bewegingen.

Binnen somaesthetics is het principe van Somatic Awareness relevant voor mijn project. Dit principe benadrukt het belang van het vergroten van de aandacht voor en het bewustzijn van onze eigen lichamelijke sensaties. Vaak zijn we ons niet bewust van subtiele signalen die ons lichaam ons geeft, of negeren we deze. Somatic Awareness streeft ernaar deze interne dialoog te versterken, waardoor we beter kunnen interpreteren wat we voelen en hoe dit zich verhoudt tot onze omgeving of acties. Het is het proces van het 'leren voelen' en het 'voelen om te leren'.

In dit project heb ik er daarom voor gekozen om gebruik te maken van de temperatuurregeling van ons eigen lichaam. Dit biologische proces ga ik inzetten om de onzichtbare ‘belasting’ van AI-gebruik voelbaar en meetbaar te maken.

Homeostase: Het Lichaam als Feedback-systeem

Een belangrijk concept dat hier perfect op aansluit, is homeostase. Simpel gezegd is homeostase het vermogen van ons lichaam om het interne evenwicht te bewaren, zelfs als de omgeving verandert. Een klassiek voorbeeld hiervan is onze lichaamstemperatuur. Ons lichaam wil altijd rond de 37°C blijven. Word je te warm? Dan ga je zweten om af te koelen. Krijg je het koud? Dan ga je rillen om warmte op te wekken.

Dit constante proces van voelen, vergelijken en corrigeren is een perfecte, natuurlijke feedbackloop. Mijn project gebruikt dit principe als inspiratie: net zoals je lichaam reageert op een te hoge temperatuur, kan een interactieve ervaring reageren op ‘te hoog’ AI-gebruik. Zo wordt de abstracte impact van AI direct gekoppeld aan een fysieke sensatie die iedereen herkent: warmte.

Experts

Twee invloedrijke denkers die veel onderzoek doen naar de milieugevolgen van kunstmatige intelligentie zijn Kate Crawford en Sasha Luccioni. Zij benaderen dit onderwerp vanuit verschillende invalshoeken, maar delen de zorg over de ecologische en maatschappelijke impact van AI-technologie.

Kate Crawford – AI als een extractief systeem

Kate Crawford is medeoprichter van het AI Now Institute en auteur van het boek *Atlas of AI* (2021). In haar werk laat zij zien dat AI geen abstracte, “schone” technologie is, maar juist sterk afhankelijk is van fysieke middelen. Crawford beschrijft AI als een extractief systeem: het onttrekt natuurlijke hulpbronnen (zoals metalen voor chips), energie (voor datacenters) en arbeid (bijvoorbeeld van mensen die data labelen voor trainingsdoeleinden).

Ze benadrukt dat de groei van generatieve AI – zoals taalmodellen en beeldgeneratie – leidt tot een nog grotere belasting van het milieu. Volgens Crawford moeten we AI dan ook niet alleen beoordelen op prestaties, maar ook op de maatschappelijke, ecologische en politieke kosten.

Sasha Luccioni – De ecologische voetafdruk van AI meten

Sasha Luccioni is onderzoeker bij Hugging Face en medeoprichter van Climate Change AI. Haar werk richt zich op het meetbaar maken van de milieu-impact van AI-systeem, bijvoorbeeld het energieverbruik en de CO₂-uitstoot bij het trainen van grote modellen. Ze ontwikkelt hulpmiddelen om deze impact inzichtelijk te maken en pleit voor meer transparantie binnen de AI-sector.

Luccioni wijst ook op het zogenaamde Jevons-effect: wanneer AI-systeem efficiënter worden, gaan ze juist vaker en grootschaliger ingezet worden, wat alsnog leidt tot méér energieverbruik. Daarom pleit zij voor beleidsmaatregelen en best practices die gericht zijn op duurzaamheid en verantwoordelijkheid binnen AI-ontwikkeling.

Overeenkomsten en verschil in benadering

Beide onderzoekers benadrukken het belang van een meer bewuste omgang met kunstmatige intelligentie. Crawford bekijkt AI vooral als een systeem dat diep verweven is met sociale ongelijkheid en milieuschade. Luccioni richt zich juist op het praktisch meetbaar maken en terugdringen van de milieu-impact van AI. Hun werk vult elkaar goed aan: waar Crawford kritische vragen stelt over de fundamentele van het AI-ecosysteem, biedt Luccioni concrete tools om de technologie duurzamer te maken.

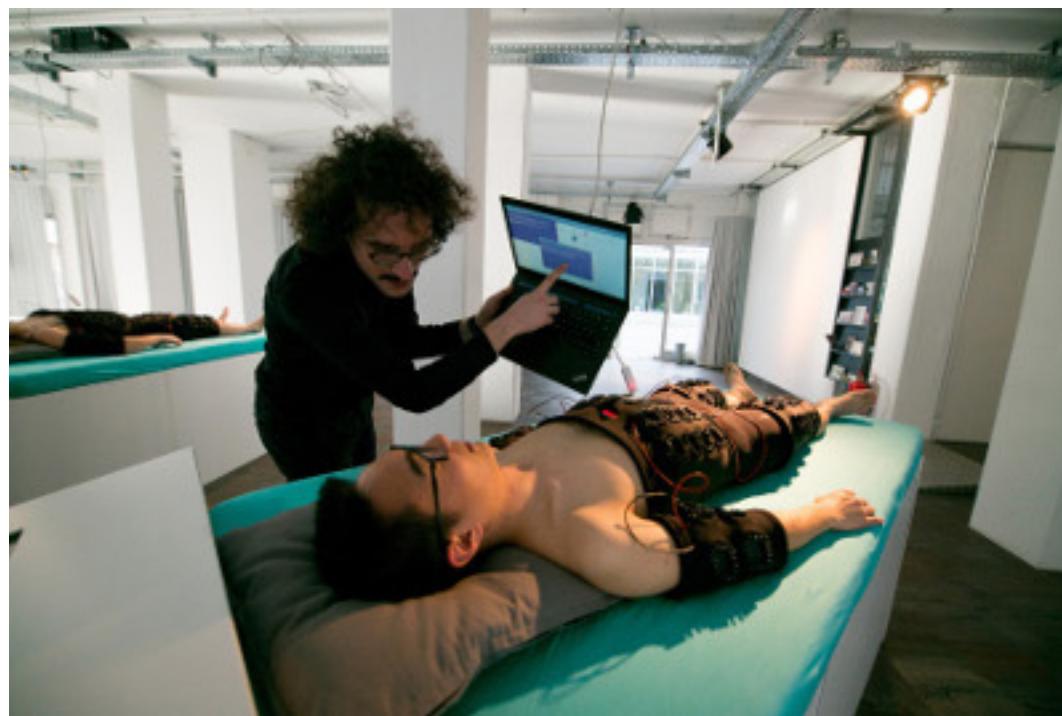
De inzichten van Kate Crawford en Sasha Luccioni sluiten aan bij mijn eigen onderzoek, waarin ik een prototype ontwikkel dat een belichaamde ervaring wil creëren rond de milieu-impact van AI. Waar Crawford de diepere, vaak onzichtbare structuren blootlegt achter de fysieke en politieke infrastructuur van AI, en Luccioni de ecologische voetafdruk concreet en meetbaar maakt, biedt hun werk een belangrijke context en onderbouwing voor mijn project. Mijn prototype wil deze abstracte en technologische realiteit tastbaar maken door de kwantumbiologische principes van temperatuurregeling in het menselijk lichaam te verbinden met de energieverbruik en milieu-impact van AI-systeem. Op deze manier wil ik niet alleen informeren, maar ook letterlijk voelbaar maken wat vaak onzichtbaar blijft: de directe relatie tussen kunstmatige intelligentie, ons lichaam en de planeet.

Inspiratie

Voor mijn project heb ik veel inspiratie gehaald uit het werk van het 'Institute of Human Obsolescence'. Zij gebruiken het menselijk lichaam op een unieke manier om complexe onderwerpen, zoals de economie, tastbaar te maken. Deze aanpak, waarbij een onzichtbaar proces wordt gekoppeld aan een fysieke ervaring, vond ik erg krachtig.

In mijn eigen prototype wil ik een vergelijkbaar idee toepassen. Ik wil de milieu-impact van AI, die normaal gesproken onzichtbaar is, verbinden met een lichamelijke sensatie: warmte. Het idee is dat je de 'kosten' van een KI-opdracht direct op je eigen lichaam voelt.

Door dit te doen, hoop ik mensen bewuster te maken van de verborgen energie die technologie verbruikt. Het gaat er niet alleen om dat je weet dat het energie kost, maar dat je het ook echt voelt. Dit sluit aan bij mijn hoofddoel: onderzoeken hoe we de waarde van de mens en ons handelen opnieuw kunnen bekijken in een wereld die steeds meer door technologie wordt gedomineerd.



Figuur 20: Beeld van het project waarbij menselijke lichaamswarmte wordt gebruikt om energie op te wekken voor cryptomining.

Een ander voorbeeld dat mij inspireerde, is het kunstwerk "Spectral Subjects" van Rafael Lozano-Hemmer. Dit is een installatie die de onzichtbare warmte in een ruimte zichtbaar maakt. Met warmtecamera's meet het de temperatuur van alles en iedereen – van de muren tot de bezoekers zelf – en projecteert dit live op de wanden.

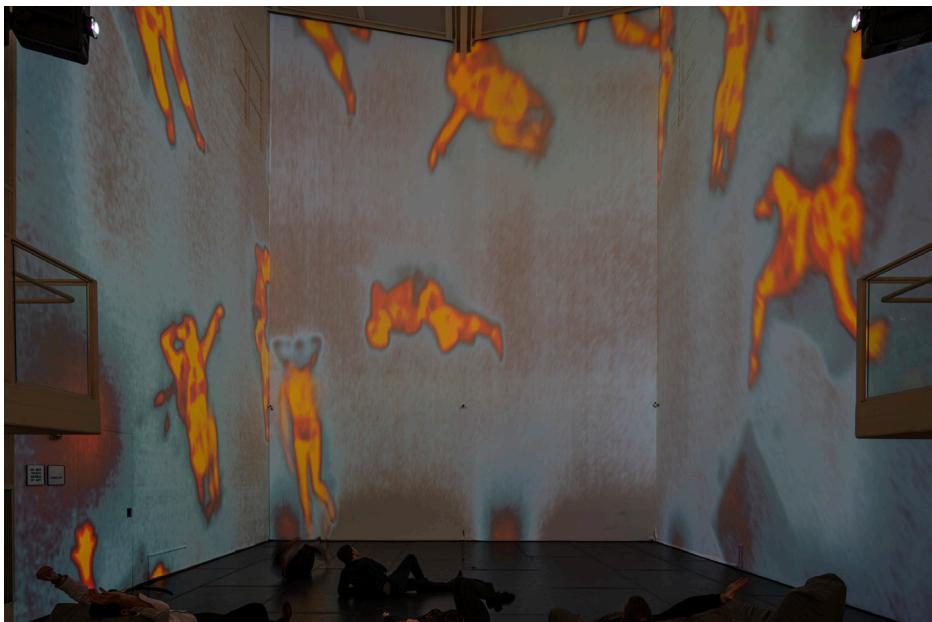
Wat het extra bijzonder maakt, is dat het werk beelden van het heden mengt met die van even geleden. Hierdoor zie je niet alleen je eigen warmte, maar ook de "thermische echo's" van mensen die er voor jou waren. Je wordt dus letterlijk omringd door deze warmtebeelden.

Voor mijn project is dit werk heel relevant, omdat het precies doet wat ik wil bereiken:

Het maakt onzichtbare energiestromen (warmte) tastbaar en visueel.

Het laat zien hoe de aanwezigheid van meerdere mensen samen één 'thermisch landschap' creëert. Dit is een mooie metafoor voor de gezamenlijke impact van ons KI-gebruik.

Het idee van 'thermische echo's' is een krachtige manier om te laten zien dat de impact van ons energieverbruik niet meteen verdwijnt, maar een spoor achterlaat. Dit project laat zien hoe je met thermische visualisatie een bewustzijn kunt creëren over onzichtbare processen en onze eigen aanwezigheid daarin. Het idee van de "thermische echo" is vooral een interessant perspectief om de langetermijneffecten van KI-gebruik te laten zien en voelen.



Figuur 21,22,23: Impressie van het kunstwerk "Spectral Subjects", dat thermische echo's van bezoekers visualiseert.

CONCEPTEN EMBODIED EXPERIENCE

Na de onderzoeksfase is het tijd om te gaan ontwerpen. Op basis van de inzichten uit mijn onderzoek, zoals; gebrek aan kennis, ... en mijn besluit om temperatuurregulatie als embodied cognition aspect te nemen ben ik begonnen met het opzetten van klein schalige concepten om te onderzoeken wat de effecten hiervan waren.

Concept 1: Heated Keyboard

Beschrijving:

Het "Heated Keyboard" experiment maakt de energie-impact van AI-gebruik direct voelbaar. Door verwarmingsfolie in een toetsenbord te integreren, voelt de gebruiker de toenemende energieconsumptie van AI-interacties als stijgende temperatuur tijdens het typen.

Doel:

- Primair: De energetische kosten van AI-interacties lichamelijk ervaarbaar maken.
- Secundair: Onderzoeken hoe deze thermische feedback het bewustzijn en gedrag van gebruikers beïnvloedt.

Theoretische Onderbouwing:

Dit experiment is gebaseerd op het idee van belichaamde cognitie. Door de onzichtbare energieconsumptie van AI via warmte voelbaar te maken, wordt een abstract milieuprobleem vertaald naar een directe, fysieke ervaring. De geleidelijke opwarming van het toetsenbord dient als metafoor voor de opstapeling van digitale impact, wat leidt tot een dieper en intuïtiever begrip van het onderwerp.

Materialen:

Standaard toetsenbord, flexibele verwarmingsfolie, arduino, temperatuurregelaar, stroomvoorziening, thermometer, laptop met AI-model, logboek en stopwatch.

Technische Opstelling & Kalibratie:

De verwarmingsfolie wordt onder een toetsenbord-stencil geplaatst. De folie wordt gekalibreerd om een voorspelbare temperatuurstijging van ongeveer 1,0°C per minuut te garanderen tijdens actieve AI-interactie, met een maximale temperatuur van 75°C voor veiligheid. Voor het aansturen van de verwarmingsfolie via het AI model zijn ook enkele software matige bestanden nodig. Hiervoor kan je de structuur opdelen in 3 files.

1. Arduino file (hardware)
2. Bridge file (link tussen de hardware & server waarop de web app draait)
3. Web app

In de volgende figuren is te zien hoe deze bestanden zijn opgebouwd.

CONCEPTEN EMBODIED EXPERIENCE

```

sketch_aug22c | Arduino IDE 2.3.6
File Edit Sketch Tools Help
Select Board
sketch_aug22c.ino
1 #include <OneWire.h>
2 #include <DallasTemperature.h>
3
4 // Temperature sensor setup
5 #define ONE_WIRE_BUS 2
6 #define TEMPERATURE_PRECISION 12
7 OneWire oneWire(ONE_WIRE_BUS);
8 DallasTemperature sensors(&oneWire);
9
10 // Heating element control
11 #define HEATER_PIN 3
12 #define HEATER_PWM_MAX 255
13
14 // System variables
15 float currentTemperature = 0.0;
16 int heaterIntensity = 0; // 0-255 PWM value
17 unsigned long lastTempRead = 0;
18 unsigned long lastHeaterUpdate = 0;
19 const unsigned long TEMP_INTERVAL = 1000; // Read temperature every 1 second
20 const unsigned long HEATER_INTERVAL = 100; // Update heater every 100ms
21
22 // Safety limits
23 const float MAX_SAFE_TEMP = 35.0; // Maximum safe temperature in Celsius
24 const float OVERHEAT_TEMP = 40.0; // Emergency shutdown temperature
25 bool safetyShutdown = false;
26
27 void setup() {
28   Serial.begin(9600);
29
30   // Initialize temperature sensor
31   sensors.begin();
32   sensors.setResolution(TEMPERATURE_PRECISION);
33
34   // Initialize heater control pin
35   pinMode(HEATER_PIN, OUTPUT);
36   analogWrite(HEATER_PIN, 0); // Start with heater off
37
38   Serial.println("Thermal Equilibrium System Initialized");
39   Serial.println("Commands:");
40   Serial.println(" HEAT:<0-100> - Set heating intensity (0-100%)");
41   Serial.println(" TEMP - Request temperature reading");
42   Serial.println(" STATUS - Get system status");
43   Serial.println(" STOP - Emergency stop heating");
44   Serial.println();
45 }
46
47 void loop() {
48   unsigned long currentTime = millis();
49
50   // Read temperature periodically
51   if (currentTime - lastTempRead >= TEMP_INTERVAL) {
52     readTemperature();
53     checkSafetyLimits();
54     lastTempRead = currentTime;
55   }
56
57   // Update heater periodically
58   if (currentTime - lastHeaterUpdate >= HEATER_INTERVAL) {
59     updateHeater();
60     lastHeaterUpdate = currentTime;
61   }
62
63   // Process serial commands
64   if (Serial.available()) {
65     processCommand();
66   }
}

```



```

sketch_aug22b | Arduino IDE 2.3.6
File Edit Sketch Tools Help
Select Board
sketch_aug22b.ino
66   }
67
68   // Send periodic temperature updates
69   static unsigned long lastUpdate = 0;
70   if (currentTime - lastUpdate >= 3000) { // Every 3 seconds
71     sendTemperatureUpdate();
72     lastUpdate = currentTime;
73   }
74
75
76   void readTemperature() {
77     sensors.requestTemperatures();
78     float newTemp = sensors.getTempCByIndex(0);
79
80     if (newTemp != DEVICE_DISCONNECTED_C) {
81       currentTemperature = newTemp;
82     } else {
83       Serial.println("ERROR: Temperature sensor disconnected");
84     }
85
86
87   void checkSafetyLimits() {
88     if (currentTemperature >= OVERHEAT_TEMP) {
89       // Emergency shutdown
90       heaterIntensity = 0;
91       analogWrite(HEATER_PIN, 0);
92       safetyShutdown = true;
93       Serial.print("EMERGENCY: Overheat detected! Temperature: ");
94       Serial.print(currentTemperature);
95       Serial.println("°C - Heater disabled");
96     } else if (currentTemperature >= MAX_SAFE_TEMP && heaterIntensity > 0) {
97       // Reduce heating intensity
98       heaterIntensity = max(0, heaterIntensity - 20);
99       Serial.print("WARNING: High temperature detected ");
100      Serial.print(currentTemperature);
101      Serial.print("°C - Reducing heater intensity to ");
102      Serial.println(heaterIntensity);
103    } else if (currentTemperature < MAX_SAFE_TEMP - 2.0 && safetyShutdown) {
104      // Reset safety shutdown when temperature drops sufficiently
105      safetyShutdown = false;
106      Serial.println("Temperature normalized - Safety shutdown reset");
107    }
108
109
110   void updateHeater() {
111     if (safetyShutdown) {
112       analogWrite(HEATER_PIN, 0);
113       return;
114     }
115
116     // Apply PWM to heater based on intensity
117     int pwmValue = map(heaterIntensity, 0, 100, 0, HEATER_PWM_MAX);
118     analogWrite(HEATER_PIN, pwmValue);
119   }
120
121   void processCommand() {
122     String command = Serial.readStringUntil('\n');
123     command.trim();
124     command.toUpperCase();
125
126     if (command.startsWith("HEAT:")) {
127       // Set heating intensity
128       int intensity = command.substring(5).toInt();
129       setHeaterIntensity(intensity);
130     } else if (command == "TEMP") {

```



```

sketch_aug22a | Arduino IDE 2.3.6
File Edit Sketch Tools Help
Select Board
sketch_aug22a.ino
131     // Send temperature reading
132     sendTemperatureUpdate();
133   } else if (command == "STATUS") {
134     // Send system status
135     sendSystemStatus();
136   } else if (command == "STOP") {
137     // Emergency stop
138     heaterIntensity = 0;
139     Serial.println("Emergency stop - Heater disabled");
140   } else {
141     Serial.println("Unknown command: " + command);
142   }
143
144
145   void setHeaterIntensity(int intensity) {
146     if (safetyShutdown) {
147       Serial.println("Cannot set heater - Safety shutdown active");
148       return;
149     }
150
151     // Clamp intensity to valid range
152     intensity = constrain(intensity, 0, 100);
153     heaterIntensity = intensity;
154
155     Serial.print("Heater intensity set to ");
156     Serial.print(intensity);
157     Serial.println("%");
158   }
159
160   void sendTemperatureUpdate() {
161     Serial.print("TEMP:");
162     Serial.print(currentTemperature, 2);
163     Serial.println();
164   }
165
166   void sendSystemStatus() {
167     Serial.println("== SYSTEM STATUS ==");
168     Serial.print("Temperature: ");
169     Serial.print(currentTemperature, 2);
170     Serial.print("°C");
171     Serial.print("Heater Intensity: ");
172     Serial.print(heaterIntensity);
173     Serial.println("%");
174     Serial.print("Safety Shutdown: ");
175     Serial.println(safetyShutdown ? "ACTIVE" : "Normal");
176     Serial.print("Max Safe Temp: ");
177     Serial.print(MAX_SAFE_TEMP);
178     Serial.println("°C");
179     Serial.print("Emergency Temp: ");
180     Serial.print(OVERHEAT_TEMP);
181     Serial.println("°C");
182     Serial.println("-----");
183   }
184
185
186
187
188
189
190
191
192
193
194
195

```

Figuur 24: Arduino file

```

arduino-bridge.js arduino/
+-----+
1 const { SerialPort } = require("serialport");
2 const { ReadlineParser } = require("@serialport/parser-readline");
3 const WebSocket = require("ws");
4
5 // Configuration
6 const PORT_NAME = process.argv[2] || "/dev/ttyACM0"; // Serial port name (pass as arg or use default)
7 const BAUD_RATE = 9600;
8 const WS_URL = "ws://localhost:5000/ws?type=arduino";
9
10 // Setup serial port connection
11 console.log(`Attempting to connect to Arduino on port ${PORT_NAME}...`);
12 const port = new SerialPort({
13   path: PORT_NAME,
14   baudRate: BAUD_RATE,
15 },
16 );
17 (err) => {
18   if (err) {
19     console.error("Error opening serial port:", err.message);
20     console.log(`Available ports:`);
21
22     // List available ports to help user
23     SerialPort.list().then(ports) => {
24       ports.forEach(port => {
25         console.log(
26           ` - ${port.path} (${port.manufacturer || "unknown manufacturer"})`,
27         );
28       });
29       console.log(
30         "\nTo specify a port, run: node arduino-bridge.js /dev/your-port-name",
31       );
32       process.exit(1);
33     });
34   }
35   return;
36 }
37
38 console.log("Serial port opened successfully");
39 },
40
41 // Create parser for incoming data
42 const parser = port.pipe(new ReadlineParser({ delimiter: "\n" }));
43
44 // Set up WebSocket connection
45 console.log(`Connecting to WebSocket server at ${WS_URL}...`);
46 const ws = new WebSocket(WS_URL);
47
48 // WebSocket connection handlers
49 ws.on("open", () => {
50   console.log("Connected to WebSocket server");
51   console.log("Ready to receive temperature data from Arduino");
52   console.log(`-----`);
53 });
54
55 ws.on("close", () => {
56   console.log("Disconnected from WebSocket server");
57   console.log("Attempting to reconnect...");
58
59 // Simple reconnection logic
60 setTimeout(() => {
61   ws = new WebSocket(WS_URL);
62 }, 5000);

