

# SUCCESVERHALEN MAAK JE MET AI

de workshop waar je met ChatGPT code gaat schrijven

# **JUST VERVAART**

## **ONDERZOEKSJOURNALIST**

---

**OMROEP GELDERLAND**

**BUREAU REGIO**

# loka.al/9eh

 <b>maartensiffels</b> Add files via upload	87ba306 · 15 hours ago	 27 Commits
 readme.md	Update readme.md	15 hours ago
 stations.geojson	Add files via upload	15 hours ago

 README



## regiohelden

### Oefening I - Google Dorking

ChatGPT chatbot voor workshop <https://regiohelden.zapier.app/>

Google <https://www.google.nl>

### Oefening II - Wikidata

Wikidata <https://www.wikidata.org/>

Wikidata Query Service <https://query.wikidata.org/>

### Oefening III - Open Street Map

Open Street Map <https://www.openstreetmap.org/>

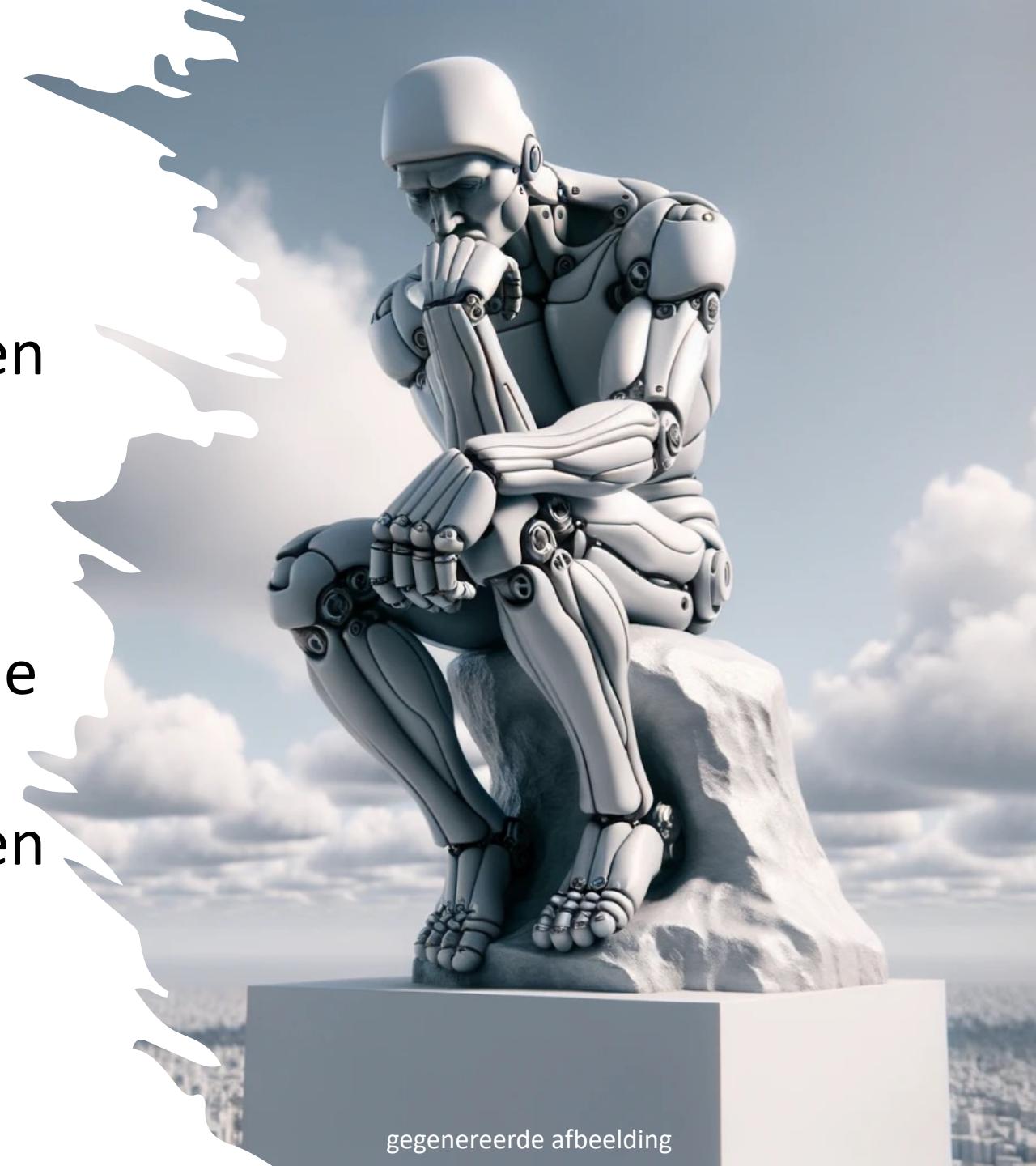
Overpass Turbo <https://overpass-turbo.eu/>

# ChatGPT

We gaan vandaag gebruik maken van ChatGPT voor het schrijven van computercode.

ChatGPT is een Generatieve AI, een kunstmatige intelligentie die in staat is om zelf te creëren.

Niet alleen tekst en afbeeldingen maar ook computercode voor complexe zoekopdrachten.



gegenereerde afbeelding

# ChatGPT 3.5 vs ChatGPT 4

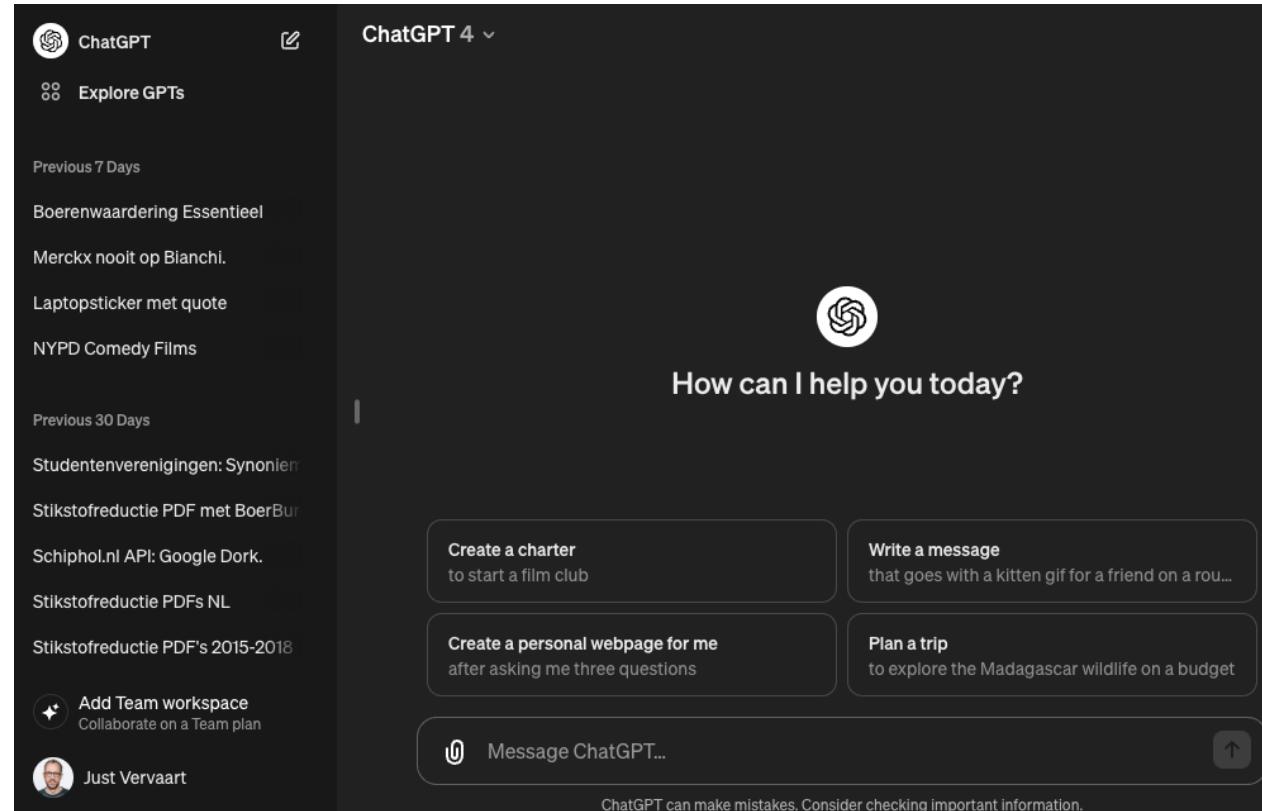
- Er zijn op dit moment twee versies van ChatGPT: V3.5 en V4.
- V3.5 is gratis en V4 kost \$24 per maand voor een persoonlijke account
- V4 is ‘slimmer’ en beter in het schrijven van computercode
- Vandaag gaan jullie dan ook aan de slag met versie 4.
- Als je na vandaag verder wilt gaan raad ik je dus aan om de portemonnee te (laten) trekken.

# ChatGPT UI

Zo ziet ChatGPT er uit als je op [chat.openai.com](https://chat.openai.com) inlogt met een eigen account.

Onderaan kun je een nieuwe vraag stellen en links zie je de oude gesprekken.

Maar dat is niet wat je vandaag gaat zien...



# regiohelden.zapier.app

Want speciaal voor deze workshop heb ik eigen chatbot geregeld waarmee jullie zonder account kunnen chatten met ChatGPT-4.

Regiohelden

Welkom bij de workshop "Succesverhalen maak je met AI".

Type hier je vraag



# Workshopformat

- Jullie gaan dadelijk zelf aan de slag. Alleen of met twee.
- Loop je vast, geef het aan dan kijk ik of ik je kan helpen.
- Als je niet krijgt wat je wilt van de AI, begin dan opnieuw (ververs de pagina) en vraag het net even anders.
- Gaat het te snel of is iets niet duidelijk. Geef dat dan aan.
- Wees niet bang om iets te proberen. Computers zijn geduldig en je kunt niets kapot maken.

# Schrik niet!

Er komen vandaag concepten en termen langs die je misschien niets zeggen. Dat is niet erg. Je hoeft ze ook niet per sé te onthouden of te snappen.

Het doel is vooral dat jullie nieuwe bronnen van informatie ontdekken en hoe je die bronnen kunt aanboren met AI.



# De AI-Code Code

1. Ook AI-systemen maken fouten of zeggen dingen die niet waar zijn.
2. Snap je de code niet, laat de AI dan uitleggen wat de code zou moeten doen en check of dat logisch is.
3. Als je informatie verzameld met code geschreven door een AI, verifieer die informatie dan.
4. Jij als journalist blijft altijd eindverantwoordelijk voor dat wat je publiceert.



gegenererde afbeelding

# OEFENING 1

-

# GOOGLE DORKING

# Google Dorking

Google Dorking is zoeken op Google met complexe zoekopdrachten.

Hiermee kun heel specifiek zoeken naar bepaalde informatie of documenten.

## Note

There should be no space between the operator and your search term. Eg. `inurl:directory`

## Basic Operators

**allinanchor:** results containing links with *term* in anchor text.

**allintext:** results containing all text matching *term*

**allintitle:** results containing title text matching *term*

**allinurl:** results containing URL matching *term*

**author:** newsgroup articles by specified author

**cache:** cached version of a webpage

**define:** definition of matching *term*

**filetype:** results matching filetype *term*

**group:** articles from specific groups/subareas

**inanchor:** results with anchor text links pointing to page matching *term*

**info:** information about corresponding page

**intext:** results containing text matching *term*

**intitle:** results containing *term* in title

**inurl:** results containing *term* in URL

**link:** pages that point to specified URL

**location:** articles from a specified location

**phonebook:** all public telephone listings for a person

**related:** pages similar to specified page

**site:** pages from specified site

## Overview

### Search Laboratory Blog

Following continued growth and success, Search Laboratory  
[www.searchlaboratory.com/blog/](http://www.searchlaboratory.com/blog/) - Cached - Similar

Title

URL

Cache

Meta Description

Related

## Special Characters

**+** results with *term*

**-** results without *term*

**~** results with *term* or a synonym of *term*

**OR** either or both *terms*

**..** results with a number in specified range

**\*** wildcard - can stand for any *term*

**""** exact *terms* or phrase in specific order

## Directories

`inurl:directory + keyword(s)`

`intitle:directory + keyword(s)`

`inurl:directory + keyword(s) -nofollow`

`keyword(s) intitle:"submit * url"`

# Google Dorking

**Opdracht:** Ga naar regiohelden.zapier.app en laat de AI een zoekopdracht voor je schrijven waarmee je alle pdf's vindt die gepubliceerd zijn op de website overheid.nl en waar 'stikstofreductie' in de titel staat.

Voer de verkregen opdracht (query) vervolgens in op google.com.

# Google Dorking

- Wie had er 4 resultaten?
- Wie geen 4 resultaten had, wat was de formule die je kreeg?
- En hoe heb je je vraag geformuleerd?

Ongeveer 4 resultaten (0,24 seconden)



overheid.nl

<https://open.overheid.nl> › documenten › pdf

[PDF](#)

⋮

## Programma Stikstofreductie en Natuurver

1 jul 2023 — 1.1. Aanleiding van dit programma. De neers  
de natuur is in Nederland al decennialang te.



overheid.nl

<https://open.overheid.nl> › documenten › pdf

[PDF](#)

⋮

## Programma Stikstofreductie en Natuurver

16 jul 2021 — 1.1 Aanleiding voor het programma. In Ned  
sprake van een te hoge stikstofbelasting op een groot aan



overheid.nl

<https://open.overheid.nl> › repository › pdf › conc...

[PDF](#)

⋮

## concept-besluit-tot-wijziging-van-enkele-

4 dec 2020 — Op de voordracht van Onze Minister van La  
Voedselkwaliteit, nr. WJZ.....; Gelet op richtlijn 92/4  
21 ...



overheid.nl

<https://open.overheid.nl> › documenten › pdf

[PDF](#)

⋮

## 1 Wijziging van de Wet natuurbeschermi

Het kabinet heeft in december 2019 aangekondigd aan ee  
stikstofproblematiek te werken. Kern van deze aanpak is n

# Google Dorking

**Opdracht:** Laat nu een query schrijven die alleen dat documenten vindt waarbij naast de vorige voorwaarden in het document ook de woorden ‘coalitieakkoord’ en ‘klimaataKKoord’ voor komen.

# Google Dorking

Hoe heb je dit aangepakt?

**Tip:** Je kunt dus een vervolgvraag stellen aan ChatGPT. Je hoeft de originele vraag niet te herhalen.

<https://gijn.org/resource/tips-for-optimizing-google-search-investigations-from-online-expert-henk-van-ess/>

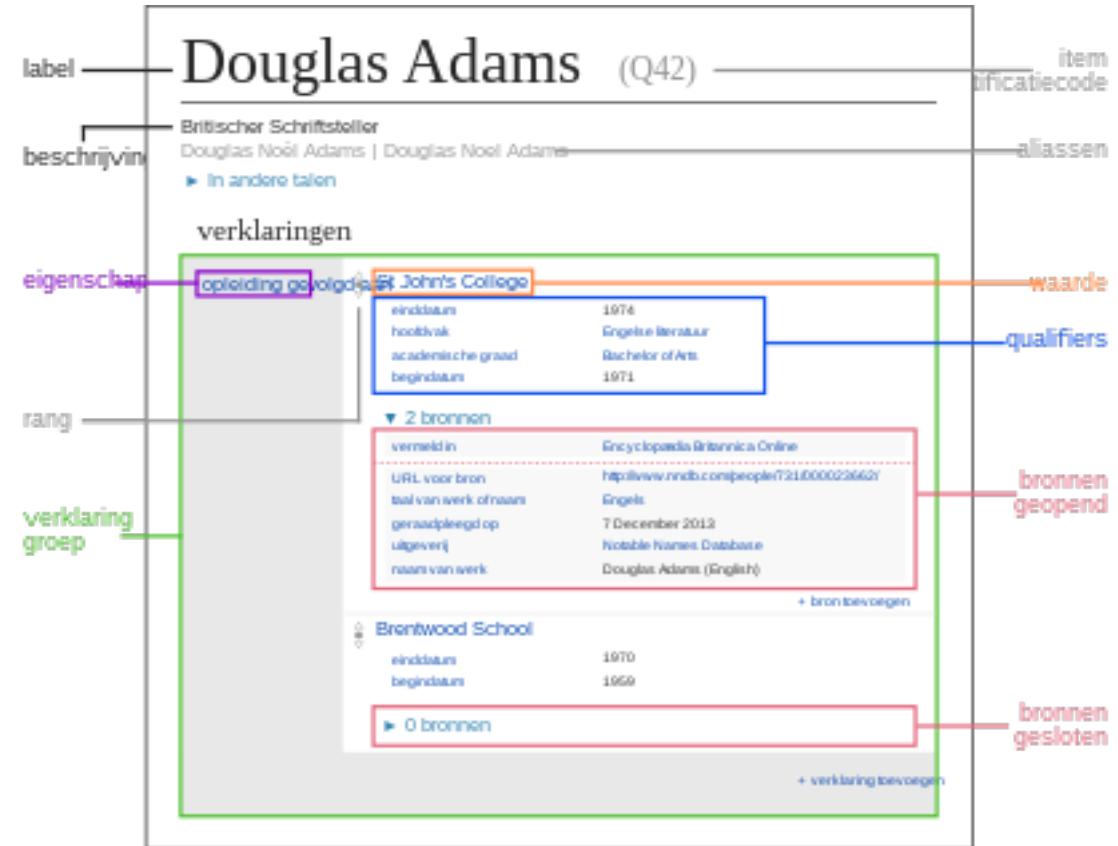
OEFENING III

-

WIKIDATA

# Wikidata

- Wikidata is een database met gestructureerde data van zo ongeveer alle informatie die ook via Wikipedia te vinden is.
- Wikidata is een mooie bron om onderzoek te doen.
- Bijvoorbeeld naar bekende plekken of personen.