```



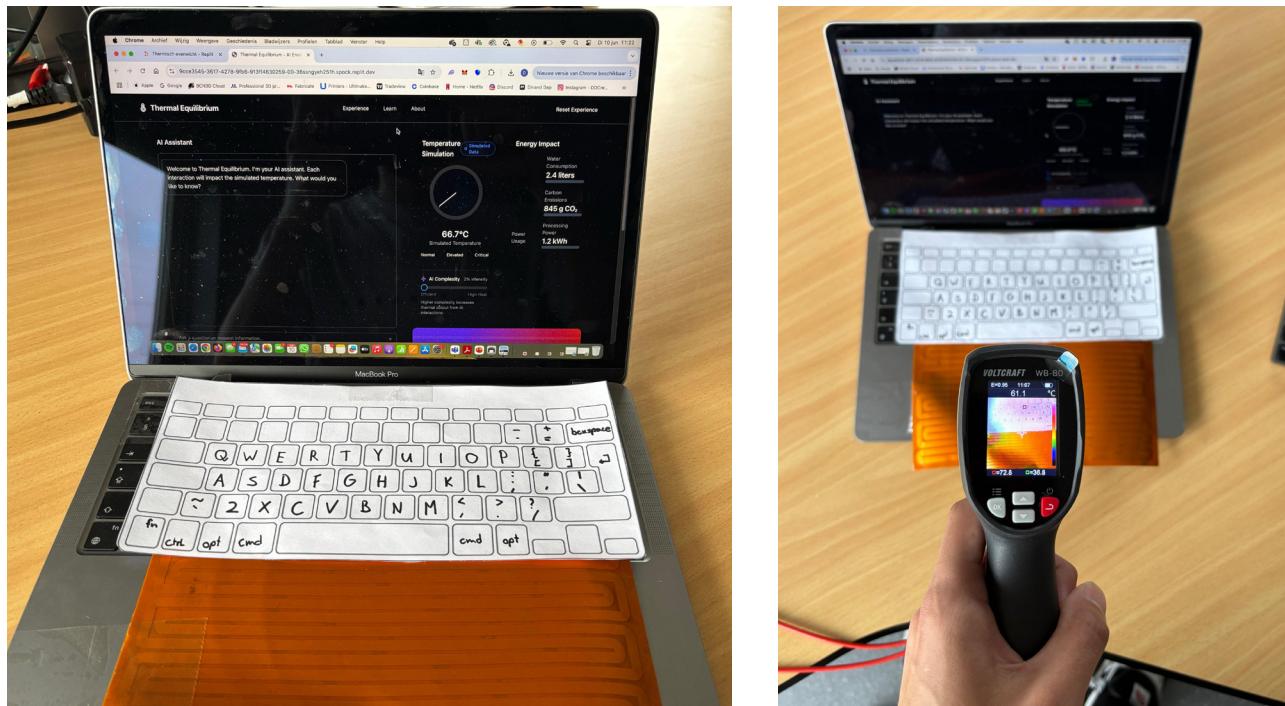
```

arduino-bridge.js arduino/
+-----+
63   });
64
65 ws.on("error", (err) => {
66   console.error("WebSocket error:", err.message);
67 });
68
69 // Handle data from Arduino
70 parser.on("data", (data) => {
71   try {
72     // Clean up and parse JSON data
73     data = data.trim();
74
75     // Skip non-JSON output (like initial messages)
76     if (!data.startsWith("{")) {
77       console.log("Arduino says:", data);
78       return;
79     }
80
81     // Parse JSON data
82     const parsedData = JSON.parse(data);
83
84     // Validate expected format
85     if (
86       parsedData.type === "temperature" &&
87       typeof parsedData.value === "number"
88     ) {
89       console.log(`Temperature reading: ${parsedData.value.toFixed(2)}°C`);
90
91       // Forward to WebSocket if connected
92       if (ws.readyState === WebSocket.OPEN) {
93         ws.send(data);
94       } else {
95         console.log("Received data from Arduino:", parsedData);
96       }
97     } catch (e) {
98       console.error("Error parsing data from Arduino:", e.message);
99       console.error("Raw data:", data);
100    }
101  });
102
103 // Handle serial port errors
104 port.on("error", (err) => {
105   console.error("Serial port error:", err.message);
106 });
107
108
109 // Handle process termination
110 process.on("SIGINT", () => {
111   console.log("\nClosing connections...");
112   if (port.isOpen) {
113     port.close();
114   }
115   if (ws.readyState === WebSocket.OPEN) {
116     ws.close();
117   }
118   process.exit(0);
119 });
120
121 // Log instructions
122 console.log(`\nPress Ctrl+C to exit\n`);

```

Figuur 25: Bridge file

Vervolgens is het concept klaar om getest te worden door de gebruiker



Figuur 26,27: De testopstelling van het "Heated Keyboard" concept, inclusief de interface en de warmtemeter.

Resultaten en Bevindingen (Kernpunten):

Kwantitatief: Gemiddelde temperatuurstijging van 1,0°C/minuut. Deelnemers rapporteerden mild discomfort bij ~42°C en significant discomfort bij ~48°C. 85% van de deelnemers vertoonde gedragsverandering (bijv. sneller typen, pauzes) na 48°C.

Kwalitatief: Deelnemers rapporteerden een verhoogd bewustzijn van de "kosten" van KI-interacties en legden spontaan de link tussen warmte en energieverbruik. Ze ontwikkelden strategieën om met de warmte om te gaan.

Analyse: Het experiment was effectief in het voelbaar maken van onzichtbare energieconsumptie. De warmte fungeerde als metafoor voor digitale impact, wat leidde tot spontane gedragsaanpassing en een groter lichamelijk bewustzijn van digitale activiteiten.

Beperkingen:

De lineaire koppeling tussen KI-gebruik en temperatuur is een vereenvoudiging. De korte duur van het experiment geeft beperkt inzicht in langdurige effecten of gewenning. De focus lag alleen op handwarmte.

Protocol:

Het experiment bestaat uit vier fasen, waarbij de deelnemer geleidelijk intensievere KI-taken uitvoert en de temperatuur van het toetsenbord toeneemt. Door deze 4 fasen heen interacteren de gebruikers op eigen idee met het AI-model door willekeurige vragen te stellen of door taken/opgaves uit te voeren die ze in hun dagelijkse leven ook zouden doen met AI:

1. **Gewenning (3 min):** Typen zonder KI-interactie, temperatuur blijft constant.
2. **Lichte KI-interactie (5 min):** Eenvoudige vragen aan KI, temperatuur stijgt langzaam (tot ~35°C).
3. **Intensieve KI-interactie (5 min):** Complexere KI-taken, temperatuur stijgt gestaag (tot ~55°C).
4. **Extreme KI-interactie (variabel, tot 10 min):** Zeer complexe KI-taken, temperatuur stijgt tot comfortlimiet (~65°C), eindigt bij stopwens of maximale tijd.

Concept 2: Heated Vest

Beschrijving:

Het "Heated Vest" experiment onderzoekt hoe directe warmte op de romp kan dienen als metafoor voor de energiebelasting van KI-systeem. De Verwarmingsfolie is nu verwerkt in een thermoshirt dat onder kleding wordt gedragen. De warmte, dicht op het lichaam, creëert een gevoel van beklemming en overbelasting, wat een fysieke vertaling is van de intensieve energie die KI-technologieën verbruiken. Dit verschilt van het eerdere experiment dat zich richtte op de handen en omgeving.

Doeleind:

- Primair: Een directe, omhullende thermische ervaring creëren die het gevoel van systemische overbelasting door KI-gebruik tastbaar maakt.
- Secundair: Onderzoeken hoe deze kerngerichte warmte de psychologische impact op besluitvorming en bewustzijn beïnvloedt.

Theoretische Onderbouwing:

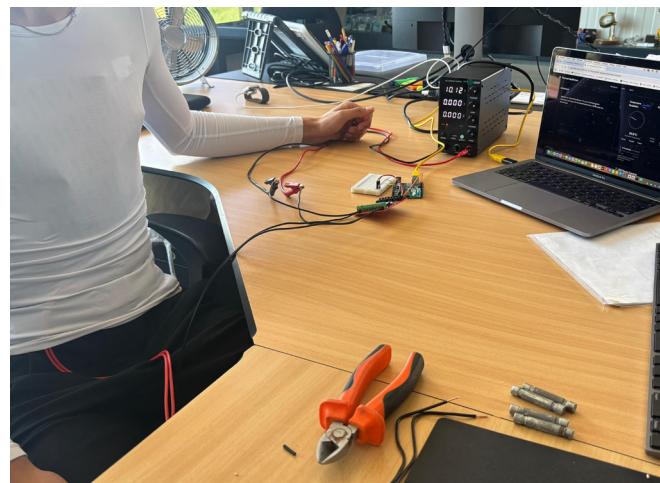
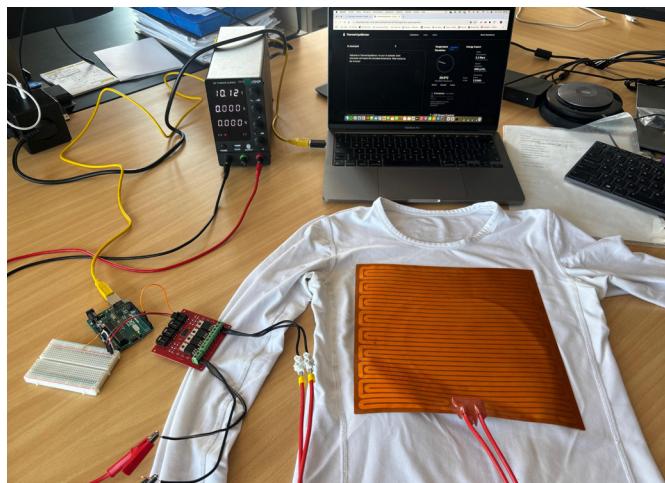
Dit experiment bouwt voort op dezelfde principes als in het "heated keyboard" experiment. Alleen nu wordt voornamelijk gefocused op de romp. Warmte op deze plek roept sterkere, diepgaandere reacties op dan warmte aan de handen. Door deze lichaamsregio te activeren geeft het experiment een intensere ervaring van de impact die KI-systeem.

Materialen:

Nauwsluitend thermoshirt, flexibele verwarmingsfolie, Arduino microcontroller met temperatuursensoren, 12Vdc generator, temperatuurregelaar, thermometer, computer met KI-model, logboek en hartslagsensor (Apple Watch).

Technische Opstelling:

De verwarmingsfolie wordt strategisch geïntegreerd in het thermoshirt, met focus op het borstgebied. Veiligheidslimieten (max. 40°C) worden ingesteld. Het shirt wordt aangesloten op de 12Vdc generator voor interactie met het KI-systeem. De manier van aansturen is hetzelfde als in het experiment "Heated Keyboard", hiervoor is dus gebruik gemaakt van dezelfde files om het experiment te laten functioneren.



Figuur 28,29: De technische opstelling van het "Heated Vest", met het thermoshirt, de verwarmingsfolie en de aansturing.

Figuur 30: Een deelnemer draagt het "Heated Vest" tijdens de testfase van het experiment.

Protocol:

Het experiment omvat voorbereiding en vijf fasen. Net als in het voorgaande concept interacteren de gebruikers op eigen idee met het AI-model door willekeurige vragen te stellen of door taken/opgaves uit te voeren die ze in hun dagelijkse leven ook zouden doen met AI:

1. Voorbereiding: Korte introductie, thermoshirt aantrekken, baseline metingen (lichaamstemperatuur, comfort).
2. Gewenning (1 min): Normale computeractiviteiten zonder KI, verwarming uit.
3. Lichte KI-interactie (3 min): Eenvoudige vragen aan KI, geleidelijke activering verwarming (tot ~32°C).
4. Gemiddelde KI-interactie (5 min): Complexere taken, temperatuur stijgt gestaag (tot ~36°C).
5. Intensieve KI-interactie (5 min): Zeer complexe taken, temperatuur stijgt tot duidelijk discomfort (~38-40°C), eindigt bij stopwens of maximale tijd.
6. Afkoeling en reflectie (2 min): KI-gebruik stopt, verwarming uit, vest blijft aan voor afkoeling en reflectie.

Resultaten en Bevindingen (Kernpunten):

Kwantitatief: Gemiddelde temperatuurstijging van 1,0°C/minuut. Eerste bewuste warmtewaarneming bij 32°C, beklemming bij 36°C. Fysiologische reacties (hartslag, hartslagvariabiliteit) namen toe. Significante afname (42%) in complexe KI-taken in fase 4.

Kwalitatief: Deelnemers ervoeren een uniek gevoel van beklemming en overbelasting, wat fundamenteel verschildde van eerdere experimenten. Dit gevoel werd geassocieerd met de onmogelijkheid om aan de warmte te ontsnappen en leidde tot een dieper, meer existentieel ongemak. Deelnemers ontwikkelden adaptieve strategieën zoals langzamer ademen, beweging minimaliseren en KI-taken drastisch vereenvoudigen.

Analyse: Het "Heated Vest" werkte beter dan het "Heated Keyboard". De warmte op de borst zorgde voor een sterker gevoel en een duidelijker link met het idee van overbelasting. Het voelde voor de deelnemers echt alsof het systeem (en daarmee het milieu) onder druk stond, wat de boodschap krachtiger maakte.

Beperkingen:

De ervaring kan voor sommigen te intens zijn. Individuele verschillen in lichaamsbouw en warmtegevoeligheid beïnvloeden de consistentie. De draagbaarheid beperkt de duur van het experiment. De sociale context van het dragen van een verwarmd kledingstuk kan de ervaring beïnvloeden.

Concept 3: Thermisch evenwicht (installatie)

Beschrijving:

"Thermisch Evenwicht" is een prototype voor een interactieve thermische cabine. Het doel is om bezoekers een directe, lichamelijke ervaring te geven van de energetische impact van KI-technologie. De installatie bestaat uit een zorgvuldig ontworpen ruimte waarin de temperatuur dynamisch reageert op het KI-gebruik van de bezoeker. Naarmate complexere of energie-intensievere KI-taken worden uitgevoerd, stijgt de temperatuur in de cabine, wat directe, fysieke feedback creëert over de anders onzichtbare energetische voetdruk van deze technologieën. De installatie functioneert als een homeostatisch systeem, waarbij de bezoeker actief moet deelnemen om een comfortabel evenwicht te behouden.



Figuur 31: Inspiratiebeelden voor de thermische cabine.

Materialen:

MDF buitenwanden, plexiglas binnenwanden met privacyfolie en geïntegreerde LED-verlichting die reageert op temperatuurveranderingen.

Hardware: Laptopscherm, MOSFET Module, Arduino Uno, Breadboard, Uploading kabel.

Structuur:

Een vrijstaande, geïsoleerde cabine (1 x 0.8 x 2.0 meter) met een transparante plexiglas voorzijde. Modulair ontwerp voor eenvoudige installatie en transport. Toegangsdeur met magnetische afdichting voor thermische isolatie. Deze installatie is opgebouwd uit 3 lossen panelen, zoals hier rechts in figuur 32 is te zien. Figuur 33 toont een compleet paneel alleen zonder verwarmingselement. Nu moeten de panelen van verwarmingselementen voorzien worden. Dit ging alleen minder gemakkelijk dan verwacht.

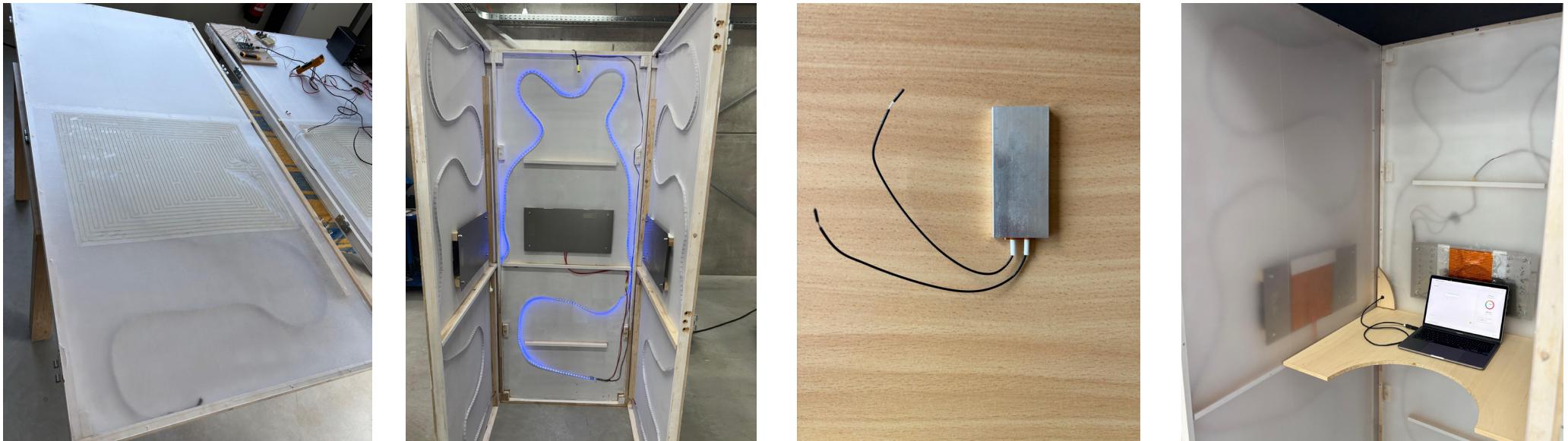
De Cabine (Ontwerkenmerken):

Esthetiek: Minimalistisch en clean design.

Kleurenpalet & Ambient Lighting: De cabine gebruikt LED-verlichting die de thermische toestand visualiseert. Wanneer er met KI wordt geïnteracteerd, gloeiit de cabine oranje/rood om warmte te symboliseren. Zonder interactie is de cabine blauw en geeft zo de afgekoelde toestand weer. Dit versterkt de visuele stimulatie van de temperatuurverandering.



Figuur 32,33: de modulaire opbouw van de thermische cabine.



Figuur 34,35,36,37: De verschillende iteraties van de verwarmingselementen die zijn getest voor de installatie.

Thermisch Systeem:

Voor de verwarming van de installatie heb ik vier verschillende versies van verwarmingselementen getest.

In versie 1 (figuur 34) gebruikte ik een verwarmingsfolie van 600 x 800 mm. Ondanks het grote formaat bleek het vermogen veel te laag hierdoor was er nauwelijks hitte in de cabine te ervaren tijdens het testen.

Voor versie 2 heb ik de werkende verwarmingsfolie uit eerdere experimenten gebruikt, die zeker 95°C kan bereiken. Om de warmte beter te verspreiden, heb ik deze folie vastgemaakt aan een grote, maar erg dunne stalen plaat. Deze opstelling is driemaal uitgevoerd, zodat elk paneel van een verwarmingscomponent was voorzien. Uit de testen bleek dat deze opstelling de hitte-ervaring significant verbeterde. Echter, de opwarmtijd en de snelheid waarmee de temperatuur steeg tijdens interactie met het systeem waren nog niet snel genoeg. Het duurde te lang voordat de hitte echt voelbaar was in de installatie.

Dit bracht mij bij versie 3, waarbij ik een PTC-element (figuur 36) heb toegepast. Dit verwarmingselement kan significant hogere temperaturen bereiken; specifiek dit element kan bij 12V een temperatuur van 240°C behalen. Ik heb dit component op dezelfde manier aan de stalen platen bevestigd om de warmteverspreiding te vergroten. Helaas bleek bij het testen dat het element zo heet werd dat het plexiglas aan de binnenkant langzaam begon te smelten. Hierdoor was het gebruik van de installatie gelimiteerd en bleek deze versie niet bruikbaar voor het project.

Als laatste versie heb ik de opstelling van versie 2 gebruikt, maar nu zijn de verwarmingselementen aan de voorkant gemonteerd in plaats van aan de achterkant. Op deze manier wordt de directe stralingswarmte onmiddellijk doorgegeven aan het plexiglas, waardoor de cabine sneller opwarmt. Met deze opstelling wordt het maximale uit het thermisch systeem gehaald en daarmee de best mogelijke embodied experience bereikt.

Het volledige systeem draait op drie 12Vdc generatoren, die elk minimaal ~25A leveren. In de installatie is ook een temperatuursensor (DS18B20) opgenomen.

Software:

Ook voor dit concept kunnen we dezelfde software structuur hanteren waar we dus gebruik maken van; een arduino file (hardware), bridge file (link tussen de hardware componenten en server waarop de web app draait) en de web app zelf (voor interacties van de gebruiker). Deze zien er als volgt uit

```

1 //include <OneWire.h>
2 //include <DallasTemperature.h>
3 //include <Adafruit_NeoPixel.h>
4
5 // Temperature sensor
6 #define ONE_WIRE_BUS 2
7 #define THERMAL_EQUIPMENT_Precision 12
8 OneWire oneWire(ONE_WIRE_BUS);
9 DallasTemperature sensors(&oneWire);
10
11 // Triple heating elements
12 #define HEATER_PIN_1_6 // Original heater
13 #define HEATER_PIN_2_9 // Additional heater 2
14 #define HEATER_PIN_3_18 // Additional heater 3
15
16 // Safety ventilator
17 #define VENTILATOR_PIN 8 // Fresh air circulation
18
19 // LED configuration - FULL STRIPS
20 #define LED_STRIP_1 300 // All 300 LEDs
21 #define LED_PIN_1 3
22 #define LED_PIN_2 4
23 #define LED_PIN_3 5
24
25 // Heater objects
26 Adafruit_NeoPixel strip1(LED_PIN_1, LED_PIN_1, NEO_GRB + NEO_KHZ800);
27 Adafruit_NeoPixel strip2(LED_PIN_2, LED_PIN_2, NEO_GRB + NEO_KHZ800);
28 Adafruit_NeoPixel strip3(LED_PIN_3, LED_PIN_3, NEO_GRB + NEO_KHZ800);
29
30 // System variables
31 float currentTemperature = 0.0;
32 float baselineTemperature = 20.0;
33 float lastTempUpdate = 0;
34 unsigned long lastTempRead = 0;
35 unsigned long lastTempUpdate = 0;
36 unsigned long lastTempUpdate = 0;
37 unsigned long lastTempUpdate = 0;
38
39 // Timing
40 const unsigned long TEMP_INTERVAL = 1000;
41 const unsigned long HEATER_INTERVAL = 1000;
42 const unsigned long LED_INTERVAL = 100;
43
44 // Enhanced heating sequence with accumulation
45 struct HeatingSequence {
46   bool isActive;
47   float lastTime;
48   float nextTime; // Always 95% as requested
49   unsigned long remainingTime; // Accumulated time from multiple questions
50   const float totalAccumulated; // Total time added from all questions
51 };
52
53 HeatingSequence heating = {
54   .isActive = false,
55   .lastTime = 0,
56   .nextTime = 0,
57   .remainingTime = 0,
58   .totalAccumulated = 0
59 };
60
61 // LED variables
62 uint8_t ledBrightness = 255; // Maximum brightness (1000)
63 uint8_t animationMode = 0; // 0-off, 1-heating, 2-cooling, 3-AT, 4-transition
64 unsigned long animationStart = 0;
65 int waveCount = 0;
66
67 // Transition variables for smooth color change
68 bool isTransitioning = false;
69 unsigned long transitionStartTime = 0;
70 const unsigned long TRANSITION_DURATION = 5000; // 5 seconds for smooth transition
71
72 // Safety limits for proper heating operation
73 const float OVERHEAT_TEMP = 45.0; // Increased from 35°C to allow proper heating
74 const float OVERTEMP = 50.0; // Increased from 40°C
75 const float OVERHEAT_SHUTDOWN = 50.0; // Increased from 40°C
76 bool safetyShutdown = false;
77
78 void setup() {
79   Serial.begin(9600);
80
81   // Initialize temperature sensor
82   sensors.begin();
83   sensors.setResolution(TEMPERATURE_PRECISION);
84
85   // Initialize triple heaters
86   pinMode(HEATER_PIN_1, OUTPUT);
87   pinMode(HEATER_PIN_2, OUTPUT);
88   pinMode(HEATER_PIN_3, OUTPUT);
89   analogWrite(HEATER_PIN_1, 0);
90   analogWrite(HEATER_PIN_2, 0);
91   analogWrite(HEATER_PIN_3, 0);
92
93   // Initialize safety ventilator
94   pinMode(VENTILATOR_PIN, OUTPUT);
95   digitalWrite(VENTILATOR_PIN, HIGH);
96
97   // Initialize LED strips
98   strip1.begin("Initializing FULL LED strips (300 LEDs each...)");
99   strip2.begin();
100  strip3.begin();
101
102  strip1.setBrightness(ledBrightness);
103  strip2.setBrightness(ledBrightness);
104  strip3.setBrightness(ledBrightness);
105
106  // Clear and test LEDs
107  clearAll();
108
109  // Quick startup test - first 10 LEDs
110  Serial.println("LED startup test..."); 
111  for (int i = 0; i < 10; i++) {
112    strip1.setPixelColor(i, strip1.Color(0, 0, 0));
113    strip1.setPixelColor(i, strip1.Color(0, 30, 0));
114    strip1.setPixelColor(i, strip1.Color(0, 0, 30));
115  }
116  showAll();
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464

```

Figuur 38: Arduino file

Sketch: sketch_arduino

```

466 // Apply triple smoothstep for extremely fluid transition - eliminates all jumps
467 float p = rawProgress;
468 float smoothProgress = p * p * p * (p * (p - 15.0) + 10.0); // Quintic smoothstep
469
470 // Continue heartbeat pattern during transition
471 pulsePhase += 0.12;
472 if (pulsePhase > 6.28) pulsePhase = 0;
473
474 float primaryBeat = sin(pulsePhase);
475 float secondaryBeat = sin(pulsePhase * 1.8 + 0.3);
476 float tertiaryBeat = sin(pulsePhase * 2.6 + 0.4) / 1.4;
477 float smoothIntensity = sin(0.0, heartbeatPattern);
478 float intensity = smoothIntensity * smoothIntensity * smoothIntensity;
479 float intensity = 0.08 + 0.92 * intensity;
480
481 for (int i = 0; i < LED_PER_STRIP; i++) {
482   // Smooth pixel color interpolation with no visible steps
483   // Starts here with a color of 0, 0, 0
484   float heatingRed = 255.0 * intensity;
485   float heatingGreen = 0.0 * intensity; // Slightly more orange
486   float heatingBlue = 255.0 * intensity;
487
488   // End: Pixel color is color
489   float coolingGreen = 0.0 * intensity; // Maintain green component
490   float coolingBlue = 255.0 * intensity;
491
492   // Direct interpolation with ultra-smooth progression
493   float red = heatingRed * (1.0 - smoothProgress) + coolingRed * smoothProgress;
494   float green = heatingGreen; // Keep consistent green throughout
495   float blue = heatingBlue * (1.0 - smoothProgress) + coolingBlue * smoothProgress;
496
497   // Apply ultra-precise color values with floating point precision
498   unsigned int finalRed = (unsigned int)(red * 0.5 + 255); // +0.5 for proper rounding
499   unsigned int finalGreen = (unsigned int)(green * 0.5 + 255);
500   unsigned int finalBlue = (unsigned int)constrain(blue * 0.5, 0, 255);
501
502   uint8_t color = strip1.color(initial, finalRed, finalGreen, finalBlue);
503   strip1.setPixelColor(i, color);
504   strip2.setPixelColor(i, color);
505   strip3.setPixelColor(i, color);
506 }
507
508 lastUpdate = currentTime;
509
510 void clearAll() {
511   strip1.clear();
512   strip2.clear();
513   strip3.clear();
514 }
515
516 void showAll() {
517   strip1.show();
518   strip2.show();
519   strip3.show();
520 }
521
522 void processCommand() {
523   String input = Serial.readStringUntil('\n');
524   input.trim();
525
526   // Handle both direct commands and JSON WebSocket commands
527   String command = input;
528
529   // Check if it's a JSON command from websocket
530   if (input.startsWith("{}")) {
531     // Parse JSON to extract command
532     int cmdIndex = input.indexOf("command");
533     int cmdStart = cmdIndex + "command".length() + 1;
534     int cmdEnd = cmdIndex + "command".length() + 11;
535     String cmd = input.substring(cmdStart, cmdEnd);
536     command = input.substring(cmdStart, cmdEnd);
537     Serial.print("Parsed websocket command: ");
538     Serial.println(command);
539   }
540 }
541
542 command.toLowerCase();
543
544 if (command.startsWith("HEAT")) {
545   int timeSeconds = command.substring(5).toInt();
546   startHeatingSequence(timeSeconds);
547 }
548 else if (command.startsWith("COOL")) {
549   int brightness = command.substring(5).toInt();
550   setLidBrightness(brightness);
551 }
552 else if (command.startsWith("STOP")) {
553   stopHeating();
554 }
555 else if (command.startsWith("EMERGENCY")) {
556   int mode = command.substring(8).toInt();
557   setEmergencyMode(mode);
558 }
559 else if (command.startsWith("TEST")) {
560   test();
561 }
562 else if (command.startsWith("STATUS")) {
563   int statusIndex = command.indexOf("status");
564   int statusEnd = statusIndex + "status".length() + 1;
565   String status = input.substring(statusIndex, statusEnd);
566   if (status.equals("on")) {
567     turnOn();
568   }
569   else if (status.equals("off")) {
570     turnOff();
571   }
572 }
573
574 void startHeatingSequence(int questionTime) {
575   if (safetyShutdown) {
576     Serial.println("Cannot start - safety shutdown");
577     return;
578   }
579
580   Serial.print("Unknown command: ");
581   Serial.println(command);
582 }
583
584 void startHeating() {
585   Serial.print("Starting heating sequence... ");
586   Serial.print("Heating stopped - cooling animation active");
587   test();
588 }
589
590 void stopHeating() {
591   Serial.print("Emergency stop - all systems off");
592   emergencyStop();
593 }
594
595 void emergencyStop() {
596   if (heating.IsActive) {
597     heating.IsActive = false;
598     heating.remainingTime = 0;
599     heating.totalAccumulated = 0;
600     heating.startTime = 0;
601     heating.remainingTime = heating.totalAccumulated - elapsed;
602   }
603
604   Serial.print("Heating stopped - cooling animation active");
605   Serial.print("Heating stopped - heating active");
606   Serial.print("Emergency stop - all systems off");
607 }
608
609 void test() {
610   Serial.print("Testing All LED strips... ");
611   Serial.print("Test complete");
612 }
613
614 void testAll() {
615   Serial.print("Testing All LED strips... ");
616   Serial.print("Test complete");
617 }
618
619 void turnOn() {
620   Serial.print("Turning on heater");
621   Serial.print("Heater is active");
622   Serial.print("Heater brightness is 100");
623   Serial.print("Heating is active");
624   Serial.print("Heating brightness is 100");
625   Serial.print("Heating total accumulated is 0");
626   Serial.print("Heating remaining time is 0");
627 }
628
629 void turnOff() {
630   Serial.print("Turning off heater");
631   Serial.print("Heater is inactive");
632   Serial.print("Heater brightness is 0");
633   Serial.print("Heating is inactive");
634   Serial.print("Heating brightness is 0");
635 }
636
637 void stopHeating() {
638   heating.IsActive = false;
639   heating.remainingTime = 0;
640   heating.totalAccumulated = 0;
641   heating.startTime = 0;
642   analogWrite(MAINT_PIN_1, 0);
643   analogWrite(MAINT_PIN_2, 0);
644   analogWrite(MAINT_PIN_3, 0);
645   digitalWrite(MAINT_PIN_4, LOW); // Turn off ventilator with heaters
646   animationMode = 2; // Switch to cooling animation
647   animationStartTime = millis(); // Reset animation timing
648   animationRemainingTime = 0; // Set initial time for cooling
649   Serial.print("Heating stopped - switching to cooling animation (mode 0)");
650   Serial.print(animationMode);
651   Serial.print("Heating stopped");
652 }
653
654 void setEmergencyMode(int mode) {
655   Serial.print("Emergency mode: ");
656   Serial.print(mode);
657   Serial.print("Current temperature: ");
658   Serial.println(currentTemperature, 2);
659 }
660
661 void setLidBrightness(int brightness) {
662   Serial.print("Setting lid brightness: ");
663   Serial.print(brightness);
664   Serial.print("Current temperature: ");
665   Serial.println(currentTemperature);
666 }
667
668 void setHeating() {
669   Serial.print("Setting heating");
670   Serial.print("Current temperature: ");
671   Serial.print(currentTemperature);
672   Serial.print("Current humidity: ");
673   Serial.print(humidity);
674   Serial.print("Current time: ");
675   Serial.print(currentTime);
676   Serial.print("Current date: ");
677   Serial.print(currentDate);
678   Serial.print("Current location: ");
679   Serial.print(currentLocation);
680   Serial.print("Current weather: ");
681   Serial.print(currentWeather);
682   Serial.print("Current wind: ");
683   Serial.print(currentWind);
684   Serial.print("Current pressure: ");
685   Serial.print(currentPressure);
686   Serial.print("Current visibility: ");
687   Serial.print(currentVisibility);
688   Serial.print("Current clouds: ");
689   Serial.print(currentClouds);
690   Serial.print("Current UV index: ");
691   Serial.print(currentUVIndex);
692   Serial.print("Current dew point: ");
693   Serial.print(currentDewPoint);
694   Serial.print("Current feels like: ");
695   Serial.print(currentFeelsLike);
696   Serial.print("Current temperature min: ");
697   Serial.print(currentTempMin);
698   Serial.print("Current temperature max: ");
699   Serial.print(currentTempMax);
700   Serial.print("Current temperature feels like min: ");
701   Serial.print(currentTempFeelsLikeMin);
702   Serial.print("Current temperature feels like max: ");
703   Serial.print(currentTempFeelsLikeMax);
704   Serial.print("Current temperature feels like current: ");
705   Serial.print(currentTempFeelsLikeCurrent);
706   Serial.print("Current temperature feels like current min: ");
707   Serial.print(currentTempFeelsLikeCurrentMin);
708   Serial.print("Current temperature feels like current max: ");
709   Serial.print(currentTempFeelsLikeCurrentMax);
710   Serial.print("Current temperature feels like current feels like: ");
711   Serial.print(currentTempFeelsLikeCurrentFeelsLike);
712   Serial.print("Current temperature feels like current feels like min: ");
713   Serial.print(currentTempFeelsLikeCurrentFeelsLikeMin);
714   Serial.print("Current temperature feels like current feels like max: ");
715   Serial.print(currentTempFeelsLikeCurrentFeelsLikeMax);
716   Serial.print("Current temperature feels like current feels like current: ");
717   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrent);
718   Serial.print("Current temperature feels like current feels like current min: ");
719   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentMin);
720   Serial.print("Current temperature feels like current feels like current max: ");
721   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentMax);
722   Serial.print("Current temperature feels like current feels like current feels like: ");
723   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLike);
724   Serial.print("Current temperature feels like current feels like current feels like min: ");
725   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeMin);
726   Serial.print("Current temperature feels like current feels like current feels like max: ");
727   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeMax);
728   Serial.print("Current temperature feels like current feels like current feels like current: ");
729   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrent);
730   Serial.print("Current temperature feels like current feels like current feels like current min: ");
731   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentMin);
732   Serial.print("Current temperature feels like current feels like current feels like current max: ");
733   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentMax);
734   Serial.print("Current temperature feels like current feels like current feels like current feels like: ");
735   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLike);
736   Serial.print("Current temperature feels like current feels like current feels like current feels like min: ");
737   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeMin);
738   Serial.print("Current temperature feels like current feels like current feels like current feels like max: ");
739   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeMax);
740   Serial.print("Current temperature feels like current feels like current feels like current feels like current: ");
741   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrent);
742   Serial.print("Current temperature feels like current feels like current feels like current feels like current min: ");
743   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentMin);
744   Serial.print("Current temperature feels like current feels like current feels like current feels like current max: ");
745   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentMax);
746   Serial.print("Current temperature feels like current feels like current feels like current feels like current feels like: ");
747   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLike);
748   Serial.print("Current temperature feels like current feels like current feels like current feels like current feels like min: ");
749   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeMin);
750   Serial.print("Current temperature feels like current feels like current feels like current feels like current feels like max: ");
751   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeMax);
752   Serial.print("Current temperature feels like current feels like current feels like current feels like current feels like current: ");
753   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrent);
754   Serial.print("Current temperature feels like current min: ");
755   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentMin);
756   Serial.print("Current temperature feels like current max: ");
757   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentMax);
758   Serial.print("Current temperature feels like current feels like: ");
759   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLike);
760   Serial.print("Current temperature feels like current feels like min: ");
761   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeMin);
762   Serial.print("Current temperature feels like current feels like max: ");
763   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeMax);
764   Serial.print("Current temperature feels like current feels like current: ");
765   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrent);
766   Serial.print("Current temperature feels like current min: ");
767   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentMin);
768   Serial.print("Current temperature feels like current max: ");
769   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentMax);
770   Serial.print("Current temperature feels like current feels like: ");
771   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLike);
772   Serial.print("Current temperature feels like current feels like min: ");
773   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeMin);
774   Serial.print("Current temperature feels like current feels like max: ");
775   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeMax);
776   Serial.print("Current temperature feels like current feels like current: ");
777   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrent);
778   Serial.print("Current temperature feels like current min: ");
779   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentMin);
780   Serial.print("Current temperature feels like current max: ");
781   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentMax);
782   Serial.print("Current temperature feels like current feels like: ");
783   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLike);
784   Serial.print("Current temperature feels like current feels like min: ");
785   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeMin);
786   Serial.print("Current temperature feels like current feels like max: ");
787   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeMax);
788   Serial.print("Current temperature feels like current feels like current: ");
789   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrent);
790   Serial.print("Current temperature feels like current min: ");
791   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentMin);
792   Serial.print("Current temperature feels like current max: ");
793   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentMax);
794   Serial.print("Current temperature feels like current feels like: ");
795   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLike);
796   Serial.print("Current temperature feels like current feels like min: ");
797   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeMin);
798   Serial.print("Current temperature feels like current feels like max: ");
799   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeMax);
800   Serial.print("Current temperature feels like current feels like current: ");
801   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrent);
802   Serial.print("Current temperature feels like current min: ");
803   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentMin);
804   Serial.print("Current temperature feels like current max: ");
805   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentMax);
806   Serial.print("Current temperature feels like current feels like: ");
807   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLike);
808   Serial.print("Current temperature feels like current feels like min: ");
809   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeMin);
810   Serial.print("Current temperature feels like current feels like max: ");
811   Serial.print(currentTempFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeCurrentFeelsLikeMax);
812 }