# SPARQL

- Informatie uit de wikidatabase vraag je op met een SPARQL-query.
- Het schrijven van die query's is complex. Dat doen we dus met AI.
- Je voert die query vervolgens in op <https://query.wikidata.org/>

The screenshot shows the Wikidata Query Service interface. At the top, there is a logo consisting of vertical bars in red, green, and blue, followed by the text "Wikidata Query Service". On the right side of the header is a menu icon. Below the header, there is a code editor area containing a SPARQL query:

```
1 SELECT (COUNT(?article) AS ?count) WHERE {  
2   ?article schema:about wd:Q57792;  
3   schema:isPartOf [ wikibase:wikiGroup "wikipedia" ].  
4 }
```

To the left of the code editor is a vertical toolbar with several icons: a blue info icon, a blue cross icon, a blue dropdown arrow icon, a blue diamond icon, a blue folder icon, a blue circular arrow icon, a blue trash bin icon, and a blue link icon. At the bottom of the code editor are two large buttons: a blue play button with a white triangle pointing right, and a grey square button.

# Mark Rutte

**Opdracht:** Achterhaal met een SPARQL-query wat het geslacht van Mark Rutte is en wanneer hij is geboren.

**Tip:** Ververs eerst de chatbot pagina om met een nieuwe conversatie te beginnen.

**Vraag:** Wat was de uitkomst?

Prime Minister of the Netherlands since 2010  
M. Rutte | Prime Minister Rutte

▼ In more languages  
[Configure](#)

Language	Label	Description	Also known as
English	Mark Rutte	Prime Minister of the Netherlands since 2010	M. Rutte Prime Minister Rutte
Dutch	Mark Rutte	minister-president van Nederland sinds 2010	premier Rutte M. Rutte minister-president Rutte
German	Mark Rutte	niederländischer Politiker (VVD), Ministerpräsident Niedersachsen	Ministerpräsident Rutte
French	Mark Rutte	Premier ministre des Pays-Bas depuis 2010	

[All entered languages](#)

## Statements

instance of	human	<a href="#">edit</a>
	▶ 1 reference	<a href="#">+ add value</a>

<a href="#">image</a>		<a href="#">edit</a>
	Mark Rutte, 23.03.23 (cropped).jpg	
	1,738 × 2,471; 1.75 MB	

<a href="#">signature</a>		<a href="#">edit</a>
	Accession Treaty 2011 Mark Rutte signature.svg	
	404 × 152; 13 KB	

# Meneer de President

Iedere entiteit (persoon) in Wikidata heeft een eigen code en iedere eigenschap (geslacht) ook. Die codes kun je zien op [wikidata.nl](https://wikidata.nl).

De code voor ‘Prime Minister of the Netherlands’ is **Q3058109**

**Opdracht:** achterhaal hoeveel mannelijke en vrouwelijke minister-presidenten Nederland ooit gehad heeft.

*Het werkt soms beter als je de wikidata code meegeeft in je vraag.*

# Eindbaas boven eindbaas

Met SPARQL kun je niet alleen bepaalde zaken opzoeken, of optellen, maar je kunt ook ingewikkelder query's laten schrijven die zaken voor je uitrekenen.

**Opdracht:** kun je achterhalen hoe lang iedere minister-president aan de macht is geweest?

**Vraag:** Wie was de langst zittende premier? Ontbreekt er iemand in het rijtje?

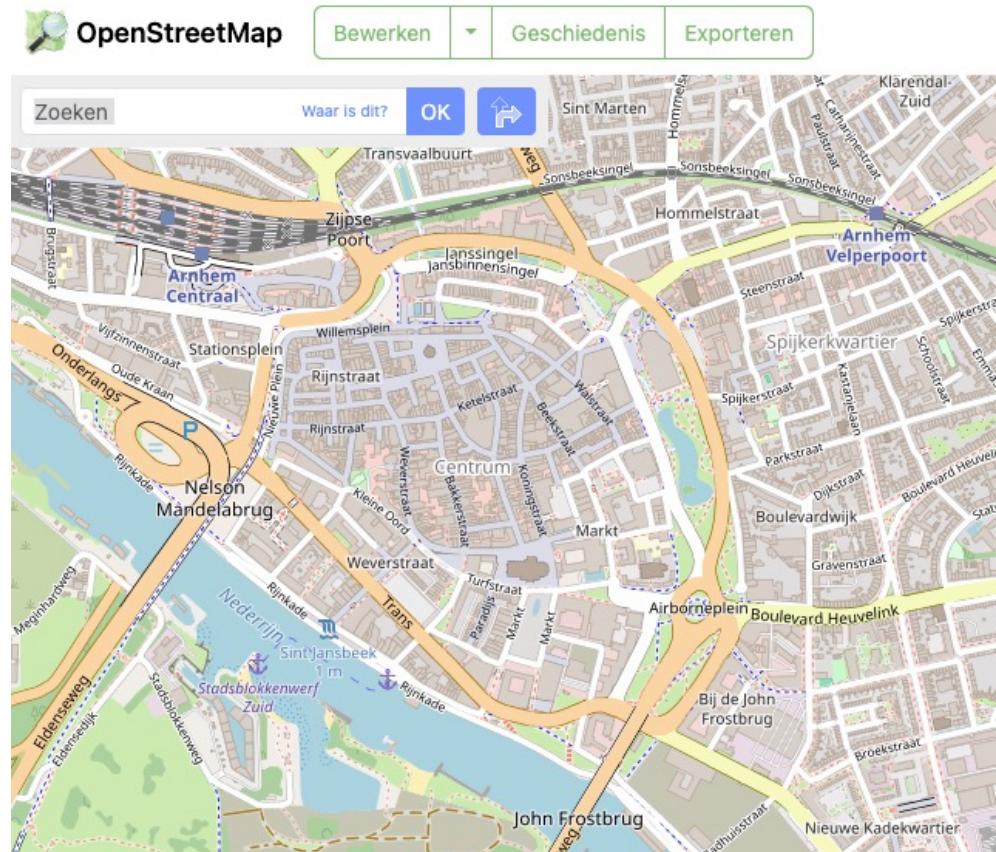
OEFENING III

-

OPEN STREET MAP

# Open Street Map

- Open Street Map is een open source alternatief voor Google Maps.  
[www.openstreetmap.org](http://www.openstreetmap.org)
- Het wordt beheert door vrijwilligers, vergelijkbaar met Wikipedia.
- Omdat het open source is, is de informatie vrij te gebruiken.



# Overpass Query Language

- Met Overpass Query Language kun je via Overpass Turbo zoeken op data die beschikbaar is in Open Street Map.
- Maar ook het gebruik van de Overpass Query Language is complex.
- Dus maken we gebruik van AI.

```
1 [out:json][timeout:25];
2 // zoek gebied "Nederland"
3 area["name"]=="Nederland"]->.searchArea;
4 // verzamel resultaten
5 (
6     // query deel voor: "shop=cannabis"
7     node["shop"]=="cannabis"](.area.searchArea)
8     way["shop"]=="cannabis"](.area.searchArea)
9     relation["shop"]=="cannabis"](.area.searchArea)
10 )->.allCannabisShops;
11 // zoek cannabiswinkels binnen 500m van elke winkel
12 (
13     node(around.allCannabisShops:500)["shop"]
14     way(around.allCannabisShops:500)["shop"]
15     relation(around.allCannabisShops:500)[]
16 );
17 // print resultaten
18 out body;
19 >;
20 out skel qt;
```

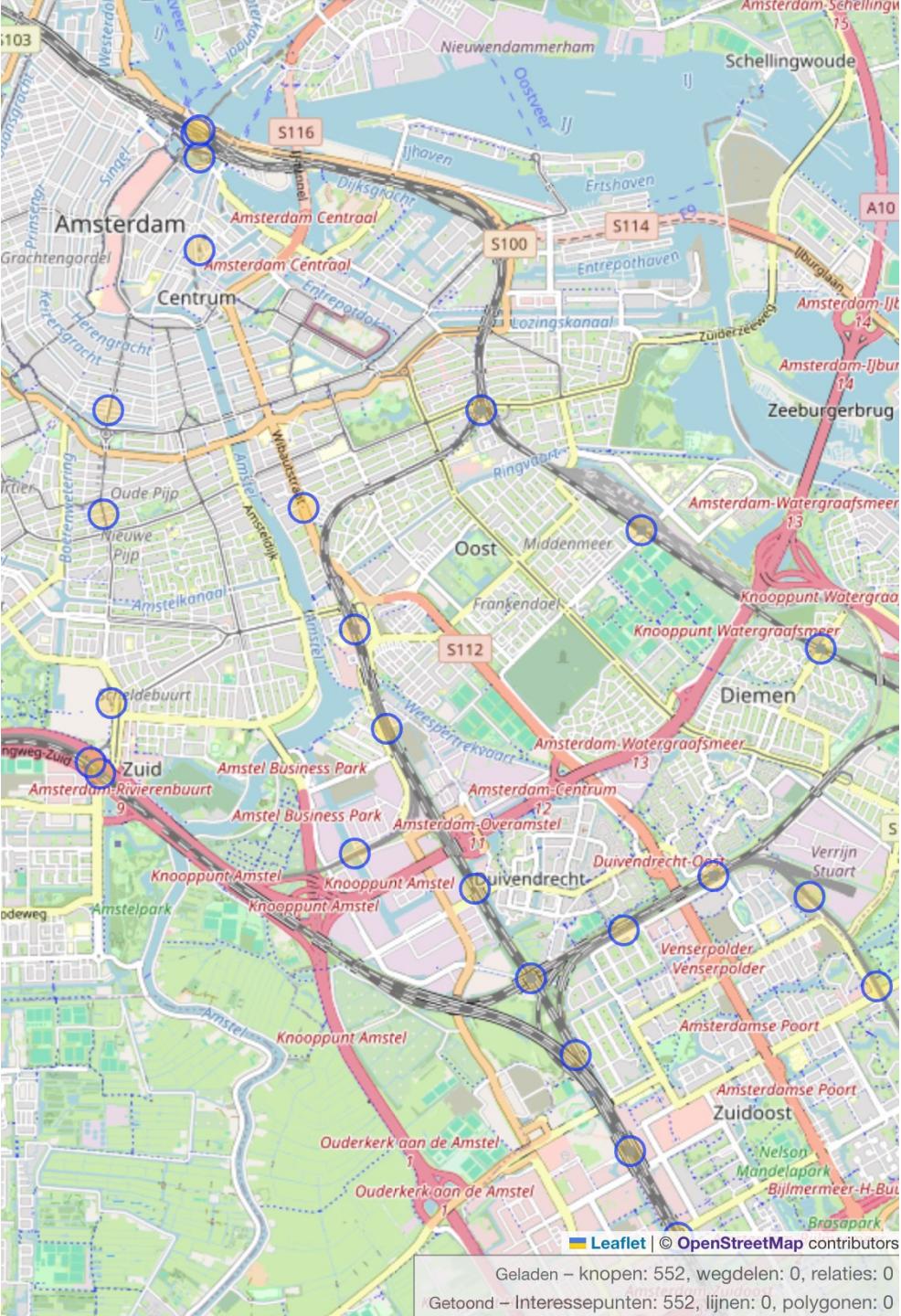
# Open Street Map

**Opdracht:** Laat ChatGPT een zoekopdracht voor je schrijven waarmee je op Overpass Turbo alle treinstations in Nederland kunt vinden en voer die query uit op [overpass-turbo.eu](https://overpass-turbo.eu)

# Open Street Map

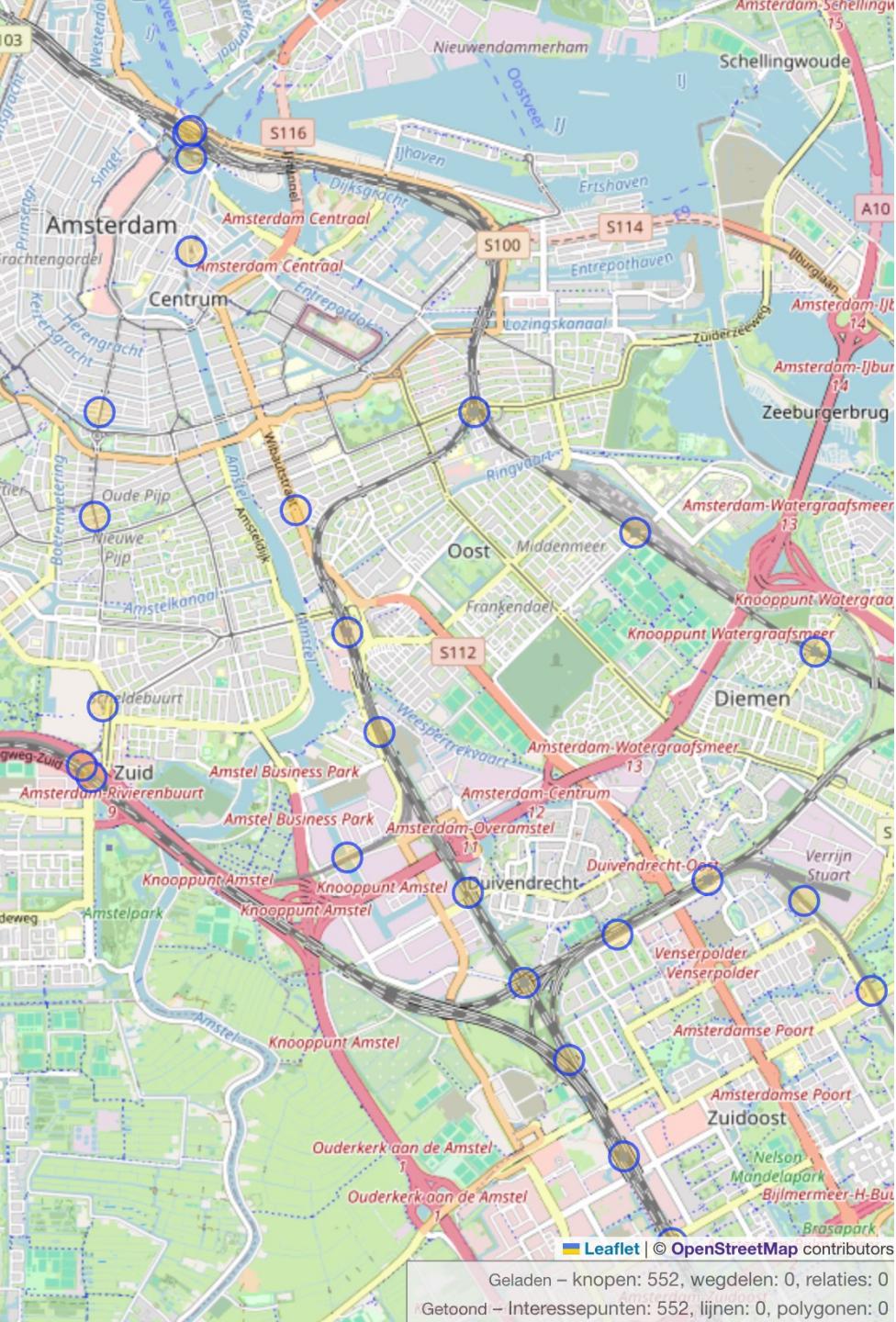
## Vragen:

- Hoeveel resultaten kreeg je?  
*dat kun je rechtsonder zien bij  
'interessepunten'*
- Hoeveel resultaten zou je je  
moeten krijgen?
- Wat zou er mis kunnen zijn?



# Open Street Map

- Kijk op overpass-turbo op de kaart en zoek een station waarvan je zeker weet dat het een treinstation is.
- Klik op de cirkel van dat station.
- Zie je info die je kan helpen bij het verfijnen van je vraag?