```

Figuur 38: Arduino file

CONCEPTEN EMBODIED EXPERIENCE

12

```

  arduino > local-arduino-bridge.js > ...
  Format  arduino > local-arduino-bridge.js > ...
  Format  arduino > local-arduino-bridge.js > ...
  Format  arduino > local-arduino-bridge.js > ...
  1 const { SerialPort } = require("serialport");
  2 const { ReadlineParser } = require("@serialport/parser-readline");
  3 const WebSocket = require("ws");
  4
  5
  6 const REPLIT_URL =
  7   "wss://9cce3545-3617-4278-9fb6-913ff4630259-00-
  8   36ssngyeh25fh.spock.replit.dev/ws?type=arduino";
  9
 10 // Serial port configuration
 11 const BAUD_RATE = 9600;
 12
 13 // Auto-detect Arduino port if not specified
 14 let portName = process.argv[2];
 15
 16 async function findArduinoPort() {
 17   try {
 18     const ports = await SerialPort.list();
 19     console.log("🔍 Scanning for Arduino devices...");
 20
 21     if (ports.length === 0) {
 22       throw new Error(
 23         "No serial ports found. Make sure your Arduino is connected via USB."
 24       );
 25     }
 26
 27     console.log("💻 Available serial ports:");
 28     ports.forEach((port, index) => {
 29       const manufacturer = port.manufacturer || "Unknown";
 30       const vendorId = port.vendorId || "Unknown";
 31       const productId = port.productId || "Unknown";
 32       console.log(
 33         ` ${index + 1}. ${port.path} - ${manufacturer} (VID: ${vendorId}, PID: ${productId})`,
 34       );
 35     });
 36
 37     // Try to auto-detect Arduino
 38     const arduinoPorts = ports.filter((port) => {
 39       const manufacturer = (port.manufacturer || "").toLowerCase();
 40       const vendorId = port.vendorId || "";
 41       return (
 42         manufacturer.includes("arduino") ||
 43         manufacturer.includes("ch340") ||
 44         manufacturer.includes("cp210") ||
 45         manufacturer.includes("ftdi") ||
 46         vendorId === "2341" || // Arduino VID
 47         vendorId === "1A86" || // CH340 VID
 48         vendorId === "0403" ||
 49       ); // FTDI VID
 50     });
 51
 52     if (arduinoPorts.length > 0) {
 53       console.log(`💡 Auto-detected Arduino at: ${arduinoPorts[0].path}`);
 54       return arduinoPorts[0].path;
 55     }
 56
 57     // If no Arduino detected, suggest the first available port
 58     if (ports.length > 0) {
 59       console.log(`💡 No Arduino auto-detected. Trying first available port: ${ports[0].path}`);
 60       return ports[0].path;
 61     }
 62
 63     throw new Error("No suitable serial ports found");
 64   } catch (error) {
 65     throw new Error(`Port detection failed: ${error.message}`);
 66   }
 67 }
 68
 69
 70 console.log(`💡 Local Arduino Bridge for Replit-hosted Thermal Equilibrium`);
 71 console.log(`=====`);
 72
 73 console.log(`🌐 Target Replit URL: ${REPLIT_URL}`);
 74 console.log(`arduino Port: ${portName}`);
 75 console.log(``);
 76
 77 // WebSocket connection to Replit app
 78 let ws = null;
 79 let reconnectAttempts = 0;
 80 const MAX_RECONNECT_ATTEMPTS = 5;
 81 let simulationModeActive = false;
 82 let reconnectCheckInterval = null;
 83
 84 function connectToReplit() {
 85   console.log(`🌐 Connecting to Replit WebSocket server...`);
 86
 87   ws = new WebSocket(REPLIT_URL);
 88
 89   ws.on("open", () => {
 90     console.log(`💡 Connected to Replit Thermal Equilibrium app`);
 91     reconnectAttempts = 0;
 92   });
 93
 94   ws.on("message", (data) => {
 95     try {
 96       const message = JSON.parse(data);
 97
 98       if (
 99         message.type === "heater_control" &&
 100        typeof message.intensity === "number"
 101     ) {
 102       // Send heater command to Arduino
 103       const heaterCommand = `HEAT:${message.intensity}\n`;
 104       console.log(`💡 Received heater command: ${message.intensity}`);
 105
 106       if (port && port.isOpen) {
 107         port.write(heaterCommand);
 108         console.log(`💡 Sent to Arduino: ${heaterCommand.trim()}`);
 109       } else {
 110         console.log(`⚠️ Arduino not connected - heater command not sent`);
 111       }
 112     } else if (message.type === "command" && message.command) {
 113       // Send general command to Arduino (including LED commands)
 114       const command = message.command + "\n";
 115     }
 116   } catch (error) {
 117     console.error(`❌ Error parsing WebSocket message: ${error.message}`);
 118   }
 119 }
 120
 121 } else {
 122   console.log(`⚠️ Arduino not connected - command not sent`);
 123 }
 124
 125 } catch (error) {
 126   console.error(`❌ Error parsing WebSocket message: ${error.message}`);
 127 }
 128
 129 ws.on("error", (error) => {
 130   console.error(`❌ WebSocket error: ${error.message}`);
 131
 132   if (error.code === "ECONNREFUSED") {
 133     console.log(`💡 Connection refused. Make sure:`);
 134     console.log(`  1. Your Replit app is running`);
 135     console.log(`  2. The REPLIT_URL is correct`);
 136     console.log(`  3. WebSocket server is accepting connections`);
 137     console.log(`  4. Try both .replit.app and .repl.co domains`);
 138   }
 139
 140   if (
 141     error.code === "ENOTFOUND" ||
 142     error.message.includes("getaddrinfo ENOTFOUND")
 143   ) {
 144     console.log(`💡 DNS resolution failed. Check:`);
 145     console.log(`  1. The URL format is correct`);
 146     console.log(`  2. Your internet connection`);
 147     console.log(`  3. The Replit app is deployed and accessible`);
 148     console.log(`  4. Current URL: ${REPLIT_URL}`);
 149   }
 150 }
 151
 152 ws.on("close", (code, reason) => {
 153   console.log(`❌ Disconnected from Replit server (code: ${code})`);
 154   if (reason) {
 155     console.log(`Reason: ${reason.toString()}`);
 156   }
 157
 158   // Check if disconnection was due to simulation mode activation
 159   if (reason && reason.toString().includes('simulation_mode_activated')) {
 160     console.log(`⚠️ Connection closed by server - simulation mode active`);
 161     console.log(`Waiting for Arduino mode to be selected...`);
 162     console.log(`Monitoring for reconnection opportunity`);
 163
 164     simulationModeActive = true;
 165     startReconnectionMonitoring();
 166     return; // Don't attempt to reconnect immediately
 167   }
 168
 169   // Only auto-reconnect for unexpected disconnections (not simulation mode)
 170   if (reconnectAttempts < MAX_RECONNECT_ATTEMPTS) {
 171     reconnectAttempts++;
 172     console.log(`Attempting to reconnect...`);
 173     setTimeout(connectToReplit, 5000);
 174   } else {
 175     console.log(`Max reconnection attempts reached. Connection and restart.`);
 176   }
 177
 178   // Monitor for Arduino mode activation
 179   function startReconnectionMonitoring() {
 180     if (reconnectCheckInterval) {
 181       clearInterval(reconnectCheckInterval);
 182     }
 183
 184     console.log(`🔍 Monitoring for Arduino mode activation`);
 185     reconnectCheckInterval = setInterval(reconnectCheckInterval, 1000);
 186
 187     if (!simulationModeActive) {
 188       return; // Already reconnected
 189     }
 190
 191     try {
 192       // Check if Arduino mode is active
 193       const testWs = new WebSocket(REPLIT_URL);
 194
 195       testWs.on("open", () => {
 196         console.log(`💡 Arduino mode detected!`);
 197         bridge!();
 198         testWs.close();
 199
 200         simulationModeActive = false;
 201         clearInterval(reconnectCheckInterval);
 202         reconnectCheckInterval = null;
 203
 204         // Reconnect with fresh state
 205         reconnectAttempts = 0;
 206         setTimeout(connectToReplit, 1000);
 207       });
 208
 209       testWs.on("error", () => {
 210         // Still in simulation mode or testWs.close();
 211       });
 212
 213       testWs.on("close", (code, reason) => {
 214         // Still in simulation mode
 215         if (reason && reason.toString().includes('simulation_mode_activated')) {
 216           console.log(`⚠️ Connection closed by server - simulation mode active`);
 217           simulationModeActive = true;
 218           startReconnectionMonitoring();
 219           return; // Don't attempt to reconnect immediately
 220         }
 221
 222         // Still in simulation mode
 223         if (reason && reason.toString().includes('simulation_mode_activated')) {
 224           testWs.close();
 225         }
 226
 227         // catch (error) {
 228           // Connection test failed, continue
 229         }
 230
 231       });
 232
 233     }
 234   }
 235
 236   L1, Col 1 • Spaces: 2 History ⏪
 237   L1, Col 1 • Spaces: 2 History ⏪
 238   L1, Col 1 • Spaces: 2 History ⏪
 239   L1, Col 1 • Spaces: 2 History ⏪
 240   L1, Col 1 • Spaces: 2 History ⏪
 241   L1, Col 1 • Spaces: 2 History ⏪
 242   L1, Col 1 • Spaces: 2 History ⏪
 243   L1, Col 1 • Spaces: 2 History ⏪
 244   L1, Col 1 • Spaces: 2 History ⏪
 245   L1, Col 1 • Spaces: 2 History ⏪
 246   L1, Col 1 • Spaces: 2 History ⏪
 247   L1, Col 1 • Spaces: 2 History ⏪
 248   L1, Col 1 • Spaces: 2 History ⏪
 249   L1, Col 1 • Spaces: 2 History ⏪
 250   L1, Col 1 • Spaces: 2 History ⏪
 251   L1, Col 1 • Spaces: 2 History ⏪
 252   L1, Col 1 • Spaces: 2 History ⏪
 253   L1, Col 1 • Spaces: 2 History ⏪
 254   L1, Col 1 • Spaces: 2 History ⏪
 255   L1, Col 1 • Spaces: 2 History ⏪
 256   L1, Col 1 • Spaces: 2 History ⏪
 257   L1, Col 1 • Spaces: 2 History ⏪
 258   L1, Col 1 • Spaces: 2 History ⏪
 259   L1, Col 1 • Spaces: 2 History ⏪
 260   L1, Col 1 • Spaces: 2 History ⏪
 261   L1, Col 1 • Spaces: 2 History ⏪
 262   L1, Col 1 • Spaces: 2 History ⏪
 263   L1, Col 1 • Spaces: 2 History ⏪
 264   L1, Col 1 • Spaces: 2 History ⏪
 265   L1, Col 1 • Spaces: 2 History ⏪
 266   L1, Col 1 • Spaces: 2 History ⏪
 267   L1, Col 1 • Spaces: 2 History ⏪
 268   L1, Col 1 • Spaces: 2 History ⏪
 269   L1, Col 1 • Spaces: 2 History ⏪
 270   L1, Col 1 • Spaces: 2 History ⏪
 271   L1, Col 1 • Spaces: 2 History ⏪
 272   L1, Col 1 • Spaces: 2 History ⏪
 273   L1, Col 1 • Spaces: 2 History ⏪
 274   L1, Col 1 • Spaces: 2 History ⏪
 275   L1, Col 1 • Spaces: 2 History ⏪
 276   L1, Col 1 • Spaces: 2 History ⏪
 277   L1, Col 1 • Spaces: 2 History ⏪
 278   L1, Col 1 • Spaces: 2 History ⏪
 279   L1, Col 1 • Spaces: 2 History ⏪
 280   L1, Col 1 • Spaces: 2 History ⏪
 281   L1, Col 1 • Spaces: 2 History ⏪
 282   L1, Col 1 • Spaces: 2 History ⏪
 283   L1, Col 1 • Spaces: 2 History ⏪
 284   L1, Col 1 • Spaces: 2 History ⏪
 285   L1, Col 1 • Spaces: 2 History ⏪
 286   L1, Col 1 • Spaces: 2 History ⏪
 287   L1, Col 1 • Spaces: 2 History ⏪
 288   L1, Col 1 • Spaces: 2 History ⏪
 289   L1, Col 1 • Spaces: 2 History ⏪
 290   L1, Col 1 • Spaces: 2 History ⏪
 291   L1, Col 1 • Spaces: 2 History ⏪
 292   L1, Col 1 • Spaces: 2 History ⏪
 293   L1, Col 1 • Spaces: 2 History ⏪
 294   L1, Col 1 • Spaces: 2 History ⏪
 295   L1, Col 1 • Spaces: 2 History ⏪
 296   L1, Col 1 • Spaces: 2 History ⏪
 297   L1, Col 1 • Spaces: 2 History ⏪
 298   L1, Col 1 • Spaces: 2 History ⏪
 299   L1, Col 1 • Spaces: 2 History ⏪
 300   L1, Col 1 • Spaces: 2 History ⏪
 301   L1, Col 1 • Spaces: 2 History ⏪
 302   L1, Col 1 • Spaces: 2 History ⏪
 303   L1, Col 1 • Spaces: 2 History ⏪
 304   L1, Col 1 • Spaces: 2 History ⏪
 305   L1, Col 1 • Spaces: 2 History ⏪
 306   L1, Col 1 • Spaces: 2 History ⏪
 307   L1, Col 1 • Spaces: 2 History ⏪
 308   L1, Col 1 • Spaces: 2 History ⏪
 309   L1, Col 1 • Spaces: 2 History ⏪
 310   L1, Col 1 • Spaces: 2 History ⏪
 311   L1, Col 1 • Spaces: 2 History ⏪
 312   L1, Col 1 • Spaces: 2 History ⏪
 313   L1, Col 1 • Spaces: 2 History ⏪
 314   L1, Col 1 • Spaces: 2 History ⏪
 315   L1, Col 1 • Spaces: 2 History ⏪
 316   L1, Col 1 • Spaces: 2 History ⏪
 317   L1, Col 1 • Spaces: 2 History ⏪
 318   L1, Col 1 • Spaces: 2 History ⏪
 319   L1, Col 1 • Spaces: 2 History ⏪
 320   L1, Col 1 • Spaces: 2 History ⏪
 321   L1, Col 1 • Spaces: 2 History ⏪
 322   L1, Col 1 • Spaces: 2 History ⏪
 323   L1, Col 1 • Spaces: 2 History ⏪
 324   L1, Col 1 • Spaces: 2 History ⏪
 325   L1, Col 1 • Spaces: 2 History ⏪
 326   L1, Col 1 • Spaces: 2 History ⏪
 327   L1, Col 1 • Spaces: 2 History ⏪
 328   L1, Col 1 • Spaces: 2 History ⏪
 329   L1, Col 1 • Spaces: 2 History ⏪
 330   L1, Col 1 • Spaces: 2 History ⏪
 331   L1, Col 1 • Spaces: 2 History ⏪
 332   L1, Col 1 • Spaces: 2 History ⏪
 333   L1, Col 1 • Spaces: 2 History ⏪
 334   L1, Col 1 • Spaces: 2 History ⏪
 335   L1, Col 1 • Spaces: 2 History ⏪
 336   L1, Col 1 • Spaces: 2 History ⏪
 337   L1, Col 1 • Spaces: 2 History ⏪
 338   L1, Col 1 • Spaces: 2 History ⏪
 339   L1, Col 1 • Spaces: 2 History ⏪
 340   L1, Col 1 • Spaces: 2 History ⏪
 341   L1, Col 1 • Spaces: 2 History ⏪
 342   L1, Col 1 • Spaces: 2 History ⏪
 343   L1, Col 1 • Spaces: 2 History ⏪
 344   L1, Col 1 • Spaces: 2 History ⏪
 345   L1, Col 1 • Spaces: 2 History ⏪
 346   L1, Col 1 • Spaces: 2 History ⏪
 347   L1, Col 1 • Spaces: 2 History ⏪
 348   L1, Col 1 • Spaces: 2 History ⏪
 349   L1, Col 1 • Spaces: 2 History ⏪
 350   L1, Col 1 • Spaces: 2 History ⏪
 351   L1, Col 1 • Spaces: 2 History ⏪
 352   L1, Col 1 • Spaces: 2 History ⏪
 353   L1, Col 1 • Spaces: 2 History ⏪
 354   L1, Col 1 • Spaces: 2 History ⏪
 355   L1, Col 1 • Spaces: 2 History ⏪
 356   L1, Col 1 • Spaces: 2 History ⏪
 357   L1, Col 1 • Spaces: 2 History ⏪
 358   L1, Col 1 • Spaces: 2 History ⏪
 359   L1, Col 1 • Spaces: 2 History ⏪
 360   L1, Col 1 • Spaces: 2 History ⏪
 361   L1, Col 1 • Spaces: 2 History ⏪
 362   L1, Col 1 • Spaces: 2 History ⏪
 363   L1, Col 1 • Spaces: 2 History ⏪
 364   L1, Col 1 • Spaces: 2 History ⏪
 365   L1, Col 1 • Spaces: 2 History ⏪
 366   L1, Col 1 • Spaces: 2 History ⏪
 367   L1, Col 1 • Spaces: 2 History ⏪
 368   L1, Col 1 • Spaces: 2 History ⏪
 369   L1, Col 1 • Spaces: 2 History ⏪
 370   L1, Col 1 • Spaces: 2 History ⏪
 371   L1, Col 1 • Spaces: 2 History ⏪
 372   L1, Col 1 • Spaces: 2 History ⏪
 373   L1, Col 1 • Spaces: 2 History ⏪
 374   L1, Col 1 • Spaces: 2 History ⏪
 375   L1, Col 1 • Spaces: 2 History ⏪
 376   L1, Col 1 • Spaces: 2 History ⏪
 377   L1, Col 1 • Spaces: 2 History ⏪
 378   L1, Col 1 • Spaces: 2 History ⏪
 379   L1, Col 1 • Spaces: 2 History ⏪
 380   L1, Col 1 • Spaces: 2 History ⏪
 381   L1, Col 1 • Spaces: 2 History ⏪
 382   L1, Col 1 • Spaces: 2 History ⏪
 383   L1, Col 1 • Spaces: 2 History ⏪
 384   L1, Col 1 • Spaces: 2 History ⏪
 385   L1, Col 1 • Spaces: 2 History ⏪
 386   L1, Col 1 • Spaces: 2 History ⏪
 387   L1, Col 1 • Spaces: 2 History ⏪
 388   L1, Col 1 • Spaces: 2 History ⏪
 389   L1, Col 1 • Spaces: 2 History ⏪
 390   L1, Col 1 • Spaces: 2 History ⏪
 391   L1, Col 1 • Spaces: 2 History ⏪
 392   L1, Col 1 • Spaces: 2 History ⏪
 393   L1, Col 1 • Spaces: 2 History ⏪
 394   L1, Col 1 • Spaces: 2 History ⏪
 395   L1, Col 1 • Spaces: 2 History ⏪
 396   L1, Col 1 • Spaces: 2 History ⏪
 397   L1, Col 1 • Spaces: 2 History ⏪
 398   L1, Col 1 • Spaces: 2 History ⏪
 399   L1, Col 1 • Spaces: 2 History ⏪
 400   L1, Col 1 • Spaces: 2 History ⏪
 401   L1, Col 1 • Spaces: 2 History ⏪
 402   L1, Col 1 • Spaces: 2 History ⏪
 403   L1, Col 1 • Spaces: 2 History ⏪
 404   L1, Col 1 • Spaces: 2 History ⏪
 405   L1, Col 1 • Spaces: 2 History ⏪
 406   L1, Col 1 • Spaces: 2 History ⏪
 407   L1, Col 1 • Spaces: 2 History ⏪
 408   L1, Col 1 • Spaces: 2 History ⏪
 409   L1, Col 1 • Spaces: 2 History ⏪
 410   L1, Col 1 • Spaces: 2 History ⏪
 411   L1, Col 1 • Spaces: 2 History ⏪
 412   L1, Col 1 • Spaces: 2 History ⏪
 413   L1, Col 1 • Spaces: 2 History ⏪
 414   L1, Col 1 • Spaces: 2 History ⏪
 415   L1, Col 1 • Spaces: 2 History ⏪
 416   L1, Col 1 • Spaces: 2 History ⏪
 417   L1, Col 1 • Spaces: 2 History ⏪
 418   L1, Col 1 • Spaces: 2 History ⏪
 419   L1, Col 1 • Spaces: 2 History ⏪
 420   L1, Col 1 • Spaces: 2 History ⏪
 421   L1, Col 1 • Spaces: 2 History ⏪
 422   L1, Col 1 • Spaces: 2 History ⏪
 423   L1, Col 1 • Spaces: 2 History ⏪
 424   L1, Col 1 • Spaces: 2 History ⏪
 425   L1, Col 1 • Spaces: 2 History ⏪
 426   L1, Col 1 • Spaces: 2 History ⏪
 427   L1, Col 1 • Spaces: 2 History ⏪
 428   L1, Col 1 • Spaces: 2 History ⏪
 429   L1, Col 1 • Spaces: 2 History ⏪
 430   L1, Col 1 • Spaces: 2 History ⏪
 431   L1, Col 1 • Spaces: 2 History ⏪
 432   L1, Col 1 • Spaces: 2 History ⏪
 433   L1, Col 1 • Spaces: 2 History ⏪
 434   L1, Col 1 • Spaces: 2 History ⏪
 435   L1, Col 1 • Spaces: 2 History ⏪
 436   L1, Col 1 • Spaces: 2 History ⏪
 437   L1, Col 1 • Spaces: 2 History ⏪
 438   L1, Col 1 • Spaces: 2 History ⏪
 439   L1, Col 1 • Spaces: 2 History ⏪
 440   L1, Col 1 • Spaces: 2 History ⏪
 441   L1, Col 1 • Spaces: 2 History ⏪
 442   L1, Col 1 • Spaces: 2 History ⏪
 443   L1, Col 1 • Spaces: 2 History ⏪
 444   L1, Col 1 • Spaces: 2 History ⏪
 445   L1, Col 1 • Spaces: 2 History ⏪
 446   L1, Col 1 • Spaces: 2 History ⏪
 447   L1, Col 1 • Spaces: 2 History ⏪
 448   L1, Col 1 • Spaces: 2 History ⏪
 449   L1, Col 1 • Spaces: 2 History ⏪
 450   L1, Col 1 • Spaces: 2 History ⏪
 451   L1, Col 1 • Spaces: 2 History ⏪
 452   L1, Col 1 • Spaces: 2 History ⏪
 453   L1, Col 1 • Spaces: 2 History ⏪
 454   L1, Col 1 • Spaces: 2 History ⏪
 455   L1, Col 1 • Spaces: 2 History ⏪
 456   L1, Col 1 • Spaces: 2 History ⏪
 457   L1, Col 1 • Spaces: 2 History ⏪
 458   L1, Col 1 • Spaces: 2 History ⏪
 459   L1, Col 1 • Spaces: 2 History ⏪
 460   L1, Col 1 • Spaces: 2 History ⏪
 461   L1, Col 1 • Spaces: 2 History ⏪
 462   L1, Col 1 • Spaces: 2 History ⏪
 463   L1, Col 1 • Spaces: 2 History ⏪
 464   L1, Col 1 • Spaces: 2 History ⏪
 465   L1, Col 1 • Spaces: 2 History ⏪
 466   L1, Col 1 • Spaces: 2 History ⏪
 467   L1, Col 1 • Spaces: 2 History ⏪
 468   L1, Col 1 • Spaces: 2 History ⏪
 469   L1, Col 1 • Spaces: 2 History ⏪
 470   L1, Col 1 • Spaces: 2 History ⏪
 471   L1, Col 1 • Spaces: 2 History ⏪
 472   L1, Col 1 • Spaces: 2 History ⏪
 473   L1, Col 1 • Spaces: 2 History ⏪
 474   L1, Col 1 • Spaces: 2 History ⏪
 475   L1, Col 1 • Spaces: 2 History ⏪
 476   L1, Col 1 • Spaces: 2 History ⏪
 477   L1, Col 1 • Spaces: 2 History ⏪
 478   L1, Col 1 • Spaces: 2 History ⏪
 479   L1, Col 1 • Spaces: 2 History ⏪
 480   L1, Col 1 • Spaces: 2 History ⏪
 481   L1, Col 1 • Spaces: 2 History ⏪
 482   L1, Col 1 • Spaces: 2 History ⏪
 483   L1, Col 1 • Spaces: 2 History ⏪
 484   L1, Col 1 • Spaces: 2 History ⏪
 485   L1, Col 1 • Spaces: 2 History ⏪
 486   L1, Col 1 • Spaces: 2 History ⏪
 487   L1, Col 1 • Spaces: 2 History ⏪
 488   L1, Col 1 • Spaces: 2 History ⏪
 489   L1, Col 1 • Spaces: 2 History ⏪
 490   L1, Col 1 • Spaces: 2 History ⏪
 491   L1, Col 1 • Spaces: 2 History ⏪
 492   L1, Col 1 • Spaces: 2 History ⏪
 493   L1, Col 1 • Spaces: 2 History ⏪
 494   L1, Col 1 • Spaces: 2 History ⏪
 495   L1, Col 1 • Spaces: 2 History ⏪
 496   L1, Col 1 • Spaces: 2 History ⏪
 497   L1, Col 1 • Spaces: 2 History ⏪
 498   L1, Col 1 • Spaces: 2 History ⏪
 499   L1, Col 1 • Spaces: 2 History ⏪
 500   L1, Col 1 • Spaces: 2 History ⏪
 501   L1, Col 1 • Spaces: 2 History ⏪
 502   L1, Col 1 • Spaces: 2 History ⏪
 503   L1, Col 1 • Spaces: 2 History ⏪
 504   L1, Col 1 • Spaces: 2 History ⏪
 505   L1, Col 1 • Spaces: 2 History ⏪
 506   L1, Col 1 • Spaces: 2 History ⏪
 507   L1, Col 1 • Spaces: 2 History ⏪
 508   L1, Col 1 • Spaces: 2 History ⏪
 509   L1, Col 1 • Spaces: 2 History ⏪
 510   L1, Col 1 • Spaces: 2 History ⏪
 511   L1, Col 1 • Spaces: 2 History ⏪
 512   L1, Col 1 • Spaces: 2 History ⏪
 513   L1, Col 1 • Spaces: 2 History ⏪
 514   L1, Col 1 • Spaces: 2 History ⏪
 515   L1, Col 1 • Spaces: 2 History ⏪
 516   L1, Col 1 • Spaces: 2 History ⏪
 517   L1, Col 1 • Spaces: 2 History ⏪
 518   L1, Col 1 • Spaces: 2 History ⏪
 519   L1, Col 1 • Spaces: 2 History ⏪
 520   L1, Col 1 • Spaces: 2 History ⏪
 521   L1, Col 1 • Spaces: 2 History ⏪
 522   L1, Col 1 • Spaces: 2 History ⏪
 523   L1, Col 1 • Spaces: 2 History ⏪
 524   L1, Col 1 • Spaces: 2 History ⏪
 525   L1, Col 1 • Spaces: 2 History ⏪
 526   L1, Col 1 • Spaces: 2 History ⏪
 527   L1, Col 1 • Spaces: 2 History ⏪
 528   L1, Col 1 • Spaces: 2 History ⏪
 529   L1, Col 1 • Spaces: 2 History ⏪
 530   L1, Col 1 • Spaces: 2 History ⏪
 531   L1, Col 1 • Spaces: 2 History ⏪
 532   L1, Col 1 • Spaces: 2 History ⏪
 533   L1, Col 1 • Spaces: 2 History ⏪
 534   L1, Col 1 • Spaces: 2 History ⏪
 535   L1, Col 1 • Spaces: 2 History ⏪
 536   L1, Col 1 • Spaces: 2 History ⏪
 537   L1, Col 1 • Spaces: 2 History ⏪
 538   L1, Col 1 • Spaces: 2 History ⏪
 539   L1, Col 1 • Spaces: 2 History ⏪
 540   L1, Col 1 • Spaces: 2 History ⏪
 541   L1, Col 1 • Spaces: 2 History ⏪
 542   L1, Col 1 • Spaces: 2 History ⏪
 543   L1, Col 1 • Spaces: 2 History ⏪
 544   L1, Col 1 • Spaces: 2 History ⏪
 545   L1, Col 1 • Spaces: 2 History ⏪
 546   L1, Col 1 • Spaces: 2 History ⏪
 547   L1, Col 1 • Spaces: 2 History ⏪
 548   L1, Col 1 • Spaces: 2 History ⏪
 549   L1, Col 1 • Spaces: 2 History ⏪
 550   L1, Col 1 • Spaces: 2 History ⏪
 551   L1, Col 1 • Spaces: 2 History ⏪
 552   L1, Col 1 • Spaces: 2 History ⏪
 553   L1, Col 1 • Spaces: 2 History ⏪
 554   L1, Col 1 • Spaces: 2 History ⏪
 555   L1, Col 1 • Spaces: 2 History ⏪
 556   L1, Col 1 • Spaces: 2 History ⏪
 557   L1, Col 1 • Spaces: 2 History ⏪
 558   L1, Col 1 • Spaces: 2 History ⏪
 559   L1, Col 1 • Spaces: 2 History ⏪
 560   L1, Col 1 • Spaces: 2 History ⏪
 561   L1, Col 1 • Spaces: 2 History ⏪
 562   L1, Col 1 • Spaces: 2 History ⏪
 563   L1, Col 1 • Spaces: 2 History ⏪
 564   L1, Col 1 • Spaces: 2 History ⏪
 565   L1, Col 1 • Spaces: 2 History ⏪
 566   L1, Col 1 • Spaces: 2 History ⏪
 567   L1, Col 1 • Spaces: 2 History ⏪
 568   L1, Col 1 • Spaces: 2 History ⏪
 569   L1, Col 1 • Spaces: 2 History ⏪
 570   L1, Col 1 • Spaces: 2 History ⏪
 571   L1, Col 1 • Spaces: 2 History ⏪
 572   L1, Col 1 • Spaces: 2 History ⏪
 573   L1, Col 1 • Spaces: 2 History ⏪
 574   L1, Col 1 • Spaces: 2 History ⏪
 575   L1, Col 1 • Spaces: 2 History ⏪
 576   L1, Col 1 • Spaces: 2 History ⏪
 577   L1, Col 1 • Spaces: 2 History ⏪
 578   L1, Col 1 • Spaces: 2 History ⏪
 579   L1, Col 1 • Spaces: 2 History ⏪
 580   L1, Col 1 • Spaces: 2 History ⏪
 581   L1, Col 1 • Spaces: 2 History ⏪
 582   L1, Col 1 • Spaces: 2 History ⏪
 583   L1, Col 1 • Spaces: 2 History ⏪
 584   L1, Col 1 • Spaces: 2 History ⏪
 585   L1, Col 1 • Spaces: 2 History ⏪
 586   L1, Col 1 • Spaces: 2 History ⏪
 587   L1, Col 1 • Spaces: 2 History ⏪
 588   L1, Col 1 • Spaces: 2 History ⏪
 589   L1, Col 1 • Spaces: 2 History ⏪
 590   L1, Col 1 • Spaces: 2 History ⏪
 591   L1, Col 1 • Spaces: 2 History ⏪
 592   L1, Col 1 • Spaces: 2 History ⏪
 593   L1, Col 1 • Spaces: 2 History ⏪
 594   L1, Col 1 • Spaces: 2 History ⏪
 595   L1, Col 1 • Spaces: 2 History ⏪
 596   L1, Col 1 • Spaces: 2 History ⏪
 597   L1, Col 1 • Spaces: 2 History ⏪
 598   L1, Col 1 • Spaces: 2 History ⏪
 599   L1, Col 1 • Spaces: 2 History ⏪
 600   L1, Col 1 • Spaces: 2 History ⏪
 601   L1, Col 1 • Spaces: 2 History ⏪
 602   L1, Col 1 • Spaces: 2 History ⏪
 603   L1, Col 1 • Spaces: 2 History ⏪
 604   L1, Col 1 • Spaces: 2 History ⏪
 605   L1, Col 1 • Spaces: 2 History ⏪
 606   L1, Col 1 • Spaces: 2 History ⏪
 607   L1, Col 1 • Spaces: 2 History ⏪
 608   L1, Col 1 • Spaces: 2 History ⏪
 609   L1, Col 1 • Spaces: 2 History ⏪
 610   L1, Col 1 • Spaces: 2 History ⏪
 611   L1, Col 1 • Spaces: 2 History ⏪
 612   L1, Col 1 • Spaces: 2 History ⏪
 613   L1, Col 1 • Spaces: 2 History ⏪
 614   L1, Col 1 • Spaces: 2 History ⏪
 615   L1, Col 1 • Spaces: 2 History ⏪
 616   L1, Col 1 • Spaces: 2 History ⏪
 617   L1, Col 1 • Spaces: 2 History ⏪
 618   L1, Col 1 • Spaces: 2 History ⏪
 619   L1, Col 1 • Spaces: 2 History ⏪
 620   L1, Col 1 • Spaces: 2 History ⏪
 621   L1, Col 1 • Spaces: 2 History ⏪
 622   L1, Col 1 • Spaces: 2 History ⏪
 623   L1, Col 1 • Spaces: 2 History ⏪
 624   L1, Col 1 • Spaces: 2 History ⏪
 625   L1, Col 1 • Spaces: 2 History ⏪
 626   L1, Col 1 • Spaces: 2 History ⏪
 627   L1, Col 1 • Spaces: 2 History ⏪
 628   L1, Col 1 • Spaces: 2 History ⏪
 629   L1, Col 1 • Spaces: 2 History ⏪
 630   L1, Col 1 • Spaces: 2 History ⏪
 631   L1, Col 1 • Spaces: 2 History ⏪
 632   L1, Col 1 • Spaces: 2 History ⏪
 633   L1, Col 1 • Spaces: 2 History ⏪
 634   L1, Col 1 • Spaces: 2 History ⏪
 635   L1, Col 1 • Spaces: 2 History ⏪
 636   L1, Col 1 • Spaces: 2 History ⏪
 637  
```

```

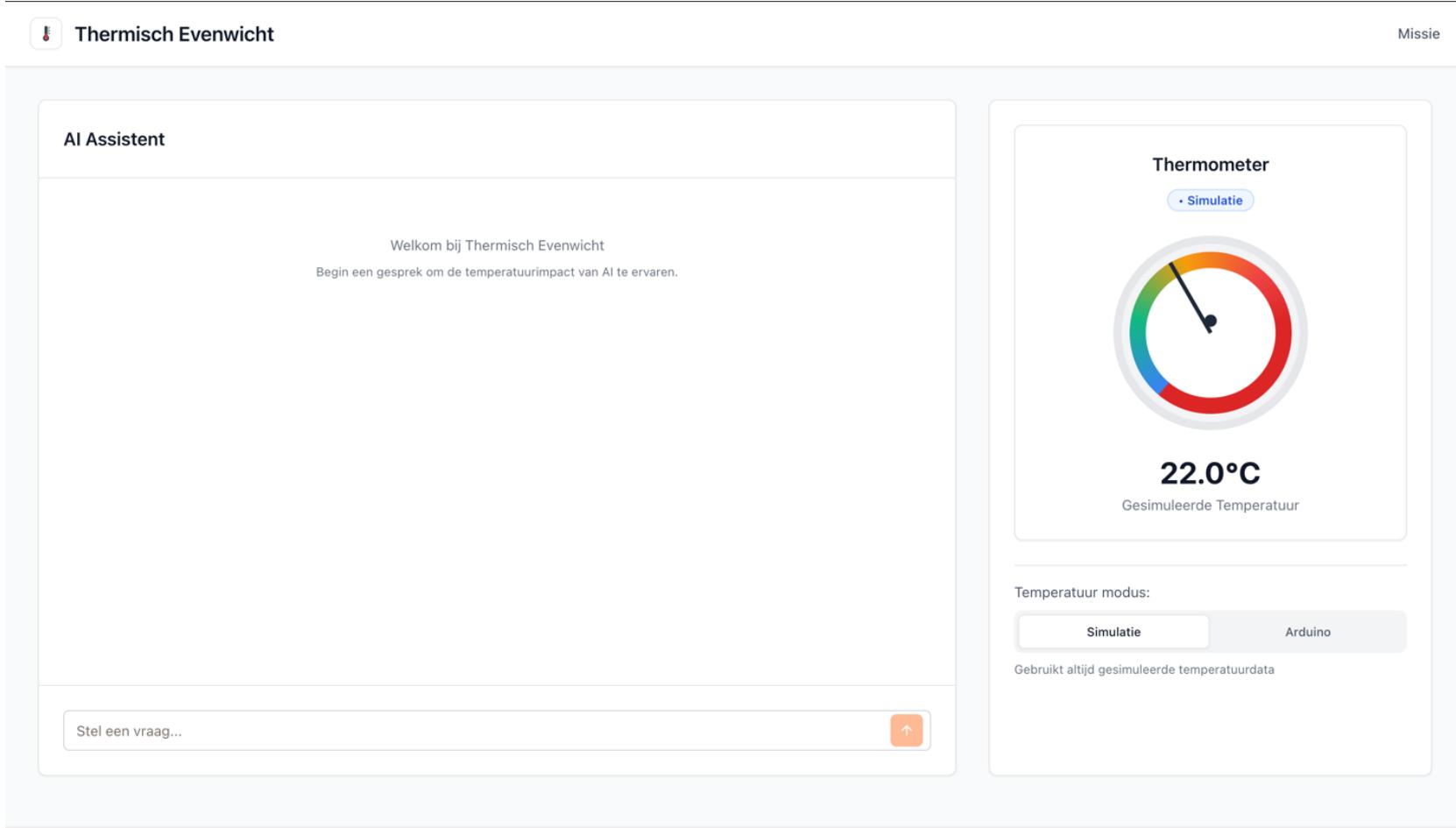
Format arduino > local-arduino-bridge.js ...
229     }, 3000); // Check every 3 seconds
230 }
231
232 // Arduino serial connection
233 let port = null;
234 let parser = null;
235
236 async function connectToArduino() {
237     try {
238         // Auto-detect port if not specified
239         if (!portName) {
240             portName = await findArduinoPort();
241         }
242
243         console.log(`Attempting to connect to Arduino on port ${portName}`);
244
245         port = new SerialPort({
246             path: portName,
247             baudRate: BAUD_RATE,
248             autoOpen: false,
249         });
250
251         parser = port.pipe(new ReadlineParser({ delimiter: "\n" }));
252
253         // Open the port
254         await new Promise((resolve, reject) => {
255             port.open((err) => {
256                 if (err) reject(err);
257                 else resolve();
258             });
259         });
260
261         console.log("Connected to Arduino");
262         console.log(`Listening for temperature data...`);
263         console.log("");
264
265         // Listen for data from Arduino
266         parser.on("data", (data) => {
267             try {
268                 const trimmed = data.toString().trim();
269                 console.log(`Arduino: ${trimmed}`);
270
271                 // Parse temperature data
272                 if (trimmed.startsWith("TEMP:")) {
273                     const tempValue = parseFloat(trimmed.substring(5));
274                     if (!isNaN(tempValue)) {
275                         const tempData = {
276                             type: "temperature",
277                             value: tempValue,
278                             timestamp: new Date().toISOString(),
279                         };
280
281                         console.log(`Temperature: ${tempValue}°C`);
282
283                         // Send to Replit app via WebSocket
284                         if (ws && ws.readyState === WebSocket.OPEN) {
285                             ws.send(JSON.stringify(tempData));
286                             console.log(`Sent to Replit: ${tempValue}°C`);
287                         } else {
288                             console.log(`WebSocket not connected - data not sent`);
289                         }
290                     }
291                 }
292
293                 // Handle other Arduino responses
294                 else if (
295                     trimmed.includes("Heater intensity set") ||
296                     trimmed.includes("Emergency") ||
297                     trimmed.includes("WARNING")
298                 ) {
299                     console.log(`Heater status: ${trimmed}`);
300                 } else if (
301                     trimmed.includes("LED") ||
302                     trimmed.includes("Animation") ||
303                     trimmed.includes("brightness")
304                 ) {
305                     console.log(`LED status: ${trimmed}`);
306                 } catch (error) {
307                     console.error(`Error parsing Arduino data: ${error.message}`);
308                     console.log(`Raw data: ${trimmed}`);
309                 }
310             });
311
312             port.on("error", (error) => {
313                 console.error(`Arduino serial port error: ${error.message}`);
314             });
315
316             port.on("close", () => {
317                 console.log(`Arduino connection closed`);
318             });
319         } catch (error) {
320             console.error(`Failed to connect to Arduino: ${error.message}`);
321             console.log("");
322
323             // Show available ports for troubleshooting
324             try {
325                 await findArduinoPort();
326             } catch (portError) {
327                 console.error(`Port detection error: ${portError.message}`);
328             }
329
330             console.log("");
331             console.log(`Troubleshooting Tips:`);
332             console.log(`1. Make sure your Arduino is connected via USB`);
333             console.log(`2. Check if Arduino IDE can see the device`);
334             console.log(`3. Try a different USB cable or port`);
335             console.log(`4. On Linux/Mac, you might need permissions: sudo chmod 666 /dev/ttyACM* /dev/ttyUSB*`);
336             console.log(`5. On Windows, check Device Manager for COM ports`);
337
338             console.log(`6. Try specifying port manually: node local-arduino-bridge.js COM3 (Windows)`);
339             console.log(`node local-arduino-bridge.js /dev/ttyACM0 (Linux/Mac)`);
340
341             process.exit(1);
342         }
343     }
344 }
345
346 // Start connections
347 async function start() {
348     // Check if REPLIT_URL has been updated
349     if (
350         REPLIT_URL.includes("your-repl-name") ||
351         REPLIT_URL.includes("your-username")
352     ) {
353         console.log(
354             `Please update the REPLIT_URL in the script with your actual Replit app URL`;
355     );
356     console.log("");
357     console.log(`To find your Replit WebSocket URL:`);
358     console.log(`1. Open your Replit project`);
359     console.log(`2. Look at the URL (e.g., https://my-project.username.replit.co/ws?`);
360
361     console.log(`3. Replace "https" with "wss" and add "/ws?`);
362     type=arduino`);
363     console.log(`4. Update REPLIT_URL in this script`);
364     console.log("");
365     console.log(`Example: wss://thermal-equilibrium.myusername.replit.co/ws?`);
366     type=arduino`;
367     );
368     process.exit(1);
369 }
370
371 // Connect to both Arduino and Replit
372 await connectToArduino();
373 connectToReplit();
374
375
376 // Graceful shutdown
377 process.on("SIGINT", () => {
378     console.log(`Shutting down Local Arduino Bridge...`);
379
380     if (port && port.isOpen) {
381         port.close();
382     }
383
384     if (ws) {
385         ws.close();
386     }
387
388     console.log(`Bridge stopped`);
389     process.exit(0);
390 });
391
392 // Handle uncaught errors
393 process.on("uncaughtException", (error) => {
394     console.error(`Uncaught error: ${error.message}`);
395     process.exit(1);
396 });
397
398 // Start the bridge
399 start().catch(console.error);
400