# Open Street Map

**Opdracht:** Je weet nu dat een je treinstation kunt herkennen aan “train=yes”

Laat AI een zoekopdracht voor je schrijven waarmee je op Overpass-Turbo alleen de treinstations in Nederland kunt vinden en voer die query uit op

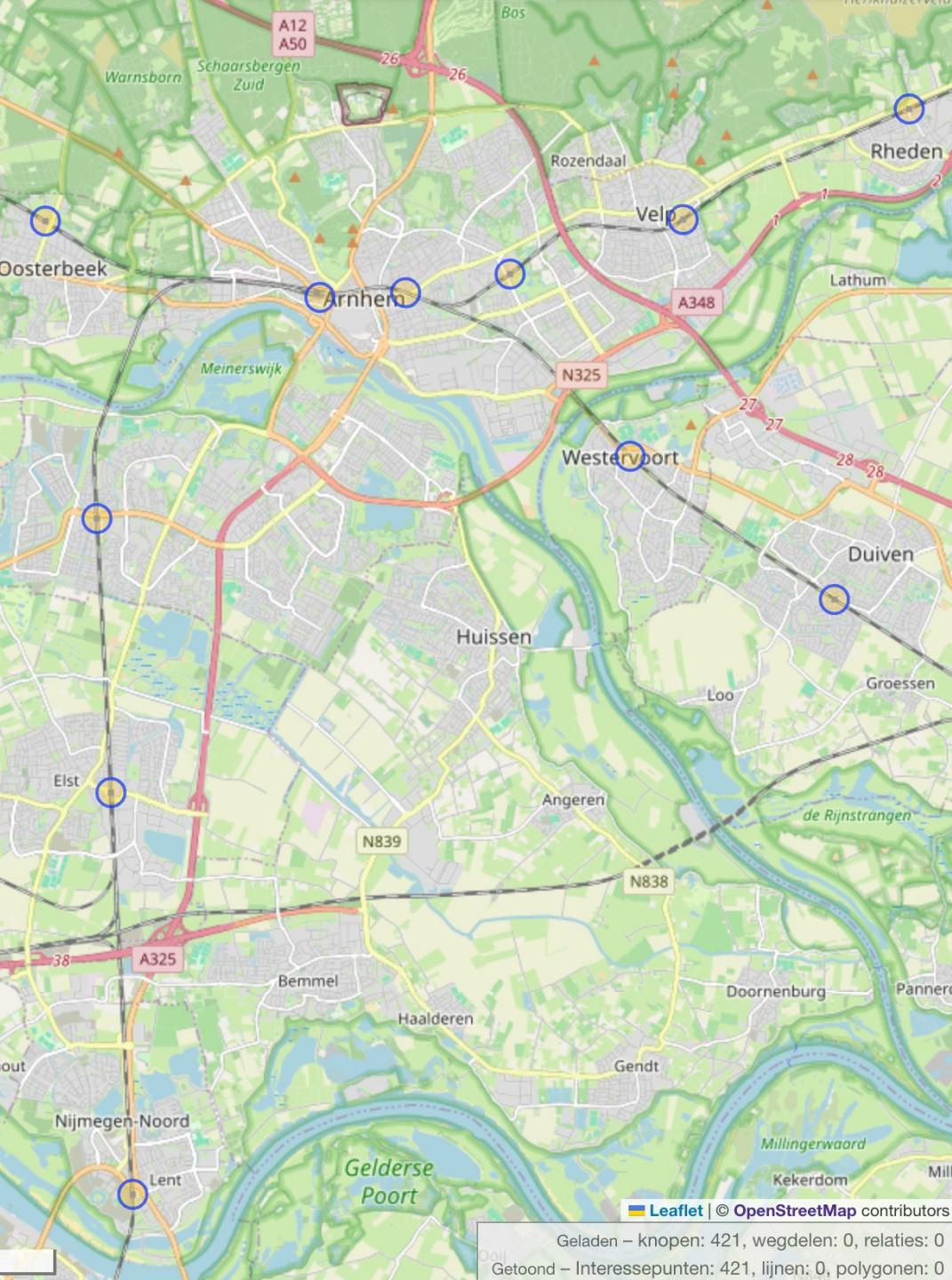
[overpass-turbo.eu](https://overpass-turbo.eu)

**Tip:** Soms moet je de AI op weg helpen door zelf iets op te zoeken en je vraag te specifiëren.

# Open Street Map

## Vragen:

- Hoeveel resultaten kreeg je nu?
- Wijkt dat nog af van wat je verwacht?
- Wat zou een verklaring kunnen zijn?



# OEFENING IV

---

## PYTHON EN GOOGLE COLAB

# Python

- Python is een programmeertaal die veel gebruikt wordt door datajournalisten.
- Python is heel erg geschikt om code te schrijven die met het internet communiceert.
- ChatGPT is heel goed in staat om Python code voor je te schrijven.

```
import pandas as pd
import requests
import io
from google.colab import files

def download_files(ecli_codes):
    base_url = "https://uitspraken.rechtspraak.nl/api/pdfdownload/"

    for code in ecli_codes:
        response = requests.get(base_url + code)

        with open(f'{code}.pdf', 'wb') as file:
            file.write(response.content)

def main():
    uploaded = files.upload()

    for fn in uploaded.keys():
        print('User uploaded file "{name}" with length {length} bytes'
              .format(name=fn, length=len(uploaded[fn])))

        df = pd.read_csv(io.StringIO(uploaded[fn].decode('utf-8')))
        ecli_codes = df['ECLI'].tolist()

        download_files(ecli_codes)

main()
```

# Kaart = Locatie

Je hebt net met Open Street Map de locaties van alle treinstations in Nederland gevonden.

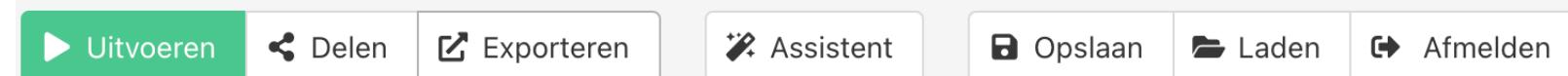
Wat ik zou willen weten is wat het meest noordelijke, oostelijke, zuidelijke en westelijke station van Nederland is.



# GeoJSON

In Open Street Map kun je met de knop Exporteren (bovenaan) een GeoJSON bestand opslaan.

In dit bestand staat alle info die Open Street Map heeft over de gevonden entiteiten plus de coördinaten. Je weet dus de locatie.



## Exporteren

### Gegevens

GeoJSON	<a href="#">download</a>	<a href="#">kopieer</a>
GPX	<a href="#">download</a>	saves the exported data as a file
KML	<a href="#">download</a>	<a href="#">kopieer</a>
ruwe OSM-gegevens	<a href="#">download</a>	<a href="#">kopieer</a>

ruwe gegevens rechtstreeks van [Overpass API](#)

gegevens in een OSM-editor laden: [JOSM, Level0](#)

### Kaart

als [png afbeelding](#)

als [Interactieve Kaart](#)

huidige [kaart weergave](#) (bbox, midden, enz...)

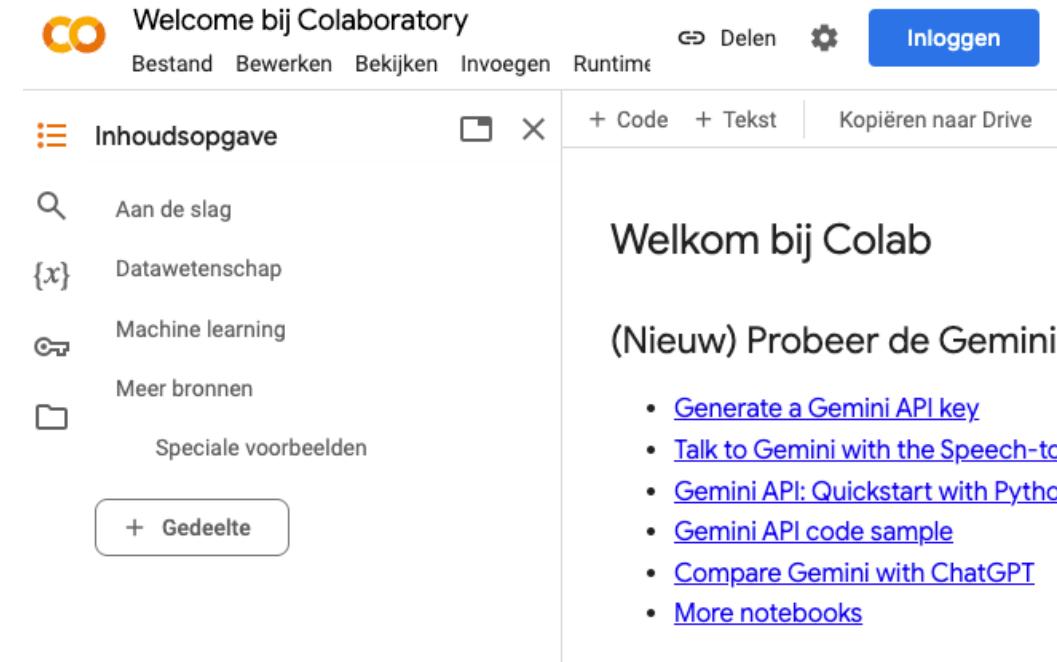
# Python

**Opdracht:** Laat ChatGPT, Python code voor je schrijven om te achterhalen wat het meest noordelijk, oostelijk, zuidelijk en westelijk gelegen station van Nederland is.



# Google Colab

- Je hebt nu code maar om die code uit te voeren heb je een omgeving nodig.
- Google Colab is een gratis omgeving waar je Python code kunt ‘runnen’.
- Ga naar [colab.research.google.com](https://colab.research.google.com) en log in met je Google Account.
- Kies onder ‘bestand’ voor ‘nieuw notebook’.



# Plak en Play

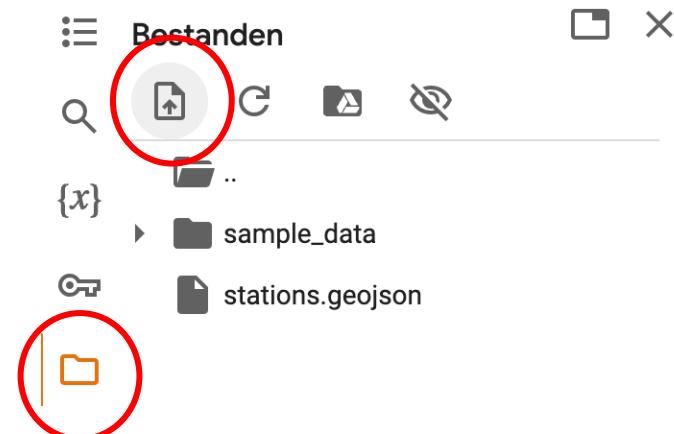
- Je krijgt dan dit scherm te zien.
- Je code plak je in het grijze veld rechts van het ‘play’-icoon.
- Om je code te uit te voeren, druk je op ‘play’.
- Maar je moet eerst het GeoJSON bestand uploaden.



The screenshot shows a Jupyter Notebook interface with the title bar 'Untitled1.ipynb'. The menu bar includes 'Bestand', 'Bewerken', 'Bekijken', 'Invoegen', 'Runtime', 'Tools', 'Hulp', and 'Alle wijzigingen zijn opgeslagen'. On the left is a sidebar with icons for '+ Code', '+ Tekst', a search bar, and file/folder navigation. The main area contains a code cell starting with '%pyodbc' followed by a placeholder text box containing the instruction 'HIER PLAK JE STRAKS JE CODE' with a play button icon.

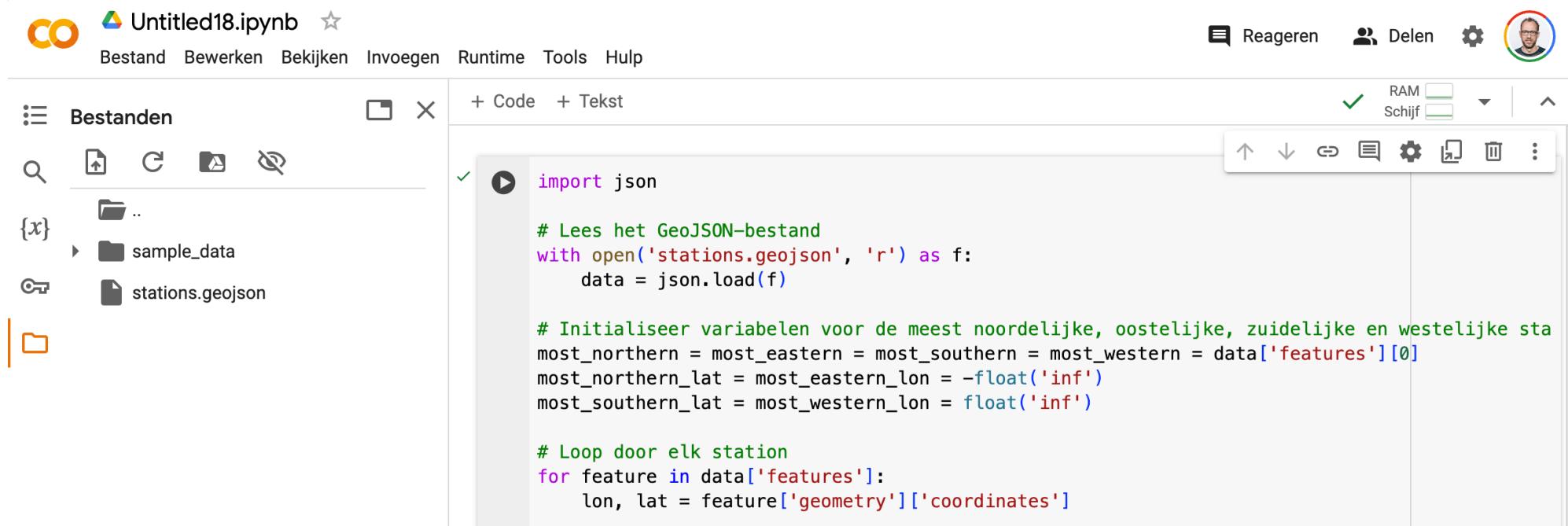
# Beam me up Scotty...

- Het GeoJSON bestand uploaden in Google Colab kan door links op de folder te klikken.
- Vervolgens op het ‘blaadje met het pijltje omhoog’
- Zorg er wel voor dat je bestandsnaam hetzelfde is als de naam die in je script staat.  
In dit geval stations.geojson
- Als opdracht III niet gelukt is staat er ook een geojson-bestand op de Github website.



# Daar gaan we.

Je zou nu zoiets moeten hebben. Links een geojson bestand en rechts een stuk python code. Druk nu op play!



The screenshot shows a Jupyter Notebook interface with the following elements:

- Title Bar:** Untitled18.ipynb
- File Browser (Left):** Shows a folder structure with files: .., sample\_data, and stations.geojson.
- Code Editor (Right):** Displays Python code for reading a GeoJSON file and initializing variables for station coordinates.
- Toolbar:** Includes buttons for Reageren (Comment), Delen (Share), and profile settings.
- Runtime Tab:** Shows RAM usage (RAM: 1.2 GB, Schijf: 1.2 GB).

```
import json

# Lees het GeoJSON-bestand
with open('stations.geojson', 'r') as f:
    data = json.load(f)

# Initialiseer variabelen voor de meest noordelijke, oostelijke, zuidelijke en westelijke sta
most_northern = most_eastern = most_southern = most_western = data['features'][0]
most_northern_lat = most_eastern_lon = -float('inf')
most_southern_lat = most_western_lon = float('inf')

# Loop door elk station
for feature in data['features']:
    lon, lat = feature['geometry']['coordinates']
```

# Uitkomst

Als alles goed gaat zou zo iets je uitkomst moeten zijn.

Most Northern Station: Eemshaven

Most Eastern Station: Bad Nieuweschans

Most Southern Station: Eijsden

Most Western Station: Vlissingen Souburg

## Vragen:

- Wie kreeg een foutmelding? Wat heb je toen gedaan?
- Wie heeft een andere uitkomst? Wat ging er mis?