```

Figuur 39: Bridge file

CONCEPTEN EMBODIED EXPERIENCE

12



© 2025 Thermisch Evenwicht Project
Een belichaamde ervaring van AI's milieueffect

Figuur 40: De interface ontworpen voor de aansturing van de "Thermisch Evenwicht" installatie.

Interface:

Om de installatie aan te sturen moet er natuurlijk ook een applicatie zijn die het systeem aanstuurt. Hiervoor is een interface gemaakt waarmee de gebruiker zijn/haar meest voorkomende AI taken kan uitvoeren. Verder is het interface uitgerust met; een intuïtieve gebruikersinterface, KI-assistent, real-time temperatuurvisualisatie, indicatieknop voor KI-modelstatus en kan het model gebruikt worden voor tekstgeneratie en -analyse.
KI-toepassingen: Beeldgeneratie, tekstgeneratie en -analyse.

Experimenteren:

Ontwikkeling van de installatie:

Voor de installatie "Thermisch Evenwicht" was het cruciaal om een effectief thermisch systeem te ontwikkelen dat de gebruiker een overtuigende embodied experience kon bieden. Het doel was om de cabine merkbaar en snel genoeg te kunnen verwarmen als reactie op AI-gebruik. Dit bleek een technische uitdaging, die in verschillende fases is aangepakt.

Experiment 1: De Basisopstelling

In de eerste opstelling werd er gebruikgemaakt van twee grote verwarmingsfolie-elementen. Deze werden strategisch in de twee zijpanelen van de installatie verwerkt. De hypothese was dat de warmte van de zijkanten een omhullend gevoel zou creëren.

Resultaat: Helaas bleek deze aanpak onvoldoende. De warmteontwikkeling was te traag en de maximale temperatuur te laag om de cabine effectief te verwarmen. De embodied experience bleef uit; de gebruiker voelde nauwelijks een verandering, waardoor de koppeling met het AI-gebruik verloren ging.

Experiment 2: Verhoogde Warmte-input

Om de warmte-input te verhogen, werd besloten een derde verwarmingsfolie-element toe te voegen. Dit element werd in het achterpaneel geplaatst, recht tegenover de gebruiker. Het idee was dat de directe stralingswarmte vanaf de achterkant, gecombineerd met de warmte van de zijkanten, voor een intensere ervaring zou zorgen.

Resultaat: Hoewel de totale warmteafgifte toenam, was het effect nog steeds niet krachtig genoeg. De cabine warmde iets sneller op, maar de sensatie was te subtiel en te traag om een directe en overtuigende feedbackloop te vormen. De embodied experience was nog steeds niet het impactvolle, confronterende gevoel dat het concept vereiste.

Experiment 3: Toevoeging van Luchtverwarming

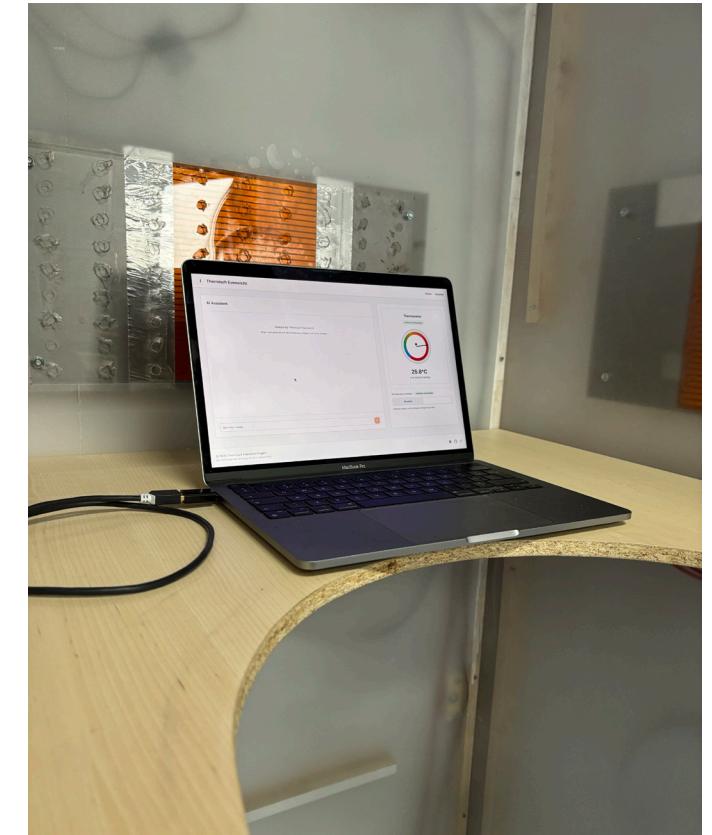
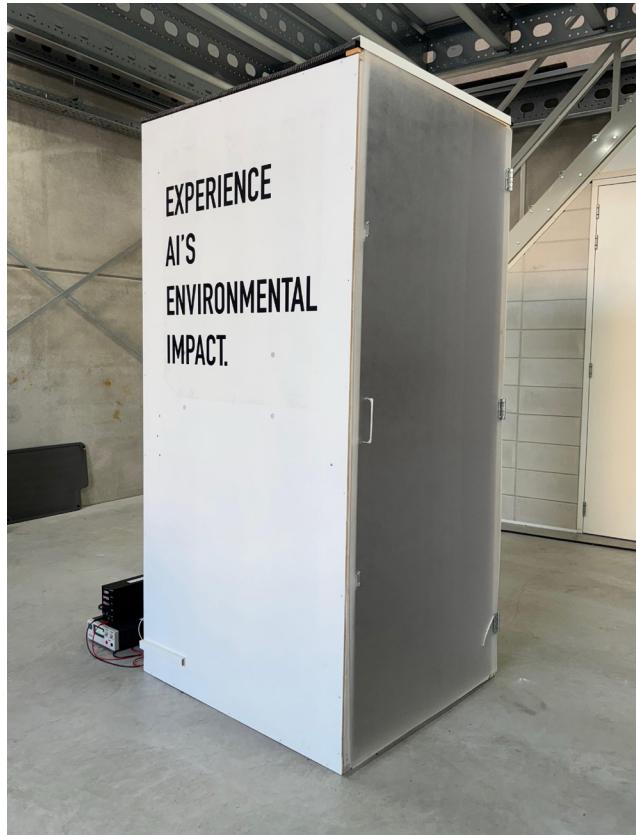
Omdat stralingswarmte alleen niet voldeed, werd als laatste poging een zelfgebouwde luchtverwarmer aan het systeem toegevoegd. Dit element was ontworpen om de warme lucht actief door de cabine te circuleren, met als doel de temperatuur sneller en gelijkmatiger te laten stijgen.

Resultaat: Ondanks de toevoeging van convectiewarmte bleef het eindresultaat onder de maat. De installatie werd weliswaar warmer, maar de ervaring voelde niet langer als een directe, beklemmende reactie op de AI-interactie. Het werd meer een algemene, langzame verandering van de omgevingstemperatuur. Het experiment was daardoor nog steeds niet overtuigend genoeg in het overbrengen van de beoogde fysieke sensatie.

Conclusie van de Experimenten:

De experimenten met de installatie maakten duidelijk dat het creëren van een snelle en intense embodied experience in een grotere, geïsoleerde ruimte technisch complex is met de beschikbare materialen. De traagheid van het systeem zorgde ervoor dat de directe koppeling tussen de actie van de gebruiker en de fysieke reactie verloren ging. Deze belangrijke leerervaring was de aanleiding om de focus te verleggen naar meer directe vormen van thermische feedback.

De installatie:



Figuur 41,42,43: De voltooide fysieke installatie van de "Thermisch Evenwicht" cabine.

Ambitie & Uitdaging:

De ambitie om een installatie te creëren die een daadwerkelijke werkplek nabootst, waar gebruikers normaal met KI interacteren, was een boeiende en leerzame uitdaging. Het idee was om de abstracte impact van KI op een concrete en immersieve manier te laten ervaren. Echter, tijdens de ontwikkeling liep ik tegen aanzienlijke technische tegenslagen aan.

De verwarmingscomponenten bleken simpelweg niet krachtig genoeg om de cabine naar de gewenste temperatuur te verwarmen. Hierdoor was de beoogde embodied experience behoorlijk afgezwakt en miste het prototype zijn oorspronkelijke doel om een overtuigende fysieke vertaling van KI's energieverbruik te bieden.

Leerpunten & Vervolg:

Deze ervaring was een belangrijke les in de technische haalbaarheid van ambitieuze concepten. Ik heb hiervan geleerd dat een directe, krachtige fysieke feedback essentieel is voor een overtuigende embodied experience. Dit inzicht heeft me gemotiveerd om verder te experimenteren met alternatieve, meer directe en haalbare methoden voor thermische feedback, zoals de "Heated Keyboard" en "Heated Vest" experimenten, om zo dichter bij het oorspronkelijke doel te komen.

CONCEPTEN EMBODIED EXPERIENCE

12

STORYTELLING

Een goed productontwerp begint met een goed verhaal. Dit verhaal helpt mij om de setting te bepalen: de precieze omgeving en het moment waarop het toekomstige product gebruikt zal worden. Het is geen fantasieverhaal, maar een verhaal gebaseerd op de realiteit van mijn doelgroep.

De setting: de moderne, digitale werkplek

Uit mijn onderzoek, zoals de enquête, blijkt een heel duidelijk beeld: de doelgroep gebruikt AI intensief en vooral voor serieuze zaken. Denk hierbij aan:

- Studenten die lange verslagen schrijven en AI gebruiken om informatie samen te vatten of hun tekst te verbeteren.
- Professionals die data analyseren, presentaties voorbereiden of complexe e-mails opstellen.
- Creatievelingen die AI inzetten voor het genereren van ideeën, afbeeldingen of code.

Hun werkplek is dus een flexible, dynamische en digitale kantooromgeving. Ze schakelen constant tussen verschillende programma's en apps. AI is hierin geen losse tool, maar een vaste partner die hen helpt sneller en slimmer te werken.