OEFENING V

-

API SCRAPEN

# Wat is een API

- API staat voor Application Programming Interface
- Een API is een manier voor computers om met elkaar te praten.
- API's zijn gestructureerd en daarom zeer geschikt om data te verzamelen.
- In deze API van Schiphol bijvoorbeeld staat voor ieder 'security filter' de actuele wachttijd vermeld.

```
{  
    "AF1": {  
        "updated": "2024-04-09T06:11:00.000Z",  
        "waitTimeInSeconds": 153,  
        "waitTimeOutOfRange": false  
    },  
    "AF2": {  
        "updated": "2024-04-09T06:11:00.000Z",  
        "waitTimeInSeconds": 167,  
        "waitTimeOutOfRange": false  
    },  
    "AF3": {  
        "updated": "2024-04-09T06:11:00.000Z",  
        "waitTimeInSeconds": 1384,  
        "waitTimeOutOfRange": false  
    },  
    "AF4": {  
        "updated": "2024-04-09T06:11:00.000Z",  
        "waitTimeInSeconds": 252,  
        "waitTimeOutOfRange": false  
    },  
    "EF-T": {  
        "updated": "2024-04-09T06:11:00.000Z",  
        "waitTimeInSeconds": 473  
    },  
    "GH-T": {  
        "updated": "2024-04-09T06:11:00.000Z",  
        "waitTimeInSeconds": 138  
    },  
    "NS-S": {  
        "updated": "2024-04-09T06:11:00.000Z",  
        "waitTimeInSeconds": 1520  
    },  
    "S-NS": {  
        "updated": "2024-04-09T06:11:00.000Z",  
        "waitTimeInSeconds": 300  
    },  
    "VF0": {  
        "updated": "2024-04-09T06:11:00.000Z",  
        "waitTimeInSeconds": 60  
    }  
}
```

# API scraping

- Wat we gaan doen is AI een Python script laten schrijven om de wachttijden op te slaan in een CSV bestand.
- CSV staat voor Comma Separated Values en is een manier om data op te slaan zodat je er iets mee kunt in een spreadsheet. (Excel)
- Het verzamelen en opslaan van data van een API of website noemen we *scraping*. Engels voor schrapen.

```
{  
    "AF1": {  
        "updated": "2024-04-09T06:11:00.000Z",  
        "waitTimeInSeconds": 153,  
        "waitTimeOutOfRange": false  
    },  
    "AF2": {  
        "updated": "2024-04-09T06:11:00.000Z",  
        "waitTimeInSeconds": 167,  
        "waitTimeOutOfRange": false  
    },  
    "AF3": {  
        "updated": "2024-04-09T06:11:00.000Z",  
        "waitTimeInSeconds": 1384,  
        "waitTimeOutOfRange": false  
    },  
    "AF4": {  
        "updated": "2024-04-09T06:11:00.000Z",  
        "waitTimeInSeconds": 252,  
        "waitTimeOutOfRange": false  
    },  
    "EF-T": {  
        "updated": "2024-04-09T06:11:00.000Z",  
        "waitTimeInSeconds": 473  
    },  
    "GH-T": {  
        "updated": "2024-04-09T06:11:00.000Z",  
        "waitTimeInSeconds": 138  
    },  
    "NS-S": {  
        "updated": "2024-04-09T06:11:00.000Z",  
        "waitTimeInSeconds": 1520  
    },  
    "S-NS": {  
        "updated": "2024-04-09T06:11:00.000Z",  
        "waitTimeInSeconds": 300  
    },  
    "VF0": {  
        "updated": "2024-04-09T06:11:00.000Z",  
        "waitTimeInSeconds": 60  
    }  
}
```

# Bekijk de API

Ga naar deze url die staat op de github pagina

**<https://www.schiphol.nl/api/proxy/v3/waittimes/security-filters>**

je krijgt dan de realtime data van alle security filters op schiphol te zien.

# Schrijf je verzoek aan de AI

**Opdracht:** Geef de AI de taak om een python script te schrijven dat de informatie uit de API opslaat in een csv bestand.

**Tip:** De AI kan zelf de structuur van de API niet bestuderen. Daarom is het handig om een deel van de API dadelijk te kopiëren en mee te geven in je vraag.

# Foutmelding!?

Als ik mijn code uitvoer komt er een uitroepteken naast mijn play-icoon en een foutmelding onder mijn code te staan.

Als dat gebeurt kopieer ik gewoon de foutcode en geef die terug aan de AI. Meestal komt die met een werkende oplossing.

The screenshot shows a Jupyter Notebook interface with the following code in a cell:

```
0 s + Code + Tekst
① ⚡ import requests
import csv
import json

# Maak een verzoek naar de API
response = requests.get("https://www.schiphol.nl/api/proxy/v3/waittimes/security-filters")

# Zet het antwoord om naar JSON
data = response.json()

# Open een nieuw CSV-bestand om de gegevens in te schrijven
with open('data.csv', 'w', newline='') as file:
    writer = csv.writer(file)

    # Schrijf de koppen van de kolommen
    headers = ['Filter', 'Updated', 'WaitTimeInSeconds', 'WaitTimeOutOfRange']
    writer.writerow(headers)

    # Schrijf de gegevens
    for filter_name, filter_data in data.items():
        row = [filter_name, filter_data['updated'], filter_data['waitTimeInSeconds'], filter_data['waitTimeOutOfRange']]
        writer.writerow(row)
```

An error message is displayed in a red-bordered box:

```
KeyError Traceback (most recent call last)
<ipython-input-13-d2244d74b784> in <cell line: 12>()
      19     # Schrijf de gegevens
      20     for filter_name, filter_data in data.items():
--> 21         row = [filter_name, filter_data['updated'], filter_data['waitTimeInSeconds'],
      22             filter_data['waitTimeOutOfRange']]
      23         writer.writerow(row)

KeyError: 'waitTimeOutOfRange'
```

At the bottom of the interface, it says "0 s voltooid om 09:19".

# Computer says yes!

The screenshot shows a code editor interface with a sidebar labeled "Bestanden" (Files) on the left. In the sidebar, three items are highlighted with red circles: a refresh icon at the top, a "data.csv" file icon, and a "sample\_data" folder icon below it. The main pane displays Python code. A green checkmark icon with "1s" next to it is circled in red, indicating the execution time of the script. The code itself performs an API request to Schiphol's API for security filters, converts the response to JSON, and then writes it to a CSV file named "data.csv".

```
import requests
import csv
import json

# Maak een verzoek naar de API
response = requests.get("https://www.schiphol.nl/api/proxy/v3/waittimes/security-filters")

# Zet het antwoord om naar JSON
data = response.json()

# Open een nieuw CSV-bestand om de gegevens in te schrijven
with open('data.csv', 'w', newline='') as file:
    writer = csv.writer(file)

    # Schrijf de koppen van de kolommen
    headers = ['Filter', 'Updated', 'WaitTimeInSeconds', 'WaitTimeOutOfRange']
    writer.writerow(headers)

    # Schrijf de gegevens
    for filter_name, filter_data in data.items():
        row = [filter_name, filter_data.get('updated'), filter_data.get('waitTimeInSeconds'),
writer.writerow(row)
```

Bestanden

RAM Schijf

Import requests, csv, json.

# Maak een verzoek naar de API

response = requests.get("https://www.schiphol.nl/api/proxy/v3/waittimes/security-filters")

# Zet het antwoord om naar JSON

data = response.json()

# Open een nieuw CSV-bestand om de gegevens in te schrijven

with open('data.csv', 'w', newline='') as file:

writer = csv.writer(file)

# Schrijf de koppen van de kolommen

headers = ['Filter', 'Updated', 'WaitTimeInSeconds', 'WaitTimeOutOfRange']

writer.writerow(headers)

data.csv

1 to 13 of 13 entries Filter

Filter	Updated	WaitTimeInSeconds	WaitTimeOutOfRange
AF1	2024-04-09T07:34:00.000Z	567	False
AF2	2024-04-09T07:34:00.000Z	930	False
AF3	2024-04-09T07:34:00.000Z	395	False
AF4	2024-04-09T07:34:00.000Z	78	False
EF-T	2024-04-09T07:34:00.000Z	60	False
GH-T	2024-04-09T07:34:00.000Z	60	False
NS-S	2024-04-09T07:34:00.000Z	594	False
S-NS	2024-04-09T07:34:00.000Z	425	False
VF0	2024-04-09T07:34:00.000Z	60	False
VF1	2024-04-09T07:34:00.000Z	850	False
VF2	2024-04-09T07:34:00.000Z	292	False
VF3	2024-04-09T07:34:00.000Z	257	False
VF4	2024-04-09T07:34:00.000Z	72	False

Show 50 per page

Schijf 83.30 GB beschikbaar

0 s voltooid om 09:35

# Wat is er ‘mis’ met de data in de API?

Hier rechts zie je realtime data van de schiphol API. Ziet iemand wat er ‘mis’ is?

**Opdracht:** Vraag de AI om het probleem op te lossen.

## Insert Web Page

This app allows you to insert secure web pages starting with https:// into the slide deck. Non-secure web pages are not supported for security reasons.

Please enter the URL below.

https:// www.schiphol.nl/api/proxy/v3/waitir

Note: Many popular websites allow secure access. Please click on the preview button to ensure the web page is accessible.

Preview

# Power-up

Je hebt nu een script dat op een gegeven moment de wachttijden opslaat.

Maar vaak wil je veranderingen over tijd kunnen zien.

**Opdracht:** Laat AI het script aanpassen om de data iedere minuut op te slaan.

Security

⌚ 5-10 mins

Volg de borden **Gate E21**



**ZONDER AI WAS  
DIT VERHAAL  
NOoit GEMAAKt**

☰ MENU



## HartslagNu is het reanimatie oproepsysteem van Nederland

Elke dag krijgen 40 mensen in Nederland buiten het ziekenhuis een hartstilstand. De overlevingskans is het grootst als iemand binnen 6 minuten reanimeert en een AED aansluit. Als elke minuut telt, is snelle hulp van levensbelang. Ook jij kunt helpen!