Ontwerpeisen voor het prototype

Met deze kennis in mijn achterhoofd, moet het ontwerp een toevoeging zijn in de bestaande werkroutes van de doelgroep. Het product mag niet onopgemerkt blijven. Het doel zal moeten zijn om de gebruiker op een fysieke en visuele manier te confronteren met de impact van zijn of haar AI-gebruik.

Het is daarom een bewuste ontwerpkeuze dat de gebruiker hierdoor even uit de flow of concentratie gehaald kan worden. Deze onderbreking moet echter niet als een storende fout voelen, maar als een betekenisvolle pauze die aanzet tot reflectie. Het moet een moment creëren waarop de onzichtbare, digitale activiteit wordt vertaald naar een tastbare, fysieke sensatie.

Het zal direct moeten werken in combinatie met de tools die de doelgroep al gebruikt, maar het moet een nieuwe, voelbare dimensie toevoegen. Het doel is niet om hun gedrag met regels te veranderen, maar om hun bestaande gedrag te verrijken met een nieuwe laag van bewustzijn, zodat ze zelf bewuster kunnen handelen.

VAN CONCEPT NAAR CONCREET PRODUCT: DE MUISMAT

Hoe creeér ik een meetbaar prototype voor mijn doelgroep?

De Keuze voor Directe Contactwarmte

Uit de eerdere concepten, met name het "Thermisch Evenwicht (Installatie)" experiment, is een cruciale les geleerd: stralingswarmte in een open ruimte is te traag en te indirect om de gewenste, directe embodied experience te verkrijgen. De koppeling tussen de actie van de gebruiker en de fysieke sensatie ging verloren. Daarom is de strategische keuze gemaakt om terug te keren naar directe contactwarmte, waarbij het warmte-element fysiek contact maakt met het lichaam van de gebruiker. Dit garandeert een snelle, onmiskenbare en persoonlijke feedback.

De Context: De Werkplek van de Doelgroep

Zoals beschreven in het hoofdstuk "Storytelling", moet het product functioneren in de bestaande werkomgeving van de doelgroep: een (thuis)kantoor, achter een bureau. Dit inzicht leidt tot de logische keuze voor een muismat als drager van de technologie. Een muismat is een object dat al vanzelfsprekend aanwezig is in deze setting. Het vereist geen nieuwe handeling of aanpassing van de gebruiker.

De keuze voor de muismat hangt ook nauw samen met de manier waarop de embodied experience wordt overgebracht. De onderarmen, die volgens een ergonomische werkhouding vaak op het bureau of de muismat rusten, zijn een ideaal contactpunt. De huid van de onderarmen is gevoelig voor temperatuurverschillen, en de constante aanwezigheid op de mat zorgt voor een continue mogelijkheid tot feedback.

Iteratieve Ontwikkeling van de Thermische Muismat

Om tot een effectief en comfortabel prototype te komen, zijn er verschillende versies van de thermische muismat ontwikkeld en getest.

Versie 1: Directe Blootstelling

- Opzet: In dit eerste, ruwe prototype werd de vorm van de onderarm direct uit een standaard muismat gesneden. De verwarmingsfolie werd hieronder geplaatst, waardoor de huid direct in contact kwam met de plastic folie wanneer deze opwarmde.
- Feedback & Analyse: Dit ontwerp faalde op het gebied van gebruikerscomfort. De feedback was unaniem negatief: de warmte werd veel te intens en geconcentreerd, wat bijna tot een branderig gevoel leidde. Daarnaast werd het plakkerige gevoel van het opwarmende plastic als zeer onprettig ervaren. De ervaring was niet confronterend, maar pijnlijk en afleidend, wat het doel voorbijschoot.

VAN CONCEPT NAAR CONCREET PRODUCT

Versie 2: Integratie onder de Oppervlaklaag

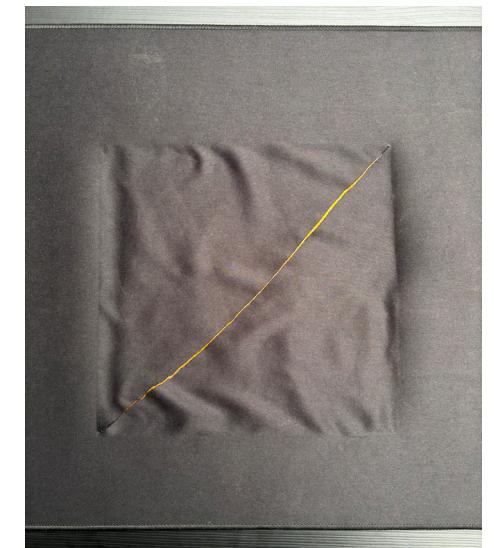
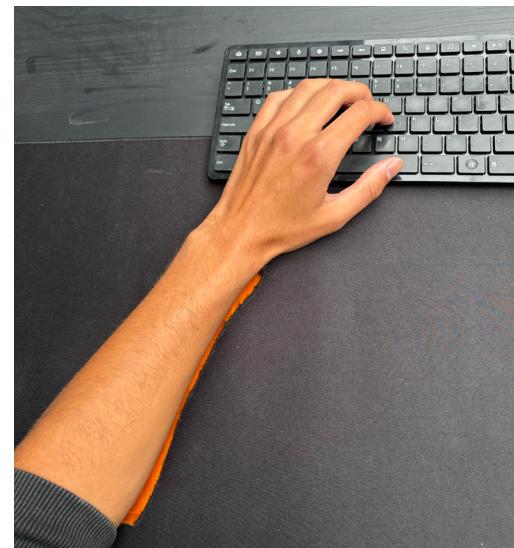
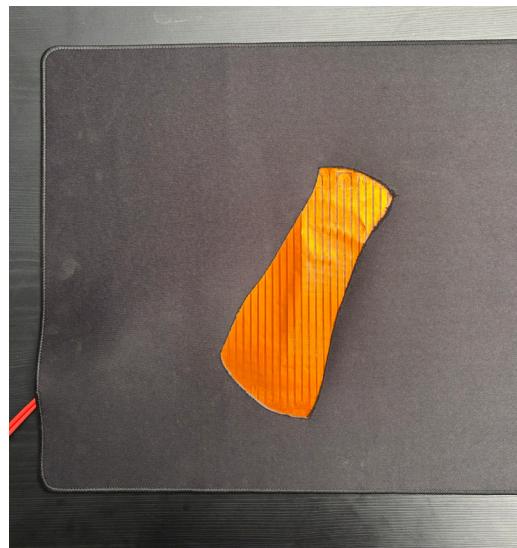
- Opzet: Om de problemen van Versie 1 op te lossen, werd de verwarmingsfolie nu onder de stoffen toplaag van de muismat geïntegreerd. Hiervoor werd de stoffen laag voorzichtig opengesneden, de folie geplaatst, en de laag vervolgens weer vastgelijmd.
- Feedback & Analyse: Dit was een significante verbetering. De warmte werd nu veel gelijkmatiger verspreid door de stoffen laag, wat zorgde voor een aangenamere en minder agressieve sensatie. Echter, de esthetiek en afwerking waren ondermaats. De lijmranden waren zichtbaar en de stof trok op sommige plekken krom, wat het prototype een onprofessionele en fragiele uitstraling gaf. Hoewel de functie beter was, faalde de vorm.

Versie 3: Naadloze Integratie van Achteren (Definitief Concept)

- Opzet: De laatste en meest succesvolle versie combineert de lessen uit de eerdere experimenten. In plaats van de toplaag open te snijden, werd de muismat nu vanaf de onderkant benaderd. Er werd een precieze uitsnede gemaakt in de foam-basis, waarna de verwarmingsfolie hierin werd verwerkt. Vervolgens werd het ontstane reliëf zorgvuldig opgevuld met het oorspronkelijke foam-materiaal dat was verwijderd.
- Feedback & Analyse: Deze aanpak bleek superieur op alle fronten. De warmte wordt, net als in versie 2, prettig en gelijkmatig door de onbeschadigde toplaag verspreid. Het cruciale verschil is dat de bovenkant van de muismat volledig intact en naadloos blijft. Er zijn geen lijmranden of oneffenheden zichtbaar, wat resulteert in een professioneel, strak en duurzaam eindproduct. Dit ontwerp is niet alleen functioneel en comfortabel, maar ook esthetisch hoogwaardig genoeg om als een overtuigend prototype te dienen. Dit werd de basis voor het definitieve ontwerp.

Conclusie van de Experimenten:

Door deze iteratieve stappen – van een ruw functioneel model (Versie 1), via een functioneel maar onafgewerkte model (Versie 2), naar een naadloos geïntegreerd ontwerp (Versie 3) – is een prototype ontwikkeld dat zowel een effectieve embodied experience levert als voldoet aan de esthetische en comforteisen voor een product dat thuis hoort op de moderne werkplek.



Figuur 44,45: De opzet van Versie 1, waarbij een uitsnede in de muismat wordt gevuld met de verwarmingsfolie voor direct contact met de onderarm.

Figuur 46,47: De constructie van Versie 2, waarbij de verwarmingsfolie onder de opengesneden toplaag wordt geplaatst en vervolgens wordt dichtgelijmd.

VAN CONCEPT NAAR CONCREET PRODUCT



Figuur 48: Het definitieve prototype van de thermische muismat (Versie 3), waarbij de verwarmingsfolie naadloos is geïntegreerd onder het oppervlak, wat resulteert in een strakke en professionele afwerking.

VAN CONCEPT NAAR CONCREET PRODUCT

CONCEPTEN INTERFACE

Nu de concepten om de ervaring over te brengen op het lichaam zijn gemaakt is het tijd om te gaan prototype met het interface. Wat moet de kernfunctie zijn? Hoe moet het eruit komen te zien? Waar moet de interface nog meer aan voldoen?

Uit de inzichten van het fieldresearch zien we dat veel gebruikers AI gebruiken voor hun werk en studie. Om een goed interface te bouwen moeten we weten met wat voor AI platform zij nu werken. Hiervoor heb ik een kleine analyse van bestaande AI platformen gedaan. Hier kwamen de volgende resultaten uit:

Analyse van bestaande AI platformen:

Voor de ontwikkeling van mijn eigen AI-platform, dat de milieu-impact van KI voelbaar maakt, is het essentieel om de huidige, populaire AI-tools te analyseren. De doelgroep (20-30 jaar) gebruikt deze platformen dagelijks voor studie en werk. Deze analyse richt zich op de interface, de gebruikerservaring en de kernelementen die ik kan overnemen of juist bewust kan aanpassen voor mijn eigen concept.

1. ChatGPT (OpenAI)

Beschrijving:

ChatGPT is een conversatie-AI, gebaseerd op een groot taalmodel (LLM). Gebruikers interacteren ermee via een chat-interface, waarin ze vragen stellen of opdrachten geven in natuurlijke taal.

Belangrijkste Elementen:

- Inputveld: Een simpele, duidelijke tekstbalk onderaan het scherm. Dit is het primaire interactiepunt.
- Chatgeschiedenis: Een zijbalk waarin eerdere gesprekken worden opgeslagen en georganiseerd. Dit stelt gebruikers in staat om context te behouden en terug te keren naar eerdere ideeën.
- Streaming Respons: De AI genereert het antwoord woord voor woord, wat de interactie dynamisch en 'levend' maakt. Het geeft de gebruiker het gevoel dat de AI 'nadenkt'.
- Feedbackknoppen: Knoppen (duim omhoog/omlaag) om de kwaliteit van de output te beoordelen.

Relevantie voor mijn interface:

Inputveld en Chatgeschiedenis: Dit is de meest intuitieve en geaccepteerde interface voor tekstgebaseerde AI. Het is verstandig om deze structuur over te nemen, omdat de doelgroep hier volledig aan gewend is. Het verlaagt de leercurve.

Streaming Respons: Het 'live' genereren van tekst is een perfect moment om de thermische feedback te synchroniseren. Terwijl de woorden op het scherm verschijnen, kan de gebruiker de warmte voelen toenemen, wat de link tussen het 'denkproces' van de AI en het energieverbruik versterkt.

2. Midjourney (via Discord)

Beschrijving:

Midjourney is een AI-beeldgenerator die werkt via commando's in een Discord-server. Gebruikers typen een beschrijving (prompt) en de AI genereert vier visuele opties.

Belangrijkste Elementen:

- Prompt-gebaseerde Interactie: De kern van de interactie is het /imagine commando, gevolgd door een beschrijvende tekst.
- Visuele Output in een Grid: De AI presenteert vier verschillende interpretaties van de prompt in een 2x2 raster.
- Iteratieve Knoppen: Onder de afbeeldingen staan knoppen voor 'Upscale' (U1-U4) en 'Variations' (V1-V4). Dit nodigt uit tot verfijning en verdere generatie, wat leidt tot een cumulatief energieverbruik.
- Community Feed: De resultaten zijn vaak openbaar zichtbaar in een kanaal, wat een gevoel van gemeenschap en inspiratie creëert.

Relevantie voor mijn interface:

Iteratie en Escalatie: Het model van Midjourney (genereren -> variëren -> upscalen) is een perfect voorbeeld van hoe AI-gebruik kan escaleren. Elke klik op 'Variations' of 'Upscale' is een nieuwe, energie-intensieve actie. Dit patroon kan direct worden gekoppeld aan een toenemende warmte in mijn prototype, waardoor de gebruiker de 'kosten' van perfectie voelt.

Hoe moeilijker de taak, hoe groter de impact. Het maken van afbeeldingen met AI kost bijvoorbeeld heel veel energie. Als ik deze functie in mijn prototype opneem, kan ik de meest intense warmtefeedback gebruiken om de boodschap extra krachtig te maken.

3. Google Gemini

Beschrijving:

Gemini is Google's concurrent van ChatGPT, met een sterke focus op integratie met het Google-ecosysteem (Search, Docs, Gmail).

Belangrijkste Elementen:

- Real-time Internettoegang: In tegenstelling tot de standaardversie van ChatGPT, kan Gemini actuele informatie van het internet halen en bronnen vermelden.
- Meerdere Concepten (Drafts): Gemini biedt vaak meerdere versies van een antwoord aan, waaruit de gebruiker kan kiezen.
- Integratie met andere tools: De mogelijkheid om content direct te exporteren naar Google Docs of Gmail.

Relevantie voor mijn interface:

Transparantie van Bronnen: Het tonen van bronnen, zoals Gemini doet, is een vorm van transparantie. Jouw project kan dit uitbreiden door niet alleen de informatiebron te tonen, maar ook de energiebron (of de kosten ervan).

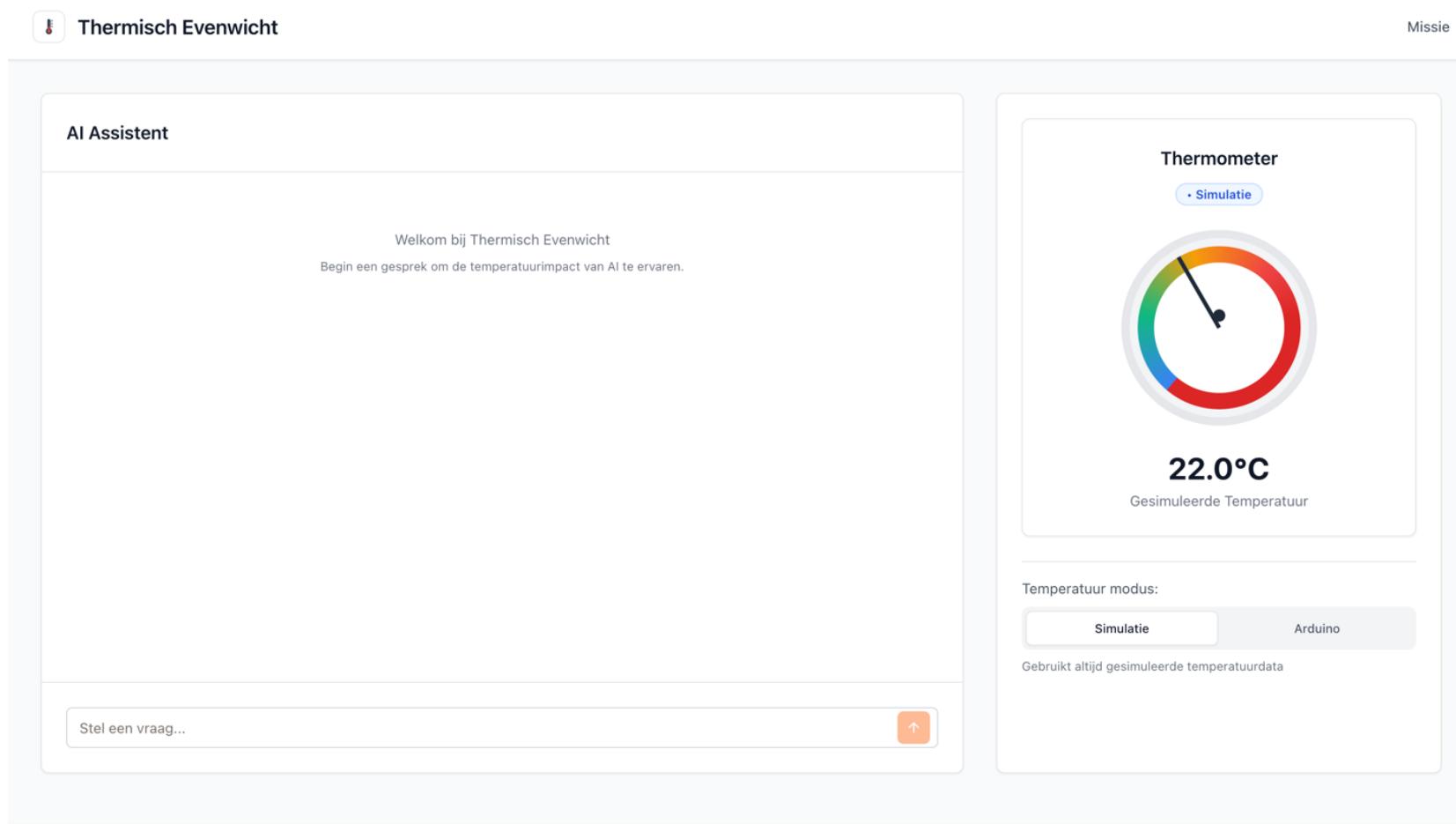
De Illusie van Moeitelosheid: Gemini's snelheid en toegang tot real-time data versterken de illusie dat informatie 'gratis' en moeitelos beschikbaar is. Jouw prototype kan juist deze illusie doorbreken door de fysieke inspanning achter deze snelle zoekopdrachten voelbaar te maken.

Alle belangrijke inzichten op een rijtje voor de ontwikkeling van mijn interface:

- Gebruik een Chat-Interface als Basis: Dit is de meest vertrouwde interactievorm voor de doelgroep. Een simpel inputveld en een conversatiegeschiedenis zijn essentieel.
- Bied Diverse Taken aan: Integreer zowel tekst- als beeldgeneratie (of simuleer dit). Dit stelt je in staat om de thermische feedback te variëren en de gebruiker het verschil in energieverbruik tussen simpele en complexe taken te laten voelen.
- Visualiseer de 'Kosten' naast de Fysieke Feedback: Combineer de warmte van de mat met een simpele, visuele indicator op het scherm (een 'energiemeter' of 'CO2-teller'). Dit versterkt de boodschap en maakt de link tussen actie en gevolg onmiskenbaar.
- Focus op de Actie-Reactie Loop: Het belangrijkste is de directe koppeling. Zodra de gebruiker op 'Enter' drukt, moet de warmte beginnen. Wanneer de AI klaar is, kan de warmte langzaam afnemen. Dit maakt de ervaring dynamisch en leert de gebruiker intuïtief over de impact.
- Houd het Doelgericht: In plaats van een eindeloze chat, kun je de gebruiker een specifieke, zelfgekozen taak laten uitvoeren (zoals in het "Deadline Mat" concept). Dit past beter bij de werk- en studiecontext en geeft de ervaring een duidelijk begin en einde, wat de reflectie achteraf versterkt.

Concept 1:

Zoals in het laatste embodied concept te zien is, is hier het eerste interface ontstaan. In dit interface zijn 2 belangrijke componenten te zien waar de gebruiker informatie uit kan halen en dus mee kan interacteren. De AI assistent en de thermometer, omdat dit interface natuurlijk ontworpen is naar de belangen en functionaliteit van het beschreven embodied experiment concept zien we dat sommige componenten erg nuttig zijn zoals de AI assistent en thermometer maar andere zoals, de temperatuurmodus ook weer overbodig zijn. Ondanks dit biedt dit eerste interface een erg goede basis om op voort te borduren en verder te ontwikkelen.



Figuur 49: Het eerste interfaceconcept, met een AI-assistent en een thermometer als visuele feedback.

Concept 2: Een Werkplek met Duidelijke Taken

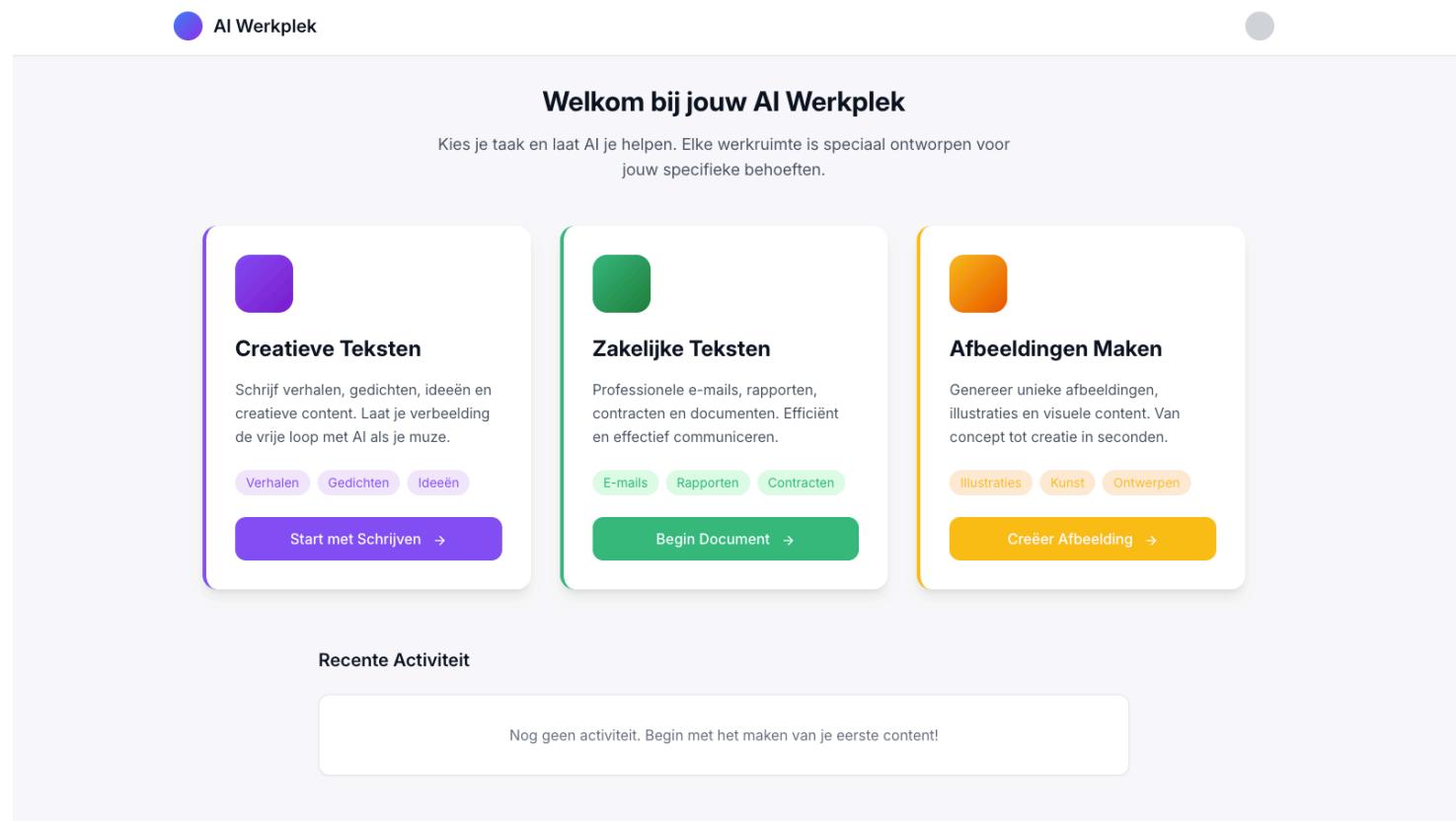
In plaats van één algemene chatbox (zoals in Concept 1), biedt dit concept een startmenu met duidelijke keuzes. Uit mijn onderzoek weet ik namelijk precies wat voor soort taken mijn doelgroep met AI doet. De belangrijkste zijn:

- Creatieve teksten schrijven (zoals verhalen of ideeën)
- Zakelijke teksten schrijven (zoals e-mails, samenvattingen, contracten en projectbeschrijvingen)
- Afbeeldingen maken

In dit concept kiest de gebruiker dus eerst wat hij wil gaan doen. Het is als een werkplek met verschillende gereedschappen, in plaats van één grote gereedschapskist. Dit past beter bij hoe mensen in hun werk of studie een taak starten: meestal met een duidelijk doel.