Ik word burgerhulpverlener

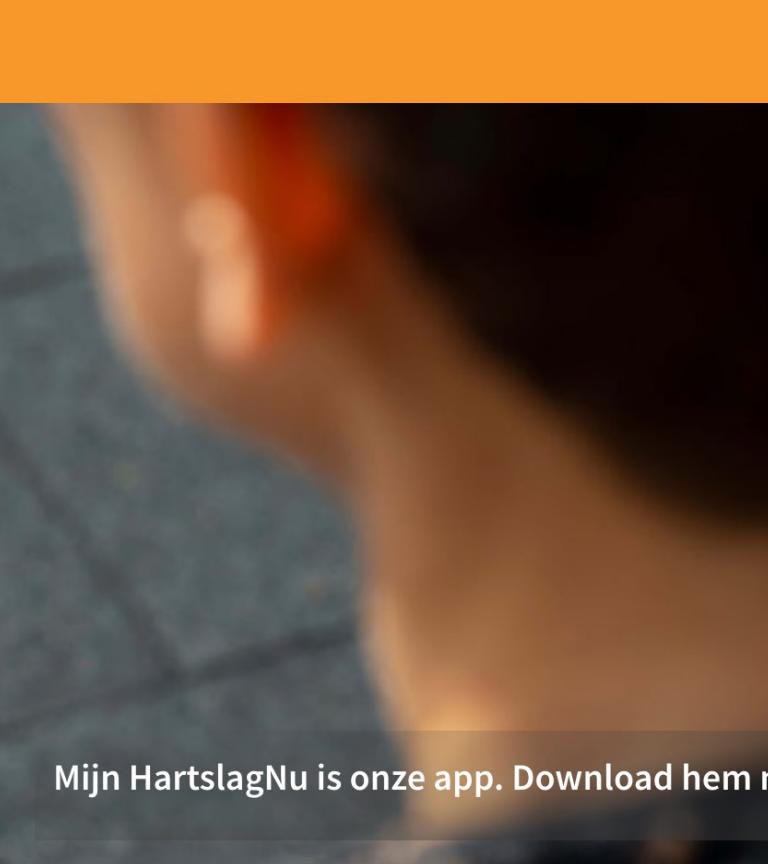
Ik meld een AED aan

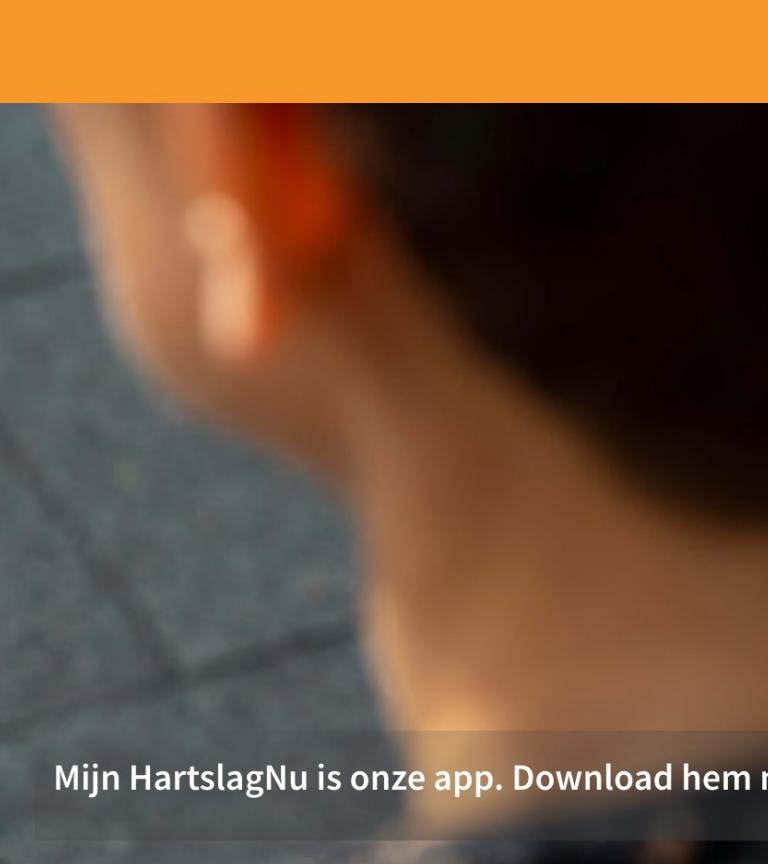


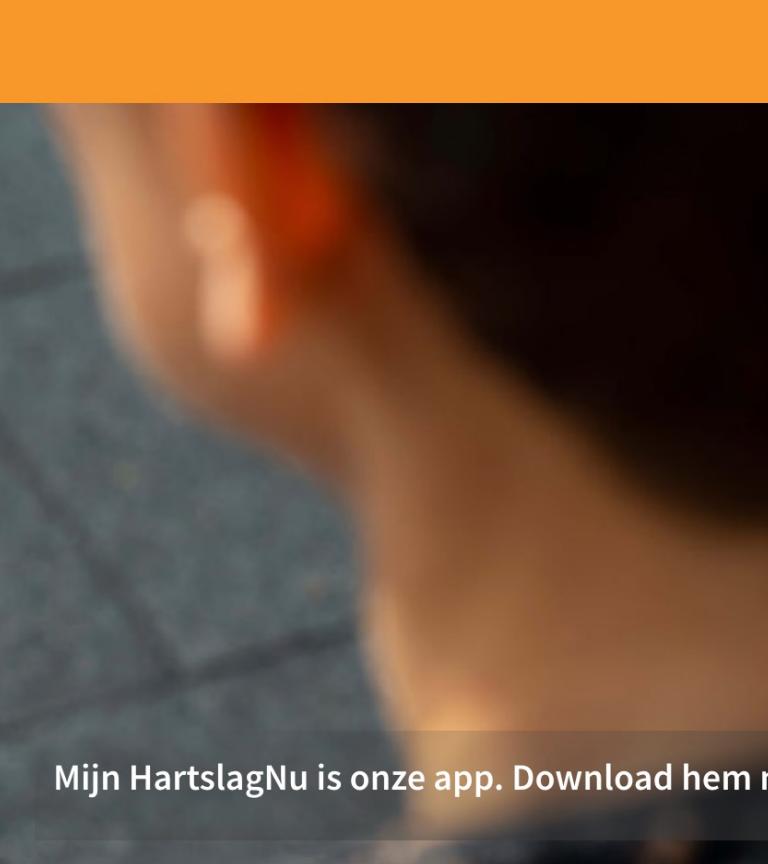
🔍Log in op Mijn HartslagNu

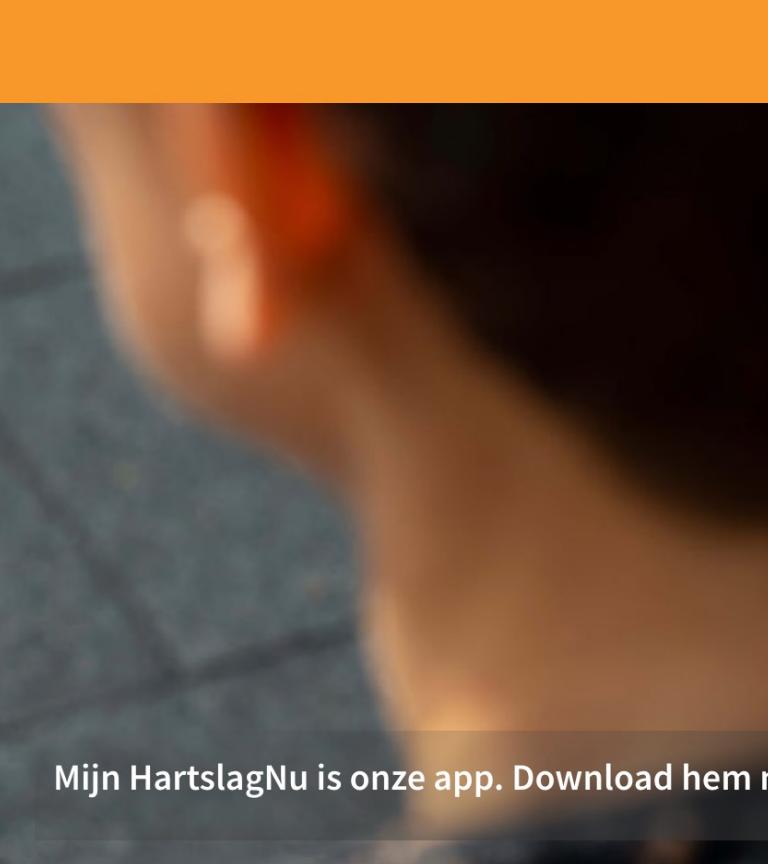
☰MENU

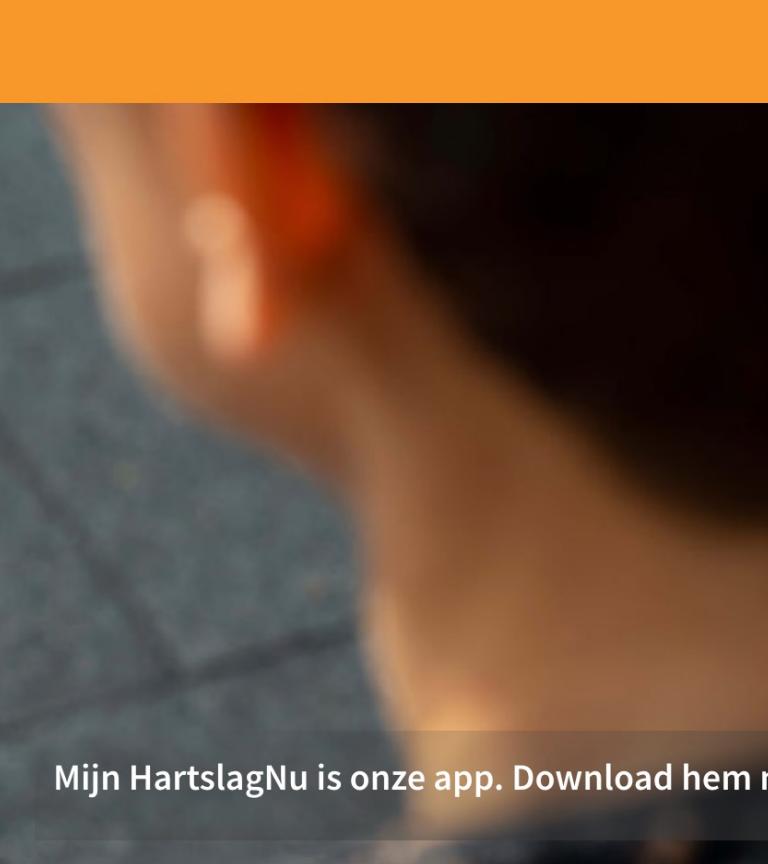
<>

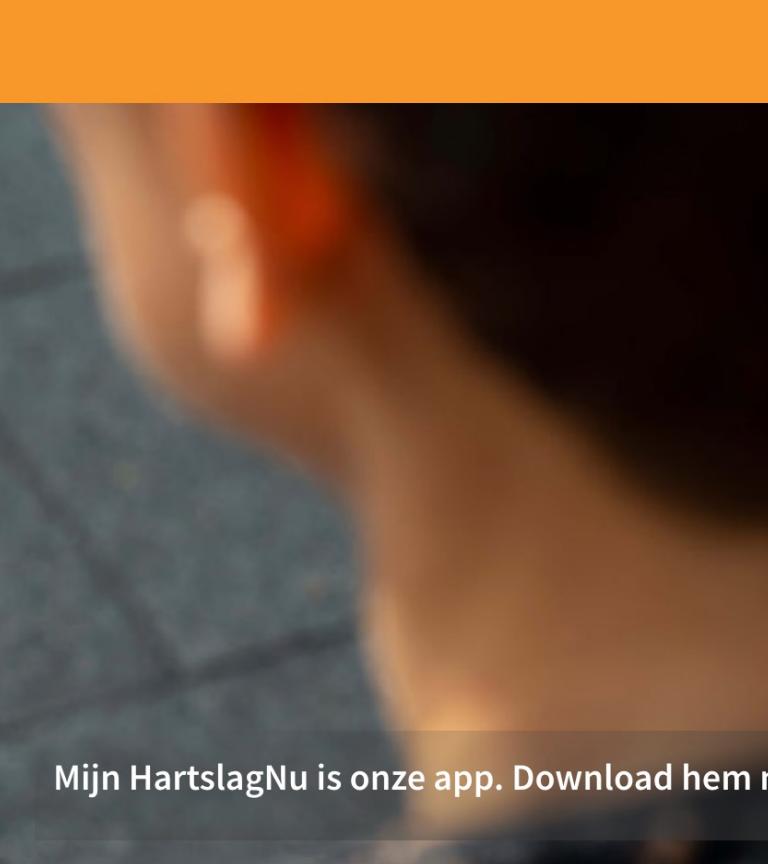
<>

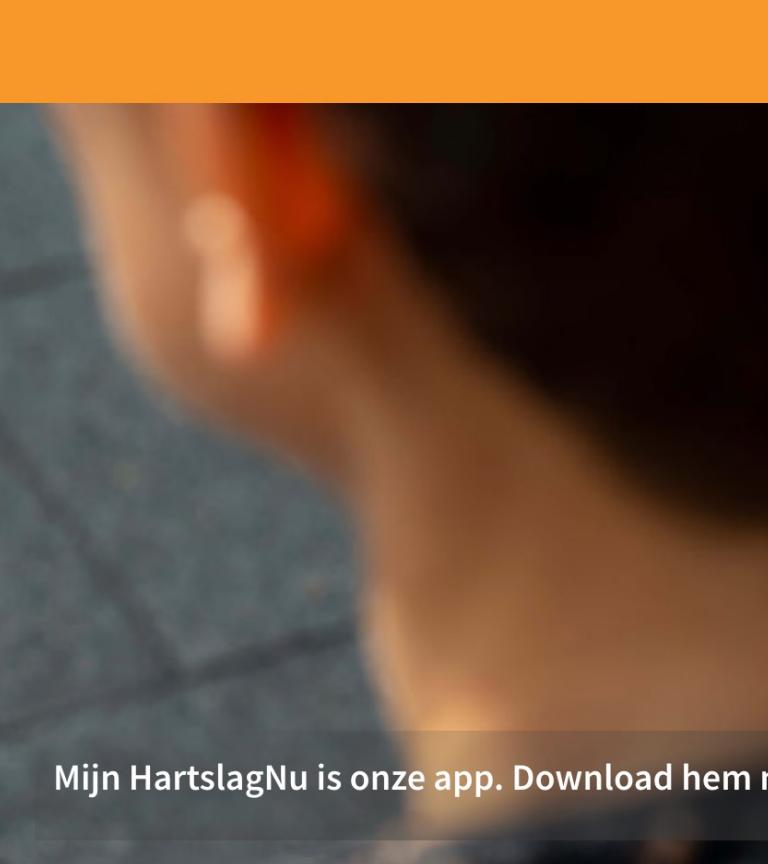
<>

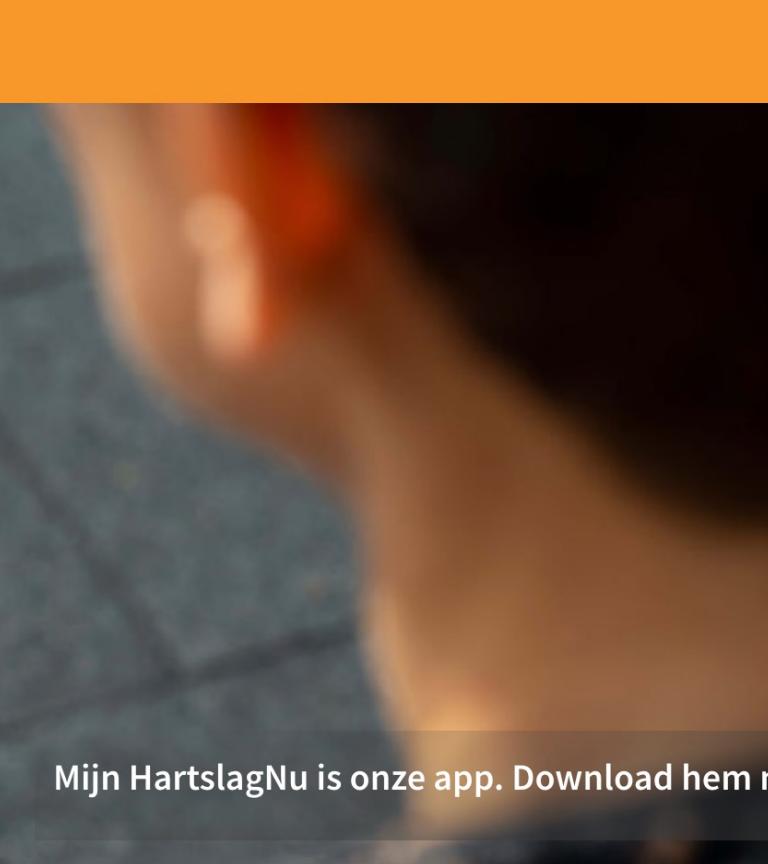
<>

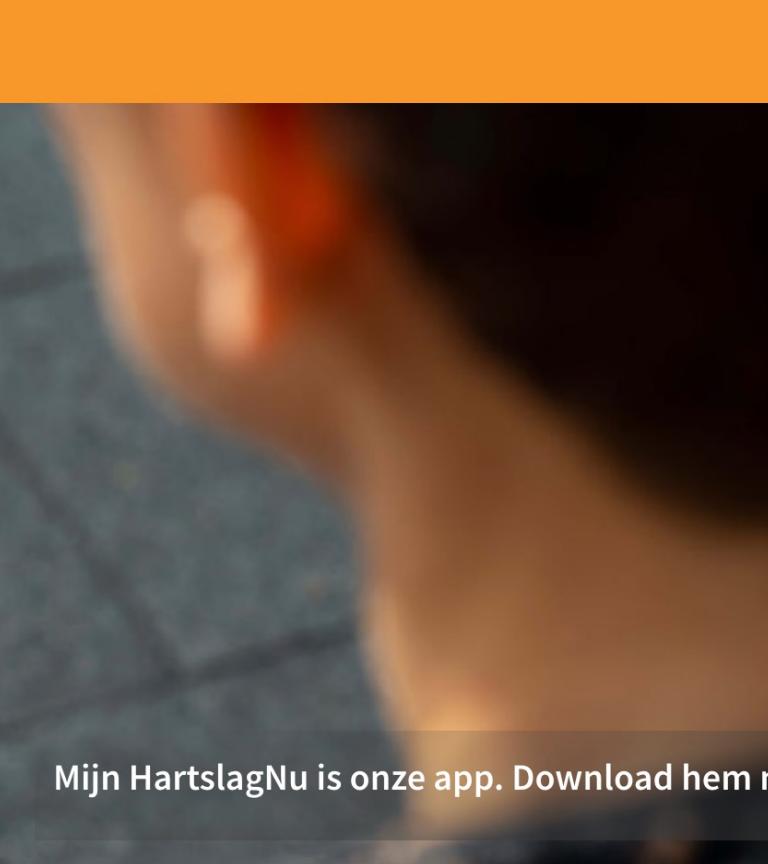
<>

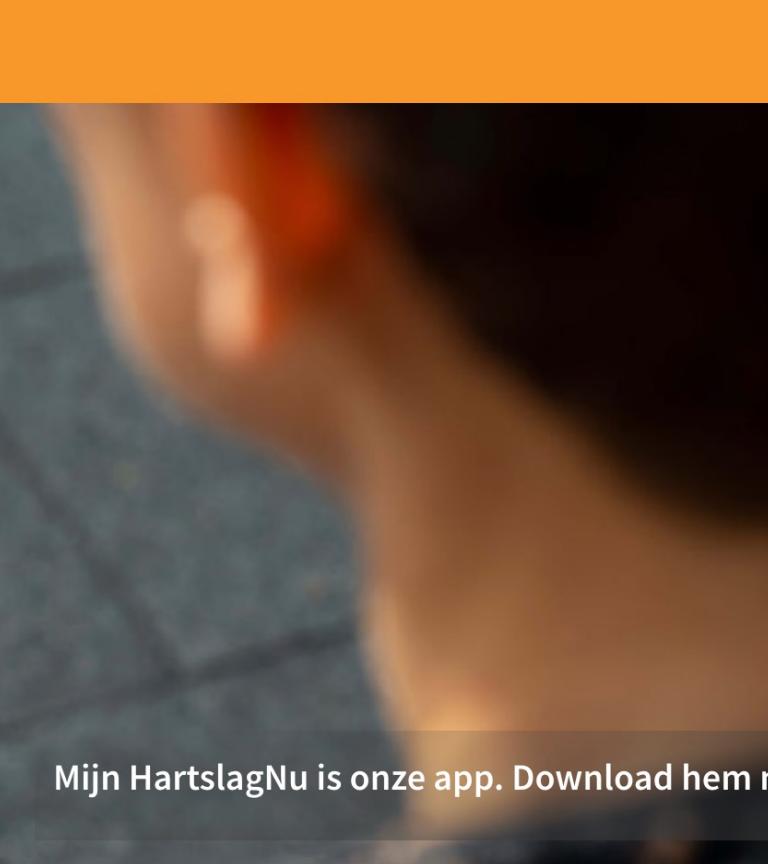
<>

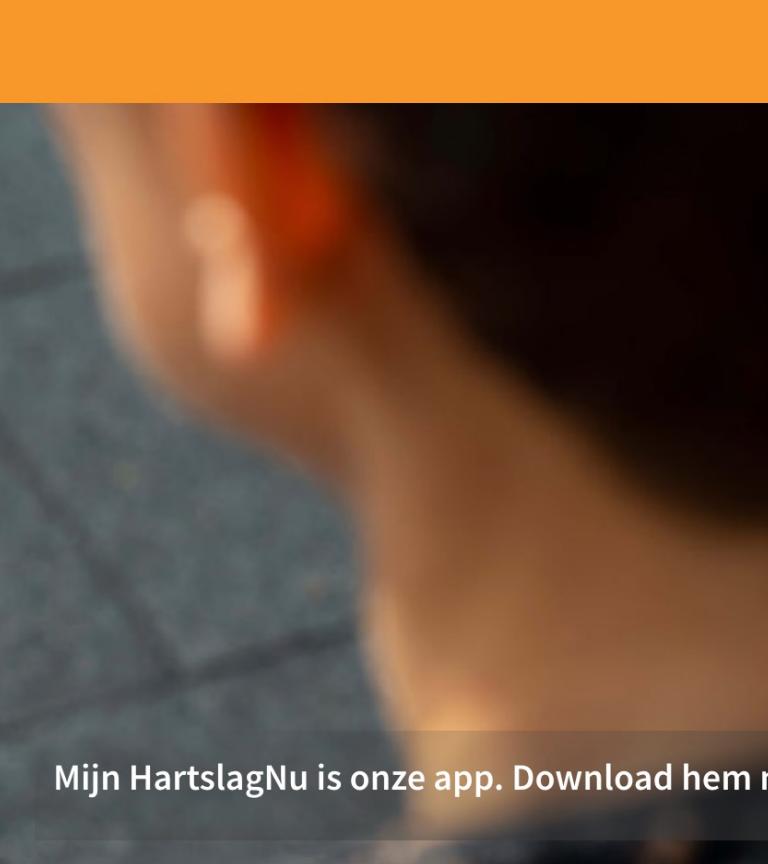
<>

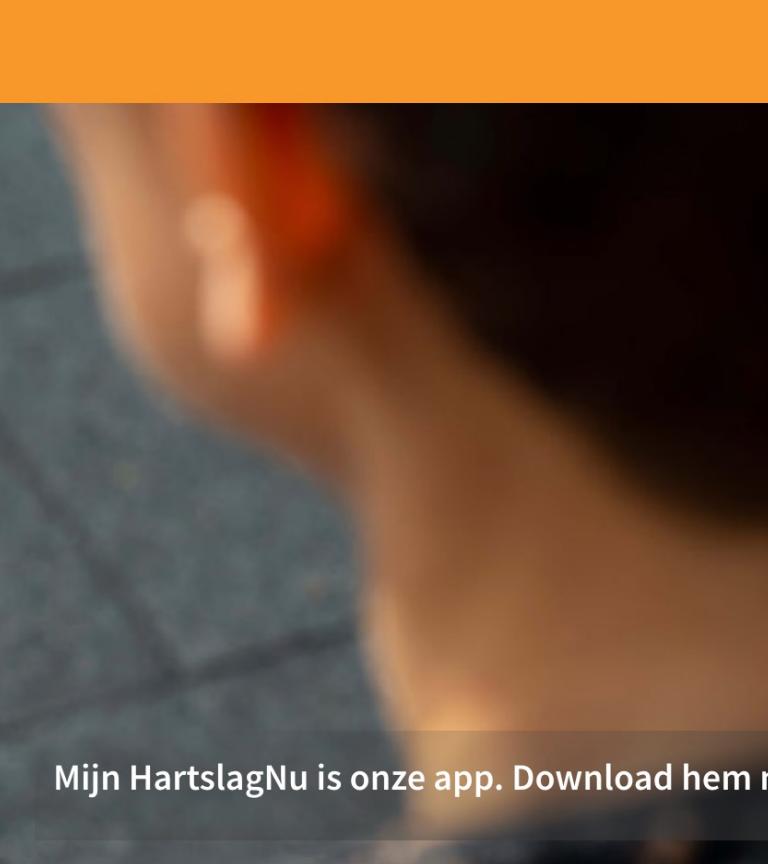
<>

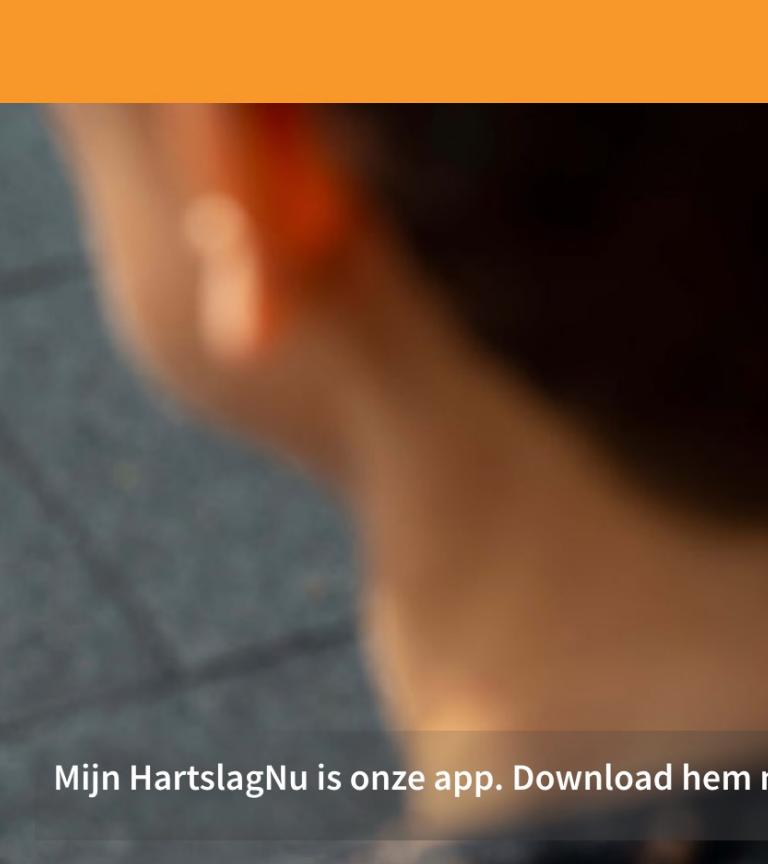
<>

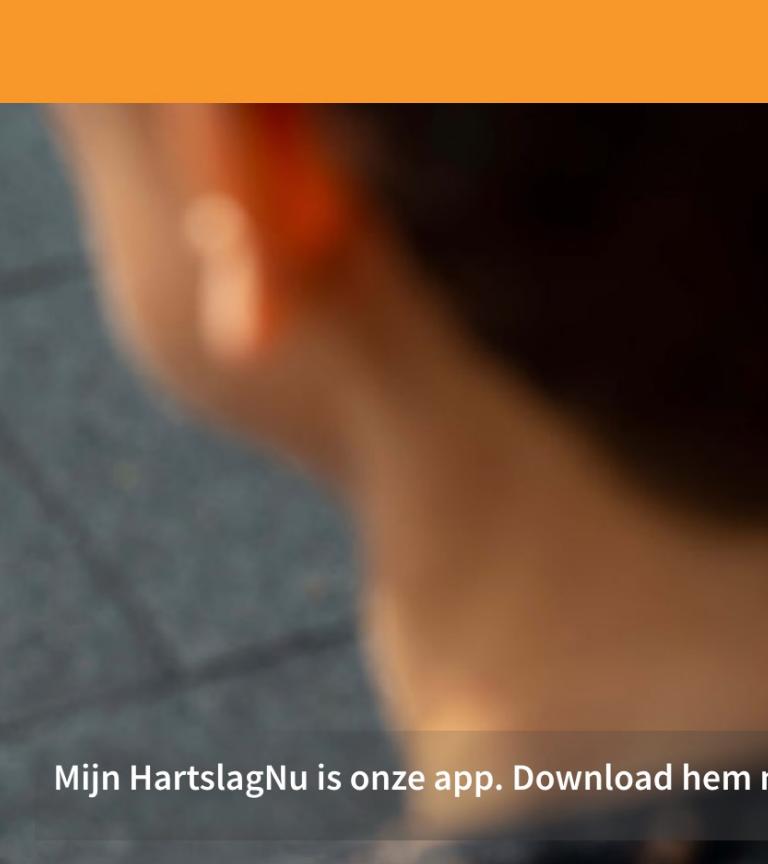
<>

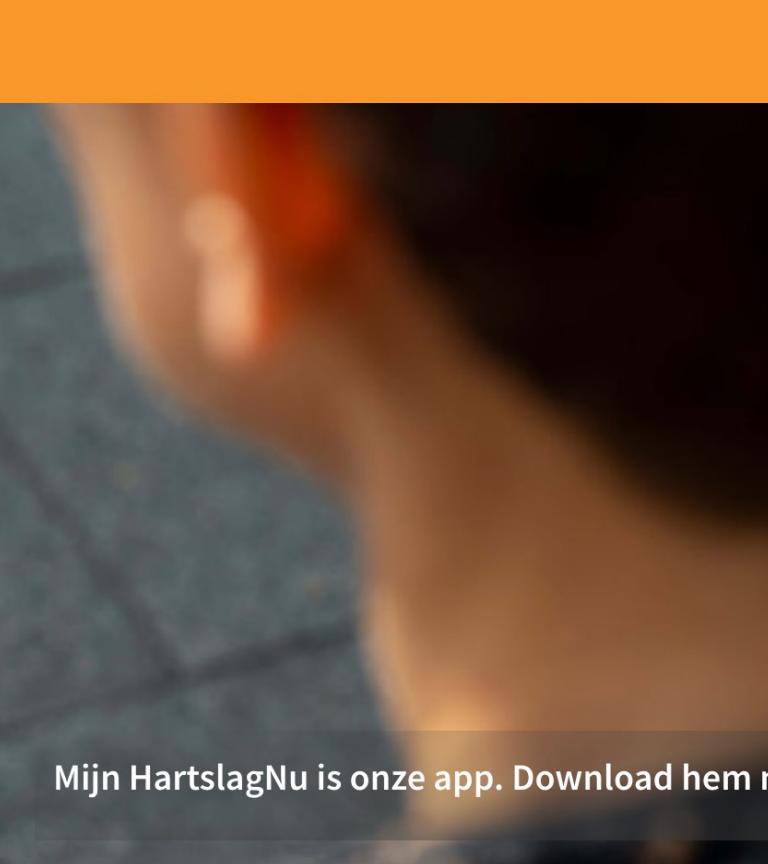
<>

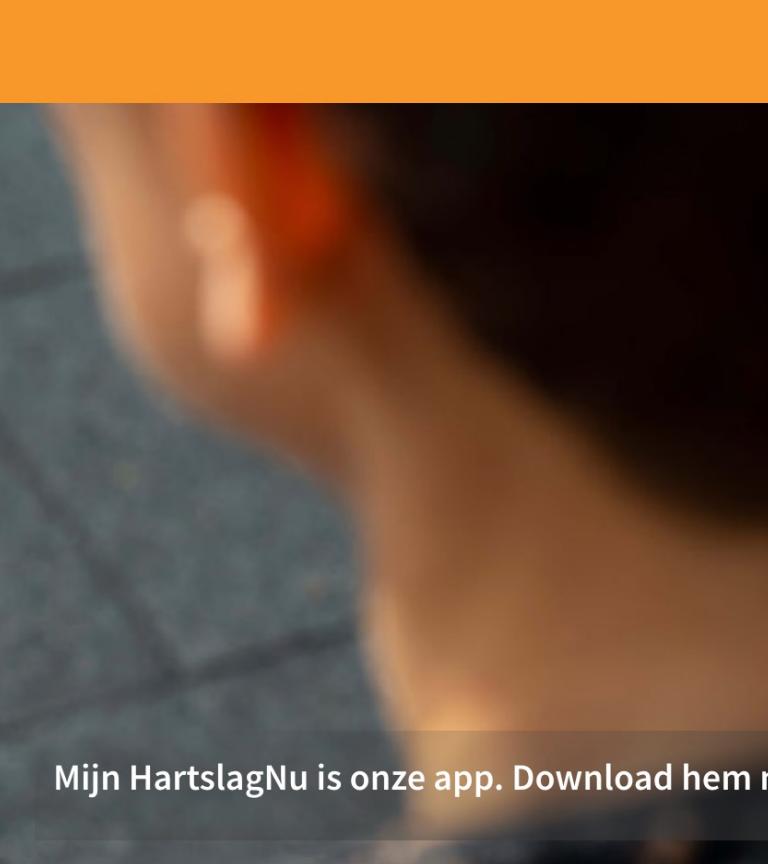
<>

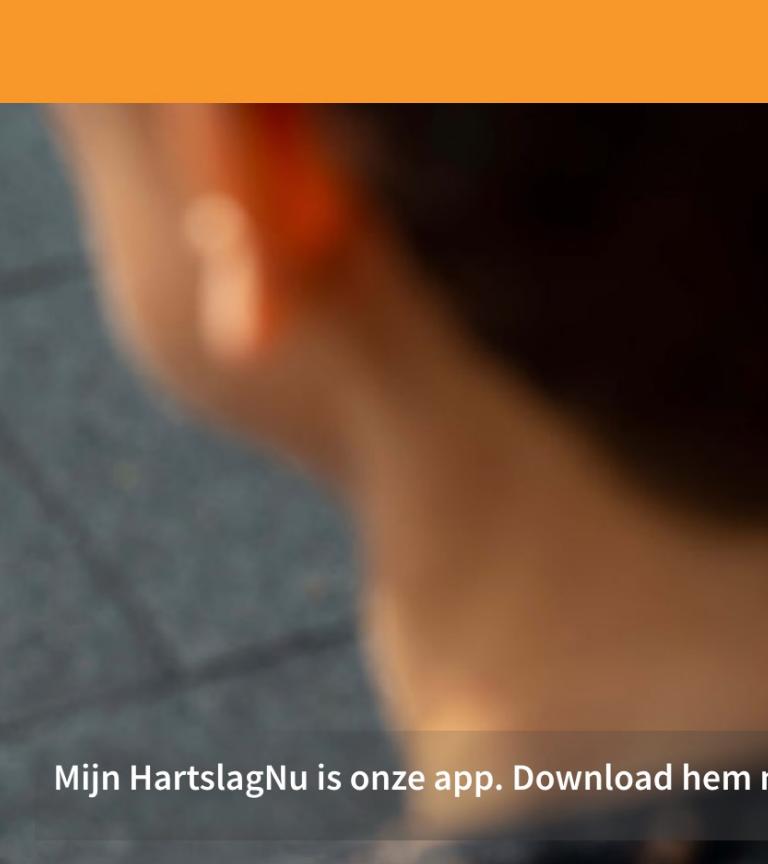