Hoe het zou werken:

Het scherm toont de drie knoppen. Klik je op "Afbeeldingen maken"? Dan krijg je een scherm dat daarvoor gemaakt is. Klik je op "Zakelijke teksten schrijven"? Dan krijg je een simpele tekst-editor. Elke taak geeft een ander gevoel van warmte.



Figuur 50: Het tweede interfaceconcept, een taakgerichte werkplek met duidelijke categorieën.

Wat is er goed aan dit idee? (Pluspunten)

- Heel herkenbaar voor de gebruiker: Omdat het ontwerp de taken gebruikt die de doelgroep al uitvoert, voelt het meteen vertrouwd en nuttig.
- Je voelt het verschil in impact: Dit idee maakt het makkelijk om de warmte aan te passen. Een afbeelding maken kost veel energie, dus dat geeft veel en snel warmte. Een simpele tekst schrijven geeft minder warmte. Zo leert de gebruiker vanzelf welke taken 'zwaar' zijn.
- Duidelijk begin en einde: Een taak is op een gegeven moment klaar. Dit is beter dan een eindeloze chat. Het geeft een duidelijk moment om even na te denken over de ervaring.

Wat is minder goed aan dit idee? (Minpunten)

- Niet zo flexibel: Wat als je iets wilt doen dat niet onder de drie knoppen valt? Het systeem is best stijf en geeft weinig vrijheid. Dat kan irritant zijn voor de gebruiker.
- Moeilijker om te maken: Het is veel meer werk om drie verschillende schermen te ontwerpen en te bouwen dan één simpele chat. Het gevaar is dat elk onderdeel er een beetje simpel uit komt te zien.
- Kan voor verwarring zorgen: De gebruiker moet eerst een keuze maken voordat hij kan beginnen. Dat is een extra stap. Als de knoppen niet precies zijn wat de gebruiker zoekt, haakt hij misschien af.

Conclusie voor concept 2:

Dit idee is een grote stap vooruit, omdat het de warmte koppelt aan echte, herkenbare taken. Het grootste nadeel is dat het de gebruiker beperkt in zijn/haar AI verzoeken. Een idee voor een volgend prototype zou zijn om het beste van twee werelden te combineren: de vrijheid van een chat, met speciale knoppen voor zware taken zoals het maken van een afbeelding.

Concept 3: Thermal AI (definitieve interface)

Het definitieve interface-ontwerp, genaamd 'Thermal AI', is de combinatie van de inzichten verkregen uit de analyse van bestaande AI-platformen en de experimenten met Concept 1 en Concept 2. Dit ontwerp is een model dat de vertrouwde interactie van een chat-interface combineert met een visuele en fysieke feedbackloop, gericht op het bewustmaken van de milieu-impact van AI-gebruik.

Overzicht van de Interface (zie figuur 51):

De interface van Thermal AI is opgebouwd uit twee hoofddelen:

1. De Zijbalk (Links): Gespreksgeschiedenis en Impactoverzicht

- 'Nieuw Gesprek' Knop: Bovenaan de zijbalk bevindt zich een duidelijke knop om een nieuwe interactie te starten, wat de gebruiker een frisse start biedt.
- Gespreksgeschiedenis: Hier worden eerdere gesprekken overzichtelijk weergegeven. Elk gesprek heeft een titel en een indicatie van de tijd die verstrekken is sinds de laatste interactie. Cruciaal is de visuele feedback over de 'warmte' of 'impact' van het gesprek.
- Impactindicatoren: Elk gesprek toont een percentage en een bijbehorende kleur/tekst (bijv. 'Koel (0%)', 'Warm (20%)', 'Zeer Heet (65%)', 'Maximum (100%)'). Dit geeft de gebruiker in één oogopslag een overzicht van de cumulatieve impact van dat specifieke gesprek. De kleur en het percentage fungeren als een visuele 'kostenmeter' voor de gehele conversatie.
- Statusindicator: Naast de impactindicatie is er een status (bijv. 'Bezig', 'Voltooid') die aangeeft of de AI nog actief is in dat gesprek.

2. Het Hoofdvenster (Rechts): Interactie en Welkom scherm

- Welkomstscherf: Bij het opstarten of bij een nieuw gesprek toont het hoofdvenster een welkomstbericht dat de gebruiker introduceert in het concept van Thermal AI: "Ervaar AI-interacties met echte energieverbruik weergave via ons thermische systeem. Gesprekken gaan door totdat 100% warmte is bereikt, waardoor abstract AI energieverbruik tastbaar wordt." Dit scherm fungeert als een korte handleiding en zet de toon voor de ervaring.
- Instructies: Drie duidelijke instructiepunten begeleiden de gebruiker:
 - "Start een nieuw gesprek vanuit de zijbalk"
 - "Bekijk hoe het warmteniveau stijgt bij elke interactie"
 - "Verbind Arduino hardware voor fysieke warmte-feedback" (Dit laatste punt benadrukt de koppeling met de fysieke muismat).
- Interactiegebied: Zodra een gesprek is gestart, transformeert dit venster in de daadwerkelijke chat-interface, waar de gebruiker prompts invoert en de AI-antwoorden verschijnen.

Bovenbalk: Systeemstatus en Branding

De bovenbalk van de interface toont belangrijke systeemstatusinformatie:

- Branding: Het 'ThermalAI' logo met de slogan "Ervaar AI energieverbruik" versterkt de merkidentiteit en de kernbericht.
- Energieverbruik Status: Een indicator "AI Energieverbruik" met een percentage (bijv. "Koel (0%)") geeft de actuele, totale impactstatus van de lopende sessie weer. Dit is de visuele tegenhanger van de fysieke warmte van de muismat.
- Hardware Status: Een indicator "Geen Arduino" toont de verbindingssatus met de fysieke hardware (de thermische muismat), wat essentieel is voor de embodied experience.

Werking:

Dit definitieve interface-ontwerp is een robuuste oplossing die de voordelen van flexibiliteit (vrije chat) en taakgerichtheid (implicit via de impact van prompts) combineert. Het biedt een vertrouwde, maar tegelijkertijd innovatieve en confronterende gebruikerservaring. Door de naadloze integratie van fysieke (via de muismat) en visuele (via de interface) feedback, is Thermal AI in staat om de abstracte milieu-impact van AI op een diepgaande en memorabele wijze over te brengen, en zo de gebruiker aan te zetten tot waardevolle reflectie op hun eigen digitale gedrag. De interface is niet alleen een tool voor interactie, maar een educatief instrument dat de gebruiker continu bewust maakt van de verborgen kosten van hun digitale acties.

The screenshot displays the ThermalAI interface. On the left, a sidebar titled "Gesprekken" shows a list of four conversations with their respective timestamps and status (Bezig/Voltooid). The first message reads: "How can I assist you today?". The second message, from "Gesprek Maximum (100%)", says: "Dat is een belangrijke toevoeging. Hier is hoe je dat op een respectvol...". The third message, from "Gesprek Zeer Heet (65%)", says: "Dank voor deze extra informatie. Het klinkt alsof we een rooster moeten...". The fourth message, from "Gesprek Warm (20%)", says: "Natuurlijk! Ik help je graag met het opstellen van een e-mail. Kun je me...".

ThermalAI
Ervaar AI energieverbruik

AI Energieverbruik Geen Arduino
Koel (0%)

Welkom bij ThermalAI

Ervaar AI-interacties met echte energieverbruik weergave via ons thermische systeem. Gesprekken gaan door totdat 100% warmte is bereikt, waardoor abstract AI energieverbruik tastbaar wordt.

- Start een nieuw gesprek vanuit de zijk balk
- Bekijk hoe het warmteniveau stijgt bij elke interactie
- Verbind Arduino hardware voor fysieke warmte-feedback

Figuur 51: Het definitieve interface-ontwerp "Thermal AI", met een geïntegreerde gespreksgeschiedenis en impactindicatoren.

DEFINITIEF ONTWERP

Mijn definitieve ontwerp bestaat uit 2 delen, het product en de daarbij ondersteunende interface.

Na een zorgvuldige analyse van de voorgaande concepten en op basis van de belangrijkste onderzoeksresultaten, presenteert ik hier het definitieve prototype. Dit ontwerp combineert de beste elementen uit de onderzoeksfase en de conceptontwikkeling tot een coherente en impactvolle ervaring die voldoet aan een reeks heldere ontwerpcriteria.

Mijn definitieve ontwerp bestaat uit twee delen die met elkaar verbonden zijn:

- Het fysieke product: Een thermische muismat die werkt als de kern van de ervaring.
- De ondersteunende interface: Een desktop-applicatie die dient als het gezicht van het AI-model.

1. Het Product: De Thermische Muismat

Het hart van het ontwerp is een muismat die warmte kan overbrengen. Wanneer een gebruiker een taak uitvoert via het AI-model, geeft de mat fysieke feedback in de vorm van warmte. Deze warmte is een directe, fysieke vertaling van de onzichtbare energie die de AI-taak verbruikt. Door deze embodied experience voelt de gebruiker letterlijk de impact van zijn of haar digitale acties.

Dit product is specifiek ontworpen voor de doelgroep van 20 tot 30 jaar die voor werk of studie al veel AI gebruikt. Omdat zij zich voornamelijk achter een bureau bevinden, is een muismat een logische keuze: het is een object dat al een natuurlijke plek heeft in hun werkomgeving en direct in contact staat met de onderarm, wat de fysieke sensatie versterkt.

2. De Interface: Het Gezicht van de AI

De muismat wordt aangestuurd en ondersteund door een speciaal ontworpen interface. Dit is meer dan alleen een dashboard; het is het gezicht en de persoonlijkheid van het AI-model zelf. Hier vindt de interactie plaats. De interface is ontworpen op basis van de belangrijkste inzichten uit de analyse van bestaande platformen:

- Vertrouwde Chat-Interface: De basis is een chat-interface, de meest intuitieve en geaccepteerde vorm voor de doelgroep.
- Diverse Taken: De gebruiker kan kiezen uit verschillende taken, zoals tekst- en beeldgeneratie. Dit maakt het mogelijk om de warmtefeedback te variëren en de gebruiker het verschil in impact tussen een simpele en een complexe taak te laten voelen.
- Visualisatie van Kosten: Naast de fysieke warmte toont de interface een simpele, visuele 'energiemeter'. Dit versterkt de boodschap en maakt de link tussen actie en gevolg onmiskenbaar.
- Doelgerichte Interactie: Om de ervaring gefocust te houden, kan de gebruiker specifieke taken starten. Dit geeft de interactie een duidelijk begin en einde, wat reflectie achteraf stimuleert.

De Werking van het Systeem: Van Prompt naar Fysieke Feedback

Het definitieve ontwerp, de "Thermische muismat", werkt op basis van een directe en intuïtieve feedbackloop. Het proces van hoe een gebruikersactie wordt omgezet in fysieke feedback kan als volgt worden omschreven:

1. De Gebruiker Voert een Prompt In: De interactie begint in de speciaal ontworpen interface. De gebruiker stelt een vraag of geeft een opdracht aan het AI-model, net zoals hij of zij dat bij andere AI-platformen zou doen.
2. Het Systeem Analyseert de Taak: Zodra de gebruiker op 'Enter' drukt, analyseert het systeem de complexiteit van de ingevoerde prompt. Het maakt hierbij onderscheid tussen verschillende soorten taken. Een verzoek voor beeldgeneratie wordt bijvoorbeeld geklassificeerd als een zeer energie-intensieve taak. Een simpele tekstanalyse of het samenvatten van een kort stuk tekst wordt gezien als een minder zware taak.
3. De Feedback wordt Berekend en Geactiveerd: Op basis van deze analyse berekent het systeem de "kosten" van de prompt en vertaalt dit naar een specifieke hoeveelheid warmte en een bepaalde opwärmsnelheid voor de muismat.

Een complexe prompt (zoals beeldgeneratie) zal direct resulteren in een snelle en intense opwarming van de mat. De gebruiker voelt de temperatuur significant stijgen, wat de hoge energiekosten van de taak symboliseert.

Een eenvoudige prompt (zoals een korte tekstvraag) zal leiden tot een milder en langzamere opwarming. De feedback is subtieeler, wat de lagere impact van de taak weerspiegelt.

4. Directe Koppeling en Reflectie: Deze directe koppeling tussen de vorm van de prompt en de intensiteit van de fysieke sensatie is de kern van de ervaring. De gebruiker leert niet via tekst of cijfers, maar door te voelen dat verschillende AI-acties een verschillende ecologische prijs hebben. Dit creëert een krachtig en onmiddellijk bewustzijn en zet aan tot reflectie over de efficiëntie en noodzaak van het eigen AI-gebruik.

Waarom dit ontwerp effectief is:

Dit geïntegreerde ontwerp is zorgvuldig opgebouwd om aan de vooraf opgestelde doelen en criteria te voldoen:

Doelen:

- Verhoogd Bewustzijn: De combinatie van voelbare warmte en zichtbare data zorgt voor een dieper bewustzijn van de onzichtbare kosten van AI.
- Gedragsreflectie: De directe, fysieke feedback zet de gebruiker aan tot nadenken over zijn of haar digitale gewoonten.
- Memorabele Ervaring: De unieke, zintuiglijke ervaring blijft langer hangen dan een simpele grafiek of tekst, wat de betrokkenheid verhoogt.

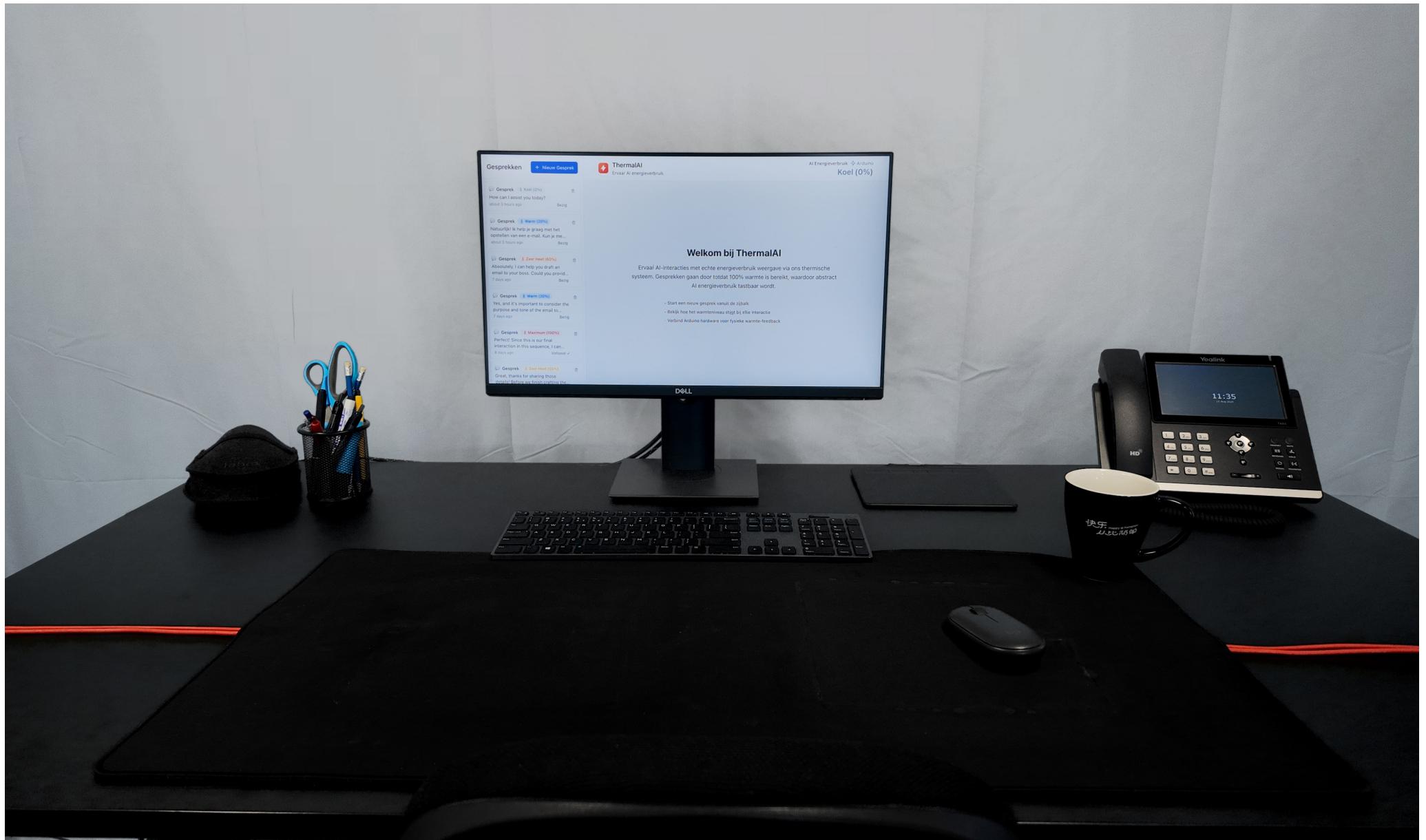
Ontwerpcriteria:

- Fysieke en Visuele Feedback: De mat levert de embodied feedback, terwijl de interface deze ondersteunt met visuele signalen.
- Direct en Begrijpelijk: De Actie-Reactie Loop is de kern. Zodra de gebruiker op 'Enter' drukt, begint de warmte. De relatie tussen actie en gevolg is direct en intuïtief.
- Aansprekend en Stabiel: Het ontwerp sluit aan bij de digitale levensstijl van de doelgroep en is ontworpen om stabiel en consistent te functioneren, wat essentieel is voor een geloofwaardige ervaring.

Conclusie: Een Antwoord op de Hoofdvraag

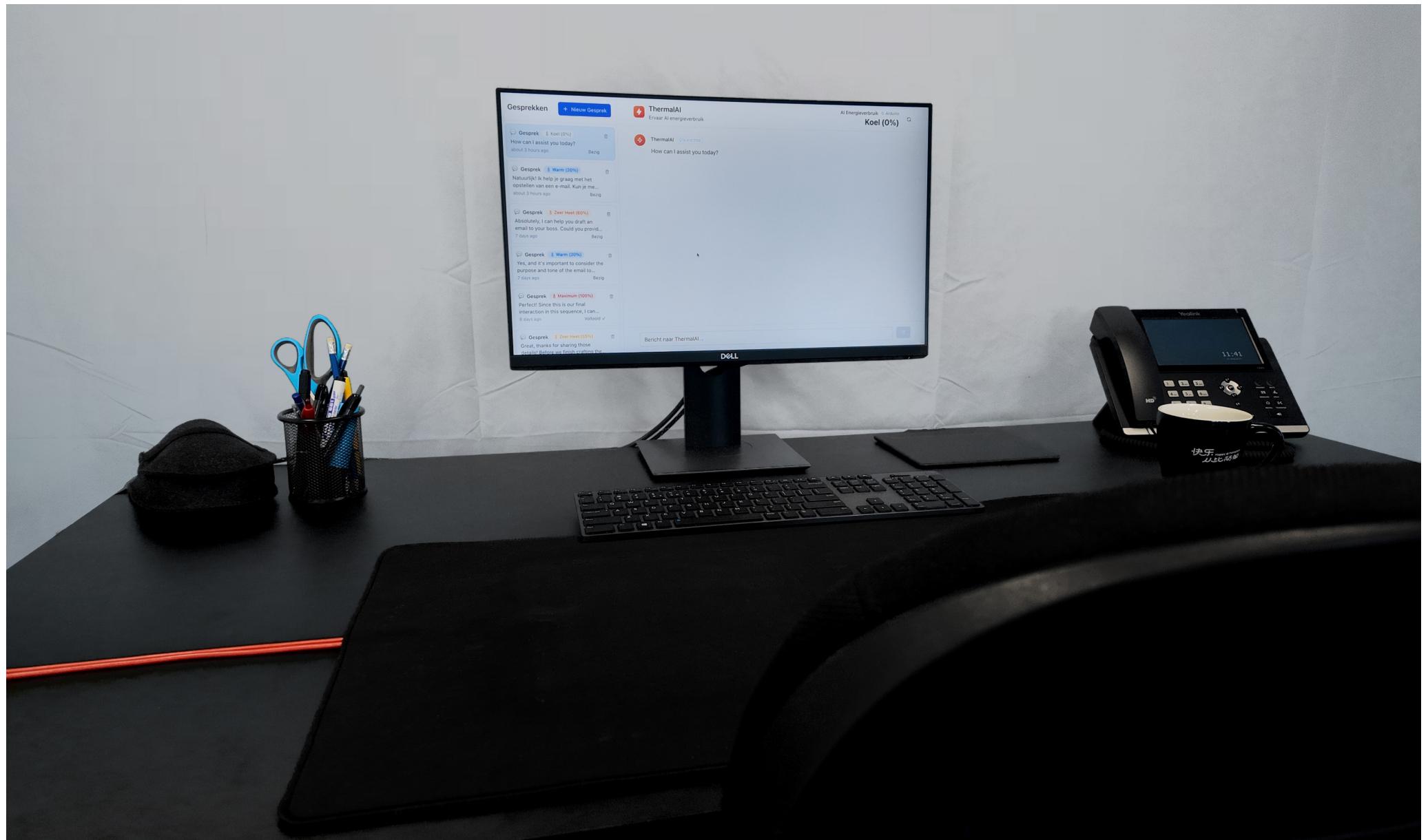
Met de ontwikkeling van de 'Thermische muismat' is een concreet antwoord geformuleerd op de centrale hoofdvraag van dit project: 'Hoe kan een embodied experience worden ingezet om gebruikers van AI-toepassingen (20-30 jaar) bewust te maken van de (negatieve) milieu-impact, met als doel een waardevolle reflectie op hun eigen AI gebruik?'