<>

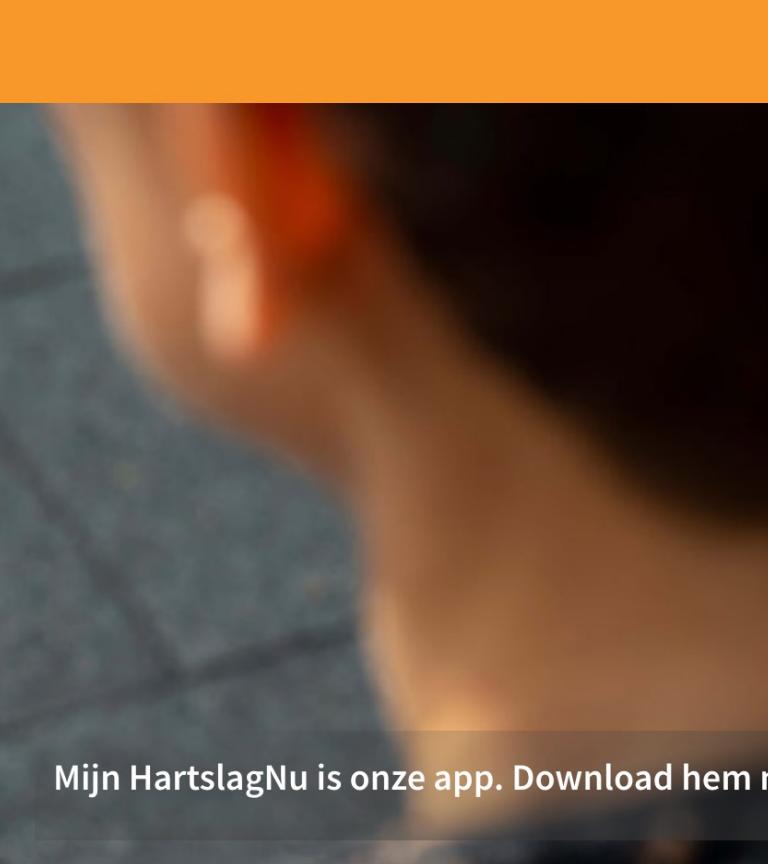
<>

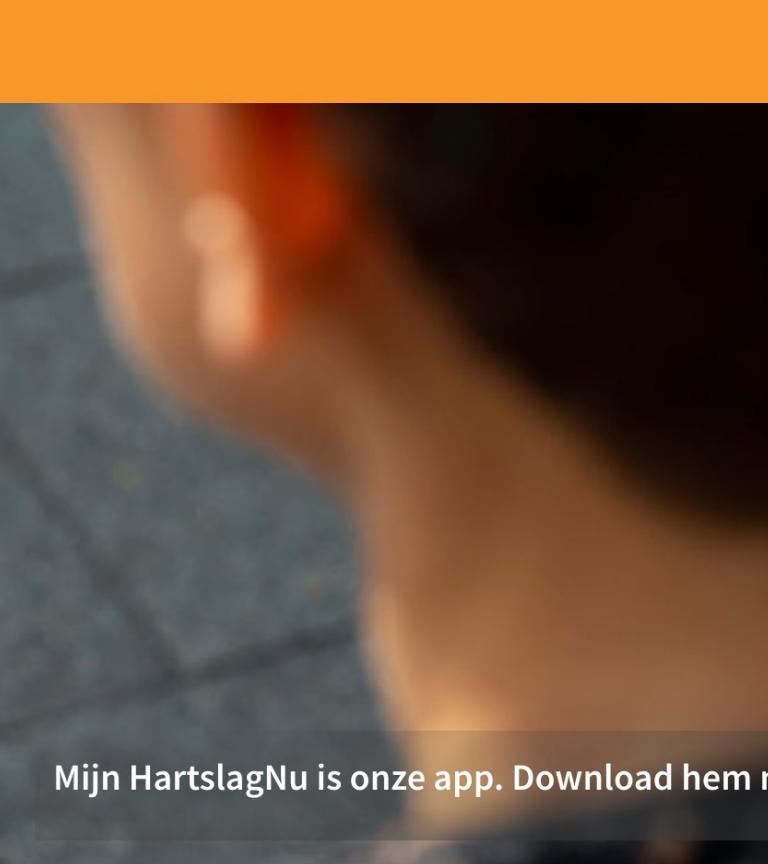
<>

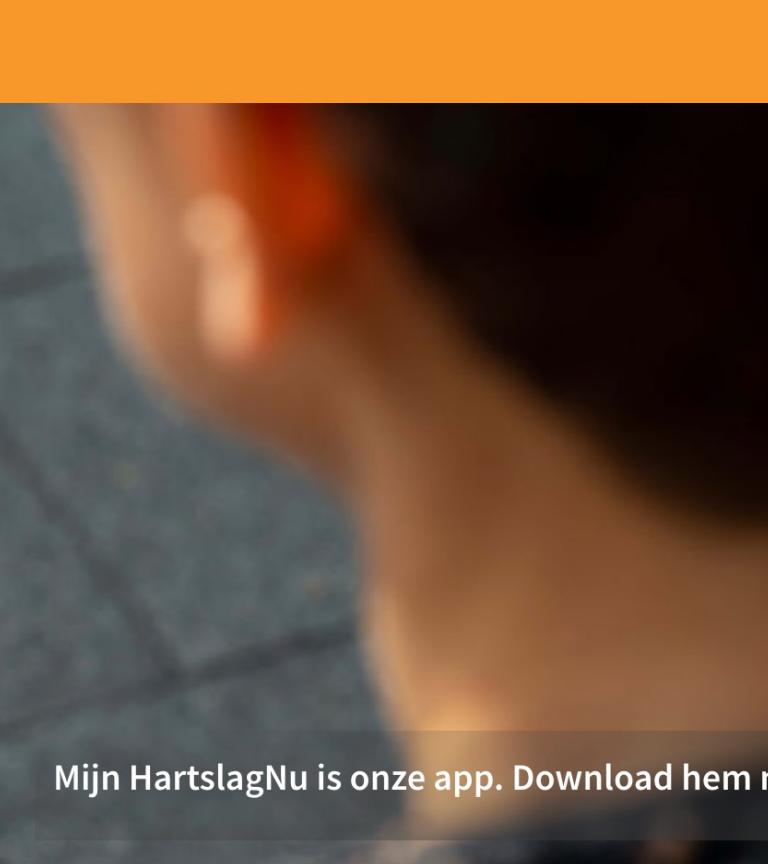
<>

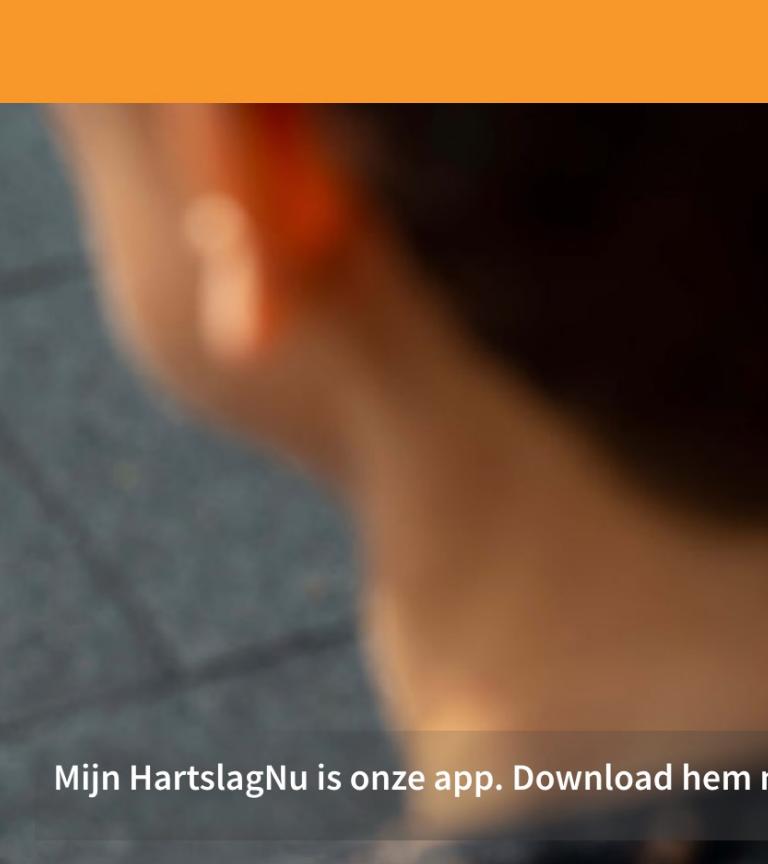
<>

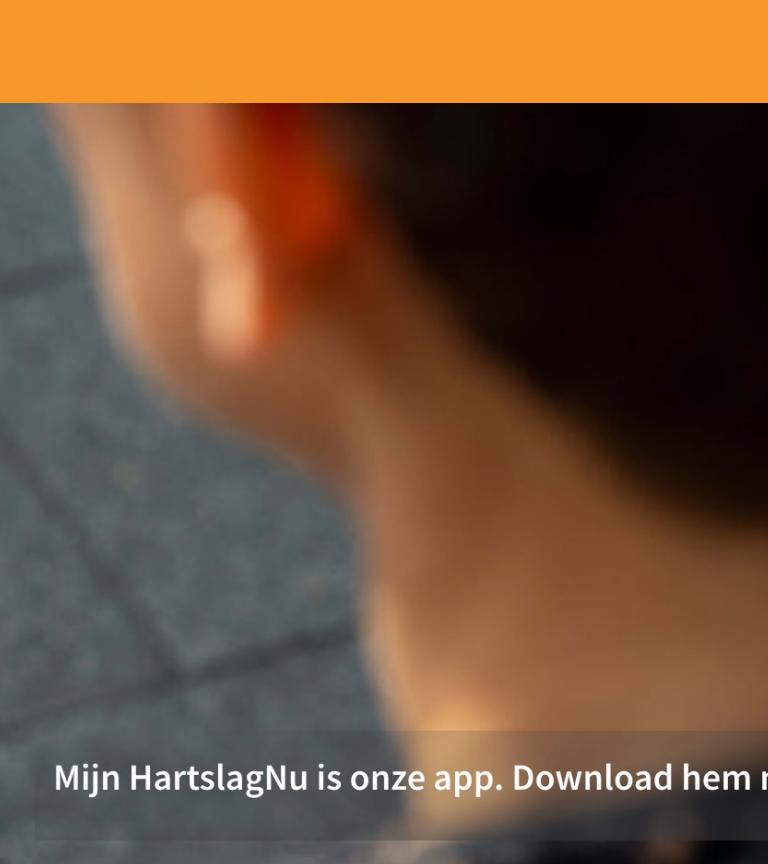
<>

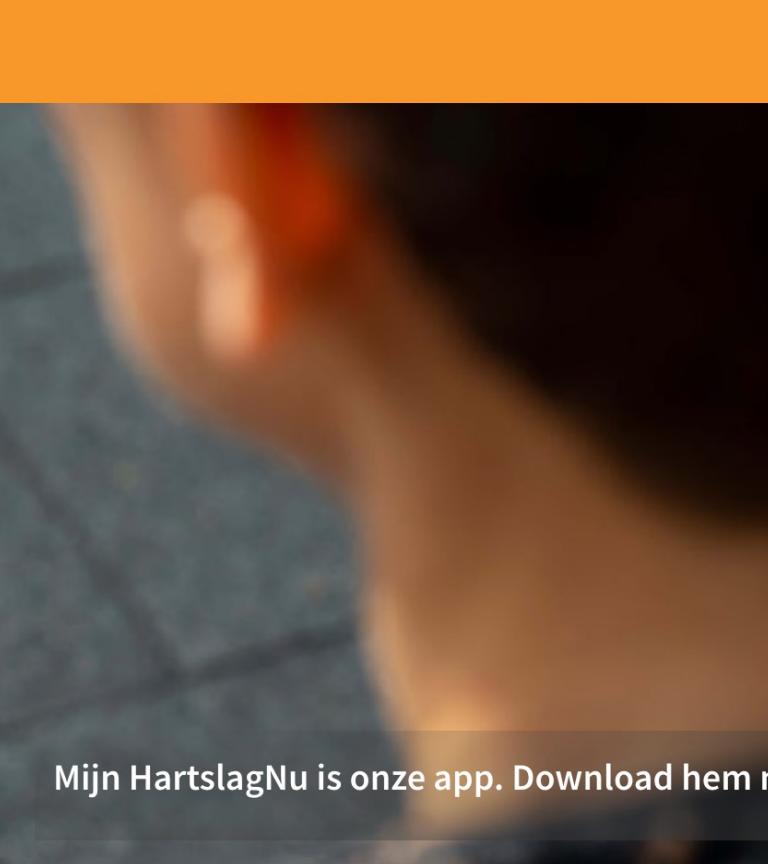
<>

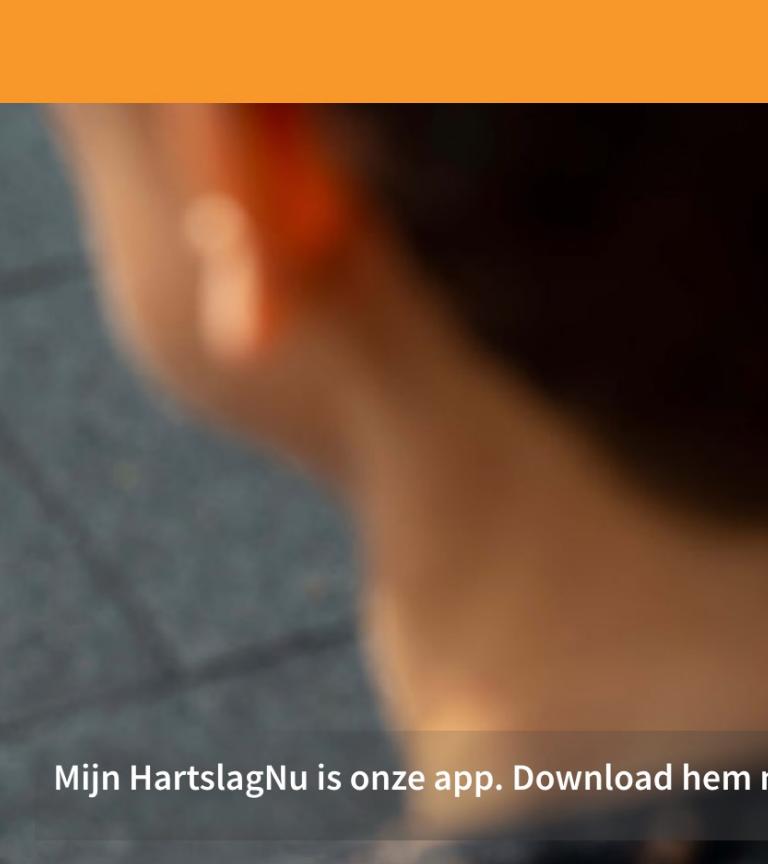
<>

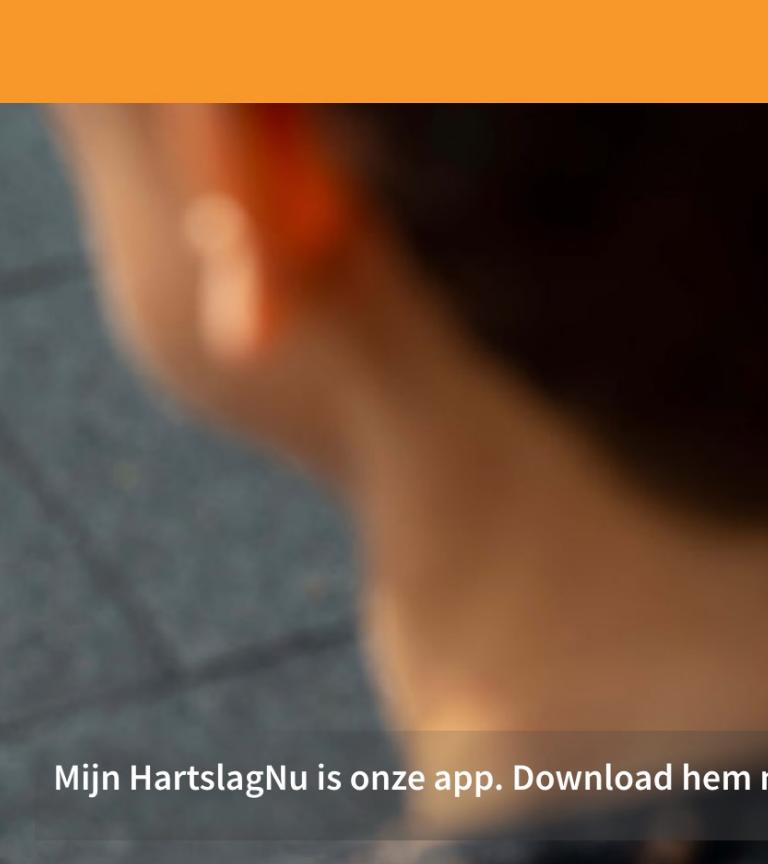
<>

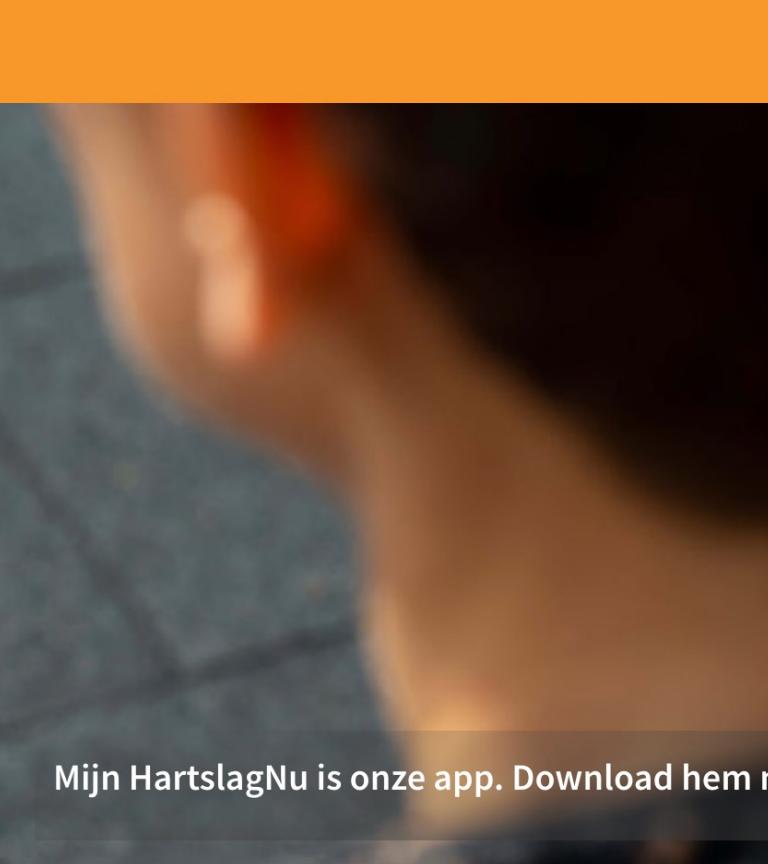