De conclusie is dat het ontworpen product werkt. Door de slimme integratie van thermische feedback in een alledaags object zoals een muismat, is het gelukt om de abstracte impact van AI-gebruik op het milieu tastbaar en direct voelbaar te maken. De 'Thermische muismat' confronteert de gebruiker op een unieke wijze met de ecologische kosten van hun digitale acties, en stimuleert daarmee een waardevolle reflectie op hun eigen AI-gebruik. Dit prototype bewijst dat embodied experience een krachtig middel is om bewustzijn te creëren en aanzet tot duurzamer handelen in de digitale wereld.

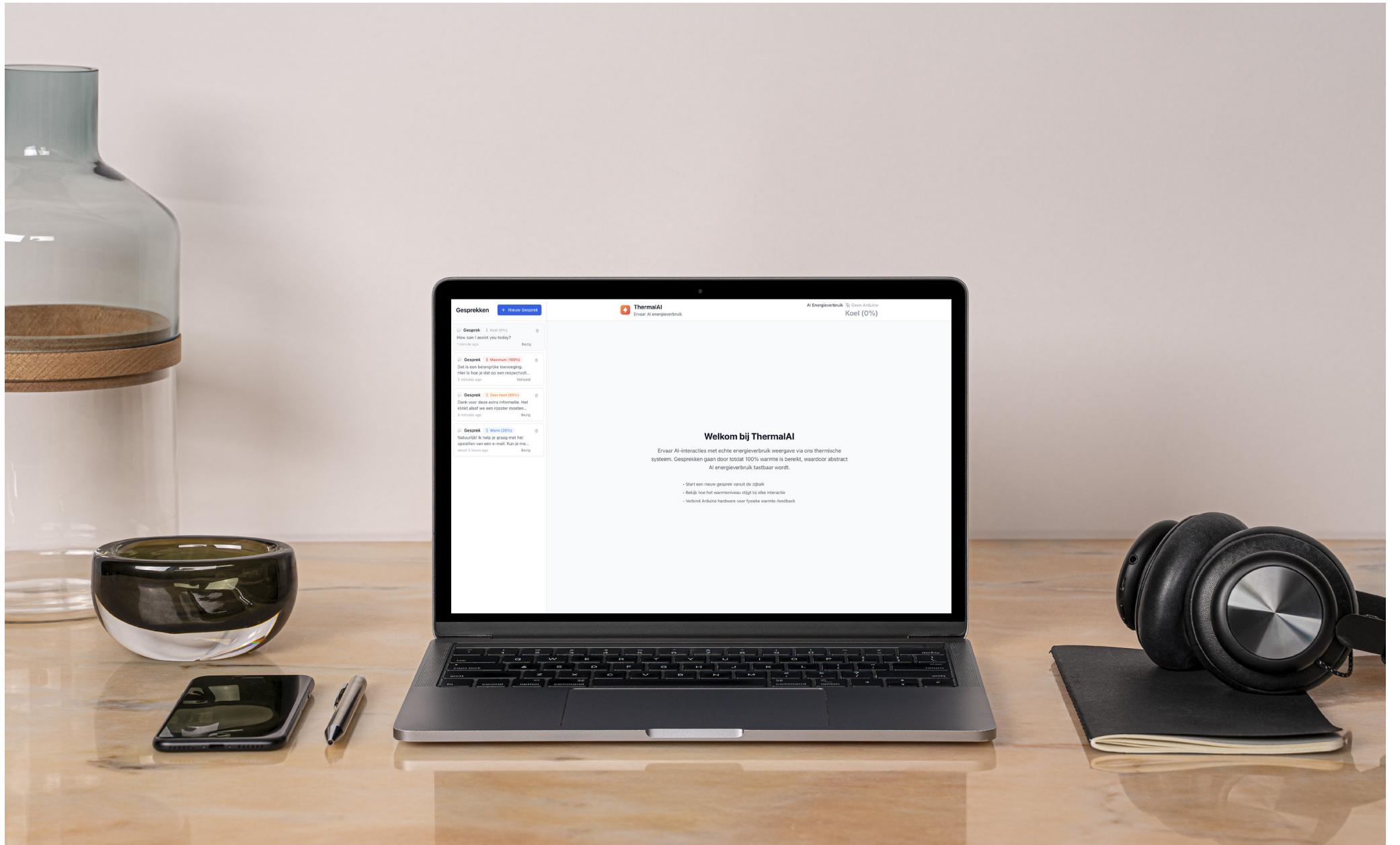


Figuur 52: Het definitieve productontwerp: de thermische muismat, het fysieke element van de "Thermische muismat" ervaring.

DEFINITIEF ONTWERP

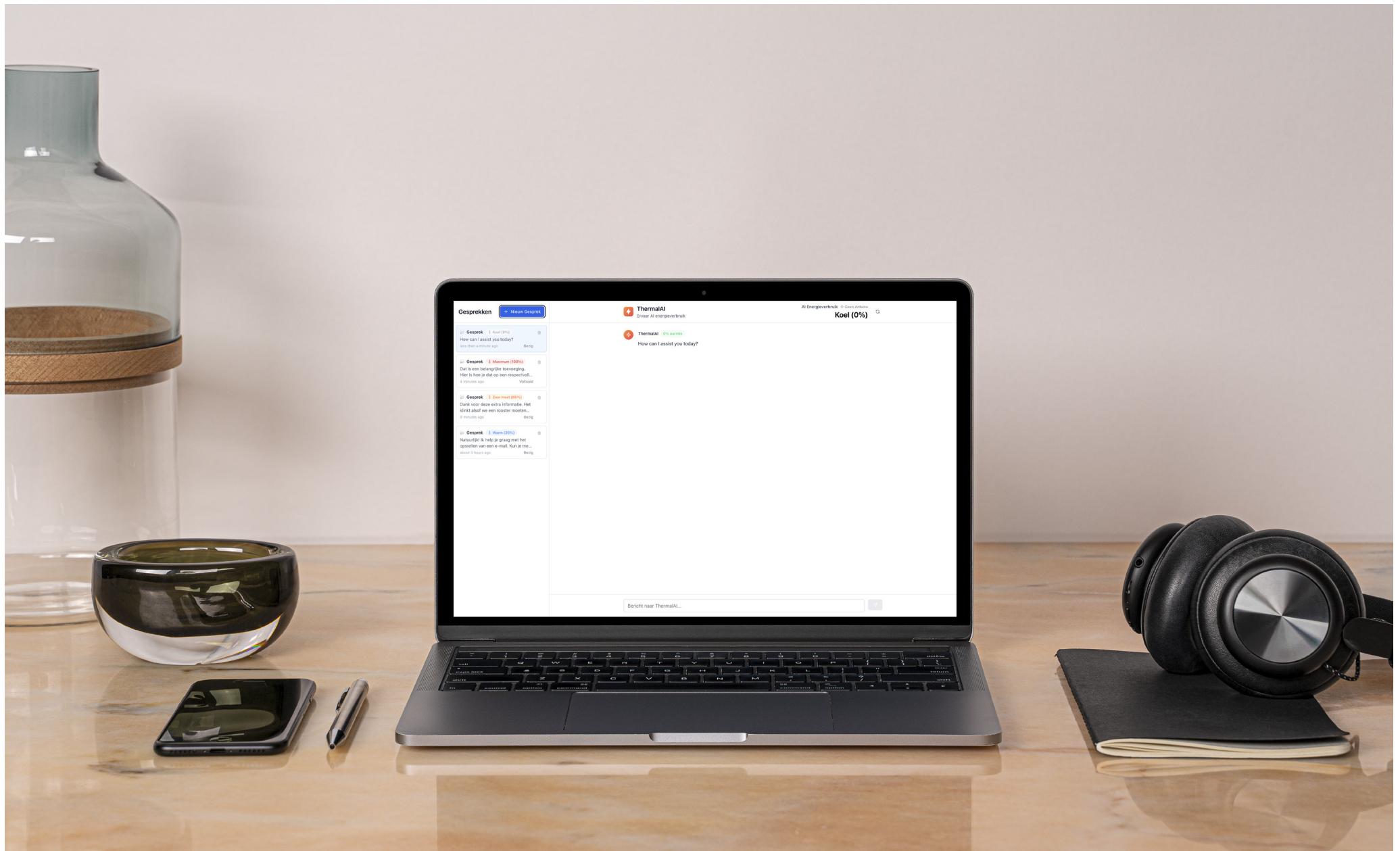


Figuur 53: De “Thermische muismat” in gebruik, illustrerend de integratie van de thermische muismat en de interface in de werkplek van de gebruiker.



Figuur 54: Het definitieve interface-ontwerp: de desktop-applicatie "Thermal AI", het visuele gezicht van het AI-model.

DEFINITIEF ONTWERP



Figuur 55: Het definitieve interface-ontwerp: de desktop-applicatie "Thermal AI", met een open chat.

DEFINITIEF ONTWERP

16

BIJLAGEN

Interview

Deelnemer: Lotte, 26 jaar

Interviewer: Dinand, 22 jaar

Datum: 24 juli 2025

Onderwerp: Perceptie van en omgang met AI en duurzaamheid.

Interviewer: "Hoi Lotte, bedankt dat je tijd wilde maken. Kun je om te beginnen iets vertellen over hoe jij AI gebruikt in je dagelijks leven?"

Lotte: "Zeker! Ik gebruik het eigenlijk de hele dag door. Voor mijn werk als data-analyst gebruik ik het om code te debuggen of om snel een stukje Python-script te laten schrijven door ChatGPT. Dat scheelt me echt uren. Privé vraag ik het ook van alles, van recepten tot het samenvatten van een ingewikkeld nieuwsartikel. En dan heb je natuurlijk nog de 'onzichtbare' AI, zoals de aanbevelingen van Spotify en Netflix. Die staan altijd aan."

Interviewer: "Dat is inderdaad frequent gebruik. Je noemt zowel werk als privé. Waar zie je de grootste waarde van AI voor jou persoonlijk?"

Lotte: "Het is vooral een enorme tijdsbesparing en een soort creatieve sparringpartner. Als ik vastzit met een probleem, kan ik het aan een AI vragen en krijg ik direct een ander perspectief. Het haalt de frictie weg. Voorheen moest ik eindeloos Googlen en forums afstruinen, nu heb ik vaak in één keer een antwoord. Het voelt als een soort oneindig geduldige assistent."

Interviewer: "En hoe zit het met creatieve tools, zoals DALL-E of Midjourney? Gebruik je die ook?"

Lotte: "Ja, af en toe. Meer voor de lol. Ik heb het wel eens gebruikt om een grappige afbeelding te maken voor een verjaardagskaart of om een concept te visualiseren voor een presentatie. Het is fascinerend om te zien wat eruit komt. Ik speel er dan wel echt mee, probeer allerlei gekke prompts uit. Gewoon om te zien wat de grenzen zijn."

Interviewer: "Duidelijk. Laten we het even over een ander onderwerp hebben: duurzaamheid. Is dat iets waar je in je dagelijks leven mee bezig bent?"

Lotte: "Ja, best wel. Ik probeer minder vlees te eten, scheid mijn afval, en ik koop liever tweedehands kleding dan fast fashion. Ik heb geen auto, dus ik doe alles op de fiets. Het zijn de standaard dingen, denk ik. Ik vind het belangrijk om mijn steentje bij te dragen, al voelt het soms als een druppel op een gloeiende plaat."

Interviewer: "Heb je ooit een verband gelegd tussen die twee onderwerpen: jouw AI-gebruik en duurzaamheid?"

Lotte: "Eerlijk gezegd... nee, nog nooit. Het is voor mij een compleet digitale, abstracte wereld. Ik zie de servers niet, ik zie geen rook uit een datacentrum komen. Als ik mijn laptop oplaad, weet ik dat het stroom kost. Maar als ik een prompt intyp bij ChatGPT... dat voelt als iets wat gewoon 'in de cloud' gebeurt. Het heeft geen fysieke impact in mijn beleving."

Nu je het zo vraagt, voelt het bijna naïef dat ik daar nog nooit bij stil heb gestaan."

Interviewer: "Dat is een heel herkenbare gedachte. Stel je voor: je zou te zien krijgen dat een complexe vraag aan een AI-model evenveel energie kost als een uur je lamp laten branden. Wat zou dat met je doen?"

Lotte: "Wow, dat zou wel wat veranderen, ja. Dan zou ik er veel bewuster mee omgaan. Nu stel ik soms vijf keer dezelfde vraag op een net iets andere manier om het perfecte antwoord te krijgen. Als ik zou weten dat elke vraag een 'prijs' heeft, zou ik misschien eerst zelf beter nadenken over hoe ik de vraag formuleer. Ik zou het niet minder gebruiken, denk ik, maar wel... efficiënter. Minder gedachteloos."

Interviewer: "En waar zou voor jou de grens liggen? Zou je het voor bepaalde 'onnodige' dingen, zoals die grappige afbeeldingen, dan laten?"

Lotte: "Goeie vraag. Misschien wel, ja. Of ik zou het in ieder geval minder doen. Als het maken van één zo'n plaatje gelijkstaat aan een dag mijn koelkast laten draaien, dan voelt het ineens niet meer zo onschuldig. Dan wordt het een luxe-uitgave van energie. Dan zou ik het bewaren voor momenten waarop ik het écht nodig heb, zoals voor die presentatie, en niet zomaar voor de lol."

Interviewer: "Bij wie ligt volgens jou de verantwoordelijkheid om dit probleem aan te pakken? Bij jou als gebruiker, of bij de grote techbedrijven die de AI-modellen aanbieden?"

Lotte: "Uiteindelijk bij de bedrijven, vind ik. Zij bouwen die systemen en zij zijn de enigen die écht weten hoeveel energie het kost. Ik vind dat ze daar transparant over moeten zijn. Maar ik snap ook dat ik als gebruiker een rol heb. Als wij er massaal om vragen of bewuster worden, moeten zij wel meebewegen. Het is een wisselwerking. Maar de eerste stap, de openheid, die moet van hen komen."

Interviewer: "En wat zou je nodig hebben om die efficiëntere keuzes te kunnen maken? Hoe zou je die informatie willen ontvangen?"

Lotte: "Inzicht, denk ik. Een soort 'energielabel' voor AI-tools zou al helpen. Of een simpele visualisatie, direct in de interface. Geen ingewikkelde grafieken, maar iets wat direct duidelijk maakt: 'Deze actie kost X'. Het moet me niet het gevoel geven dat ik iets 'fout' doe, maar me eerder helpen om een betere keuze te maken. Een soort bewustwordingstool. Misschien een browser-extensie die een kleine boom laat groeien als je duurzame keuzes maakt, en laat zien hoeveel 'water' je AI-gebruik kost. Iets wat het speels en tastbaar maakt."

Interviewer: "Tot slot, als je morgen één ding zou kunnen veranderen aan hoe AI werkt in relatie tot duurzaamheid, wat zou dat dan zijn?"

Lotte: "Dat er een universele, verplichte standaard voor transparantie komt. Dat elke AI-tool moet laten zien wat de ecologische kosten zijn per handeling. Net zoals je op een voedselverpakking de calorieën ziet. Dat zou de hele dynamiek veranderen. Dan wordt duurzaamheid een factor in de concurrentiestrijd en gaan bedrijven vanzelf innoveren om hun modellen efficiënter te maken. Dat lijkt me de enige echte oplossing op de lange termijn."

Interviewer: "Dankjewel, Lotte. Dit was ontzettend helder en waardevol."

BRONNENLIJST

NOTE:

Voor het ontwikkelen van de meeste applicaties is gebruikgemaakt van het online platform Replit. Dit platform stelde mij in staat om snel en efficiënt te coderen en diverse experimenten uit te voeren binnen een toegankelijke en interactieve ontwikkelomgeving.

<https://replit.com/>

Daarnaast is voor het verzamelen en ordenen van relevante literatuur en artikelen ter ondersteuning van het onderzoek gebruikgemaakt van ChatGPT. Deze tool diende als hulpmiddel bij het genereren van zoektermen, het structureren van informatie en het verkrijgen van eerste inhoudelijke verkenningen.

<https://chatgpt.com/>

AG Connect. (2023, 24 juli). Eén ChatGPT-vraag kost 1 liter schoon water. Geraadpleegd van <https://www.agconnect.nl/artikel/een-chatgpt-vraag-kost-1-liter-schoon-water>

EY. (2024, 22 november). AI en Duurzaamheid: kansen, uitdagingen en impact. Geraadpleegd van https://www.ey.com/nl_nl/sustainability/ai-en-duurzaamheid-kansen-uitdagingen-en-impact

ESG News. (2024, 21 mei). Google lanceert het eerste externe warmteterugwinningsproject in Finland, dat 97% koolstofvrije energie levert aan de lokale gemeenschap. Geraadpleegd van <https://esgnews.com/google-launches-first-external-heat-recovery-project-in-finland-delivering-97-carbon-free-energy-to-local-community/>

Google. (n.d.). Hamina, Finland – Google Data Center Location. Geraadpleegd op 6 juli 2025, van <https://www.google.com/about/datacenters/locations/hamina-finland/>

Hardware Info. (2023, 12 oktober). Voorspelling: AI-energieverbruik kan binnenkort Nederland overtreffen. Geraadpleegd van <https://nl.hardware.info/nieuws/86964/voorspelling-ai-energieverbruik-kan-binnenkort-nederland-overtreffen>

IEA. (2024, 15 oktober). Stroom vooruit: Het wereldwijde elektriciteitslandschap tot 2026. Geraadpleegd van <https://www.eeip.eu/power-ahead-the-global-electricity-landscape-to-2026/>

InkopersCafe.nl. (2025, 24 april). Productie AI-chips in Oost-Azië vergroot CO2-uitstoot. Geraadpleegd van <https://www.inkoperscafe.nl/2025/04/productie-ai-chips-in-oost-azie-vergroot-co2-uitstoot/>

IO+. (2024, 15 oktober). Energieverbruik datacenters stijgt, maar ze worden wel efficiënter. Geraadpleegd van <https://www.io-plus.nl/energieverbruik-datacenters-stijgt-maar-ze-worden-wel-efficiënter/>

KIA Community. (2024, 23 september). De paradox van AI's milieu-impact: dit moet je weten. Geraadpleegd van <https://kiacomunity.nl/blogs/de-paradox-van-ais-milieu-impact-dit-moet-je-weten>

Markey, E. (2024, 1 februari). Markey, Heinrich, Eshoo, Beyer Introduce Legislation to Investigate, Measure Environmental Impacts of Artificial Intelligence. Geraadpleegd van <https://www.markey.senate.gov/news/press-releases/markey-heinrich-eshoo-beyer-introduce-legislation-to-investigate-measure-environmental-impacts-of-artificial-intelligence>

Tunity. (2024, 13 augustus). Vindbaarheid op ChatGPT en Google: praktische SEO en marketingtips. Geraadpleegd van <https://tunity.nl/vindbaarheid-chatgpt-google-seo-marketingtips/>

Undetectable AI. (2025, 27 april). Hoeveel energie verbruikt ChatGPT?. Geraadpleegd van <https://undetectable.ai/nl/hoeveel-energie-verbruikt-chatgpt/>

Dodge, J., Applegate, M., Singh, A., Luccioni, A., & Smith, N. A. (2022). Measuring and Mitigating the Environmental Impact of AI. arXiv preprint arXiv:2204.05149.

Google. (2023, 14 juni). Environmental Report 2023. Geraadpleegd van <https://sustainability.google/reports/environmental-report-2023/>

Microsoft. (2023, 13 april). Microsoft Environmental Report 2023. Geraadpleegd van <https://www.microsoft.com/en-us/corporate-responsibility/environmental-report>

Ren, S., Yang, J., & Zhang, H. (2021). The Carbon Footprint of AI: A Review. arXiv preprint arXiv:2107.08764.

Stichting Internet Domeinregistratie Nederland. (2024, 20 maart). Datacenters in Nederland: een overzicht. Geraadpleegd van <https://www.sidn.nl/nieuws/datacenters-in-nederland-een-overzicht>

Europese Commissie. (2024, 13 maart). AI Act: First regulation on artificial intelligence. Geraadpleegd van <https://digital-strategy.ec.europa.eu/en/policies/artificial-intelligence-act>

Green Software Foundation. (n.d.). Software Carbon Intensity (SCI). Geraadpleegd op 6 juli 2025, van <https://greensoftware.foundation/sci>

ISO. (n.d.). ISO/IEC 5259: Information technology — Artificial intelligence — Sustainability. Geraadpleegd op 6 juli 2025, van <https://www.iso.org/standard/83457.html>

Kennisnet. (2024, 10 mei). Onderzoek: scholieren bewust van AI, maar niet van impact op milieu. Geraadpleegd van <https://www.kennisnet.nl/artikel/onderzoek-scholieren-bewust-van-ai-maar-niet-van-impact-op-milieu>

As use of A.I. soars, so does the energy and water it requires. (n.d.). Yale E360. <https://e360.yale.edu/features/artificial-intelligence-climate-energy-emissions>

- Atlas of AI - Power, Politics, and the planetary costs of Artificial intelligence - Kate Crawford | Are.Na. (n.d.). Are.na. <https://www.are.na/block/11669646>
- Çam, E., Hungerford, Z., Schoch, N., Pinto Miranda, F., Yáñez De León, C. D., & International Energy Agency. (2024). Electricity 2024. In Electricity 2024. <https://iea.blob.core.windows.net/assets/18f3ed24-4b26-4c83-a3d2-8a1be51c8cc8/Electricity2024-Analysisandforecastto2026.pdf>
- Corporate Responsibility. (2025, June 27). Environmental Sustainability Report. <https://www.microsoft.com/en-us/corporate-responsibility/sustainability/report/>
- Danelski, D. (2025, March 5). Professor's TED Talk warns of AI's hidden water costs. News. <https://news.ucr.edu/articles/2025/03/05/professors-ted-talk-warns-ais-hidden-water-costs>
- Institute of Human Obsolescence. (n.d.). Institute of Human Obsolescence. <https://speculative.capital/>
- KIA community. (n.d.). <https://kiacomunity.nl/thoughts/20334>
- Luccioni, S. (n.d.). AI is dangerous, but not for the reasons you think [Video]. TED Talks. https://www.ted.com/talks/sasha_luccioni_ai_is_dangerous_but_not_for_the_reasons_you_think/transcript
- Perry, C. (2025, June 9). Hoeveel energie verbruikt ChatGPT? AI Detector | ChatGPT Detector | AI Humanizer - Ondetecteerbare AI. <https://undetectable.ai/blog/nl/hoeveel-energie-verbruikt-chatgpt/>