<>

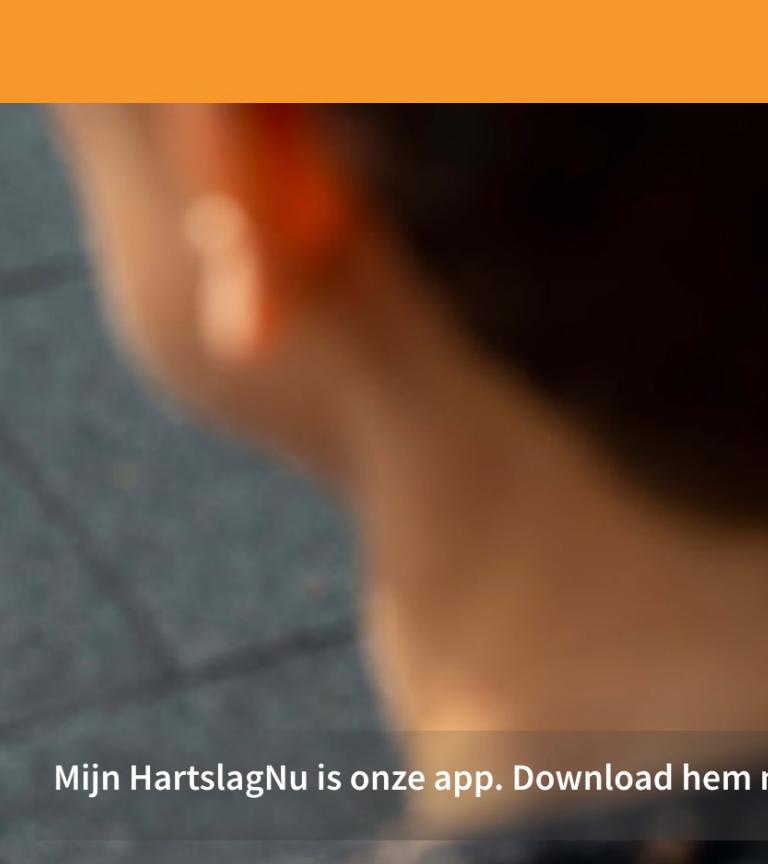
<>

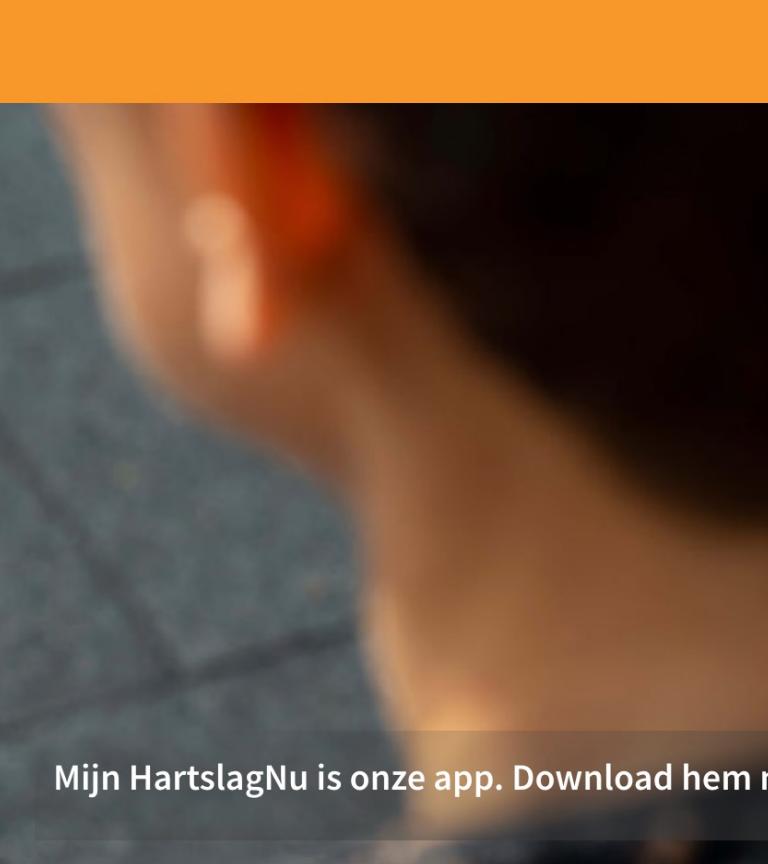
<>

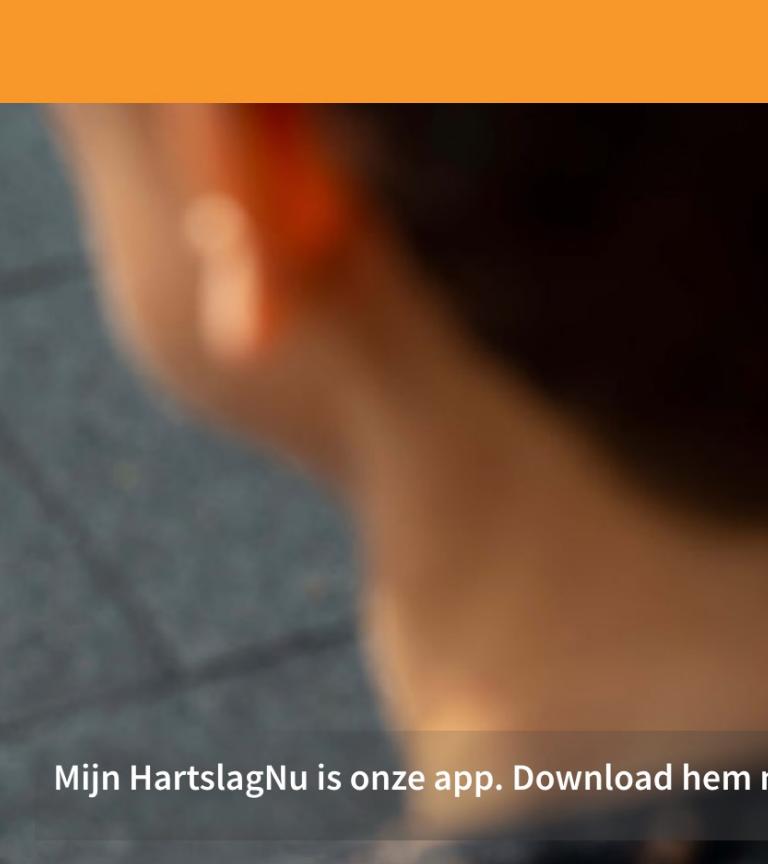
<>

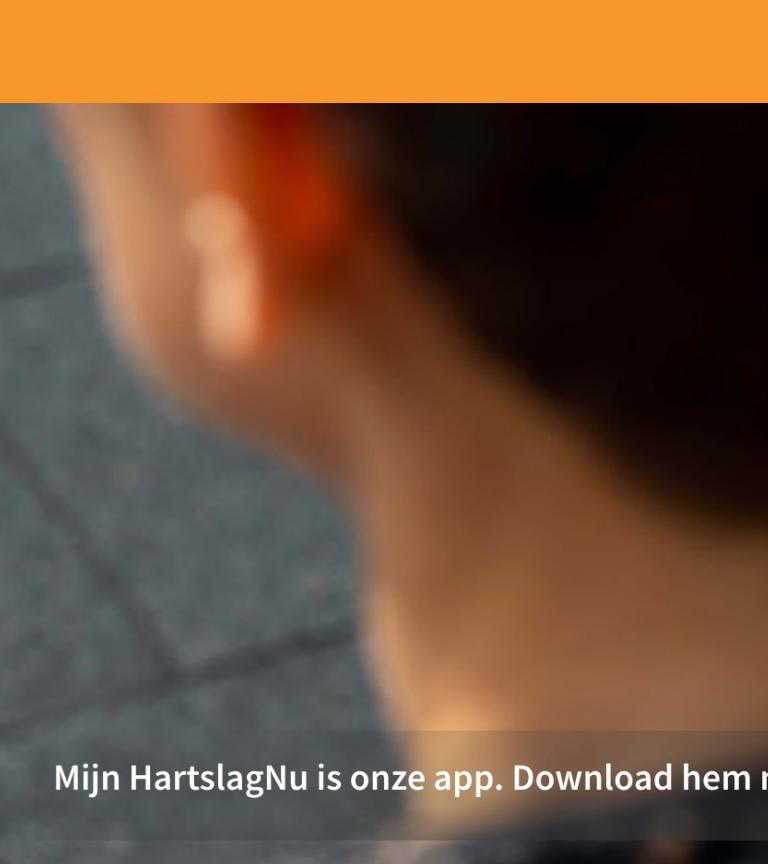
<>

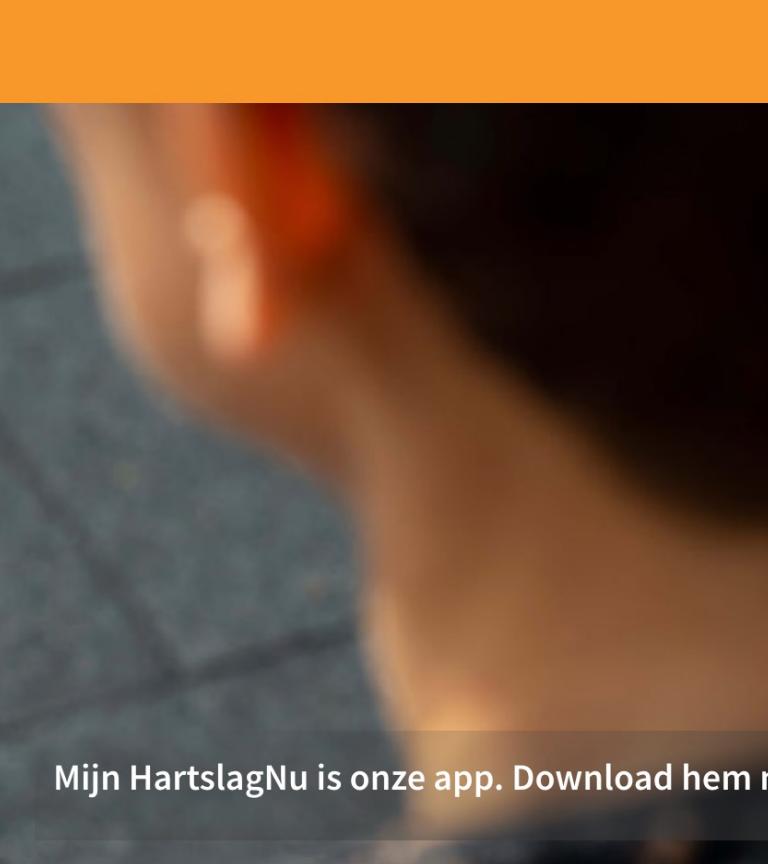
<>

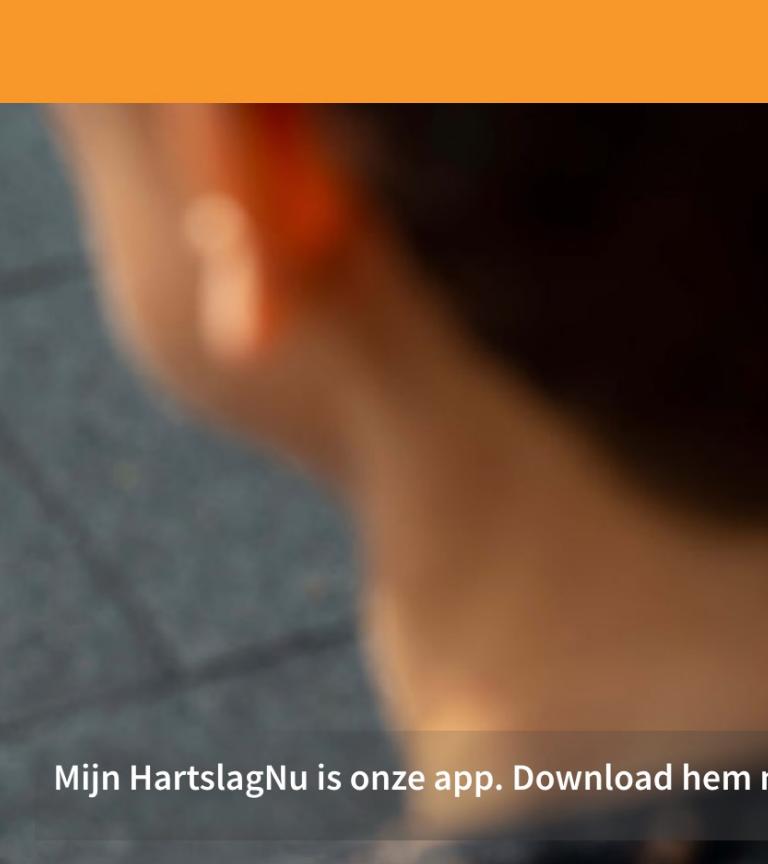
<>

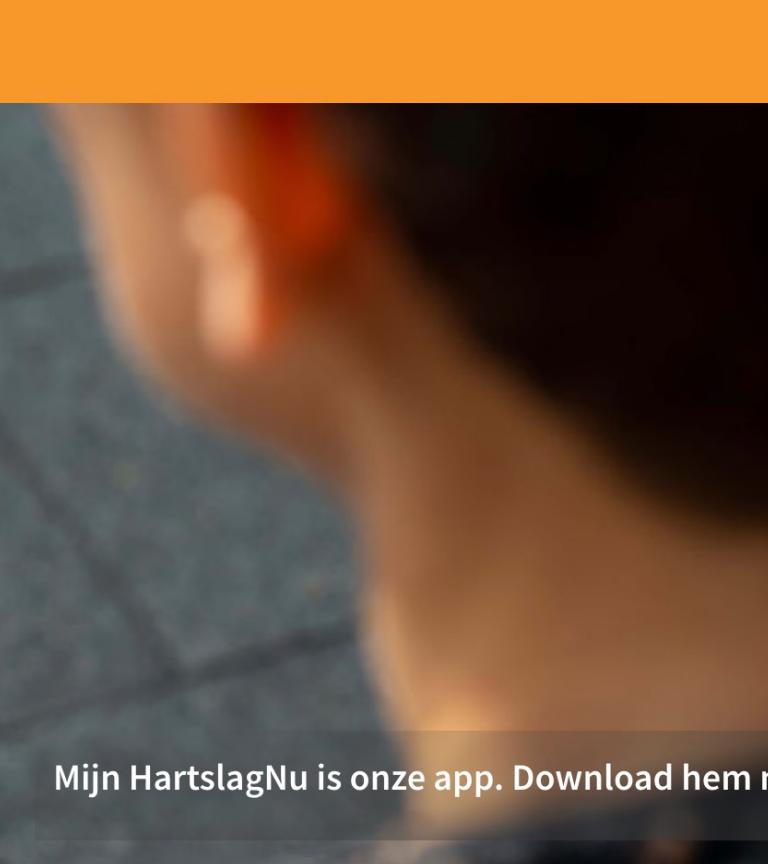
<>

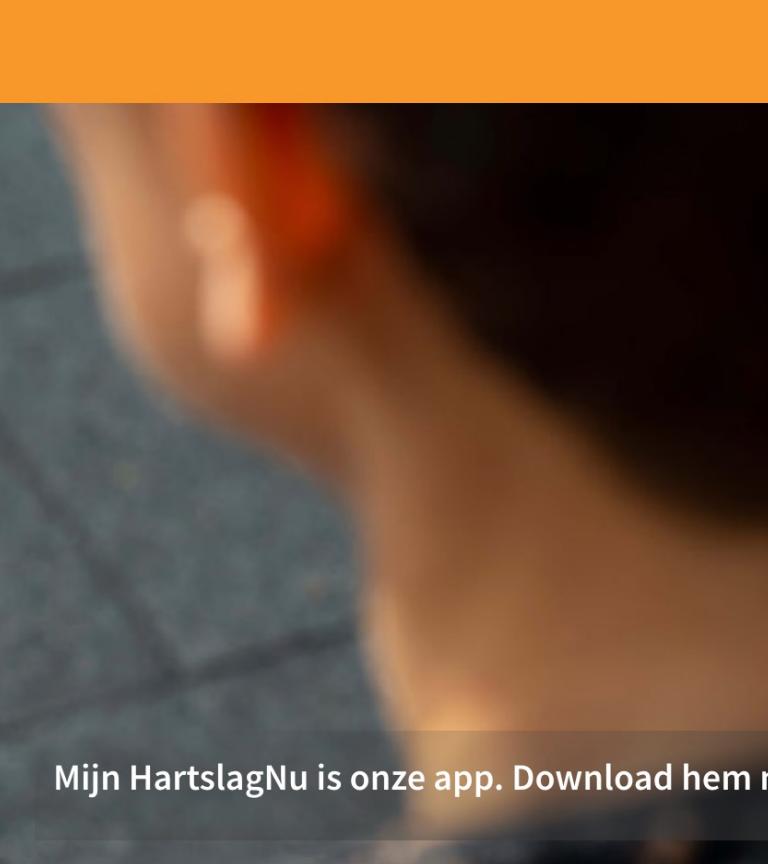
<>

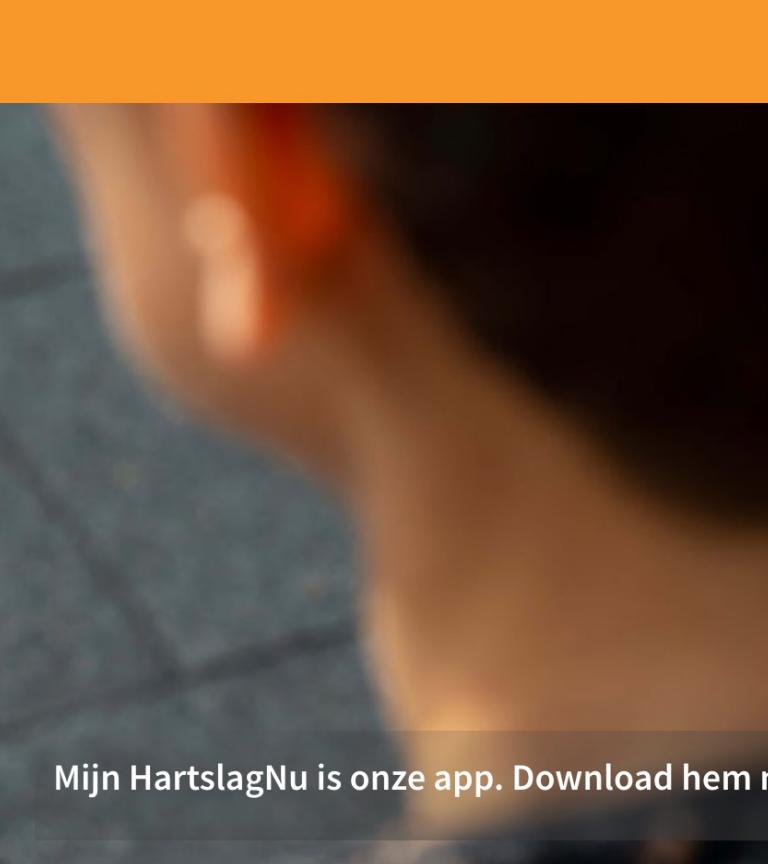
<>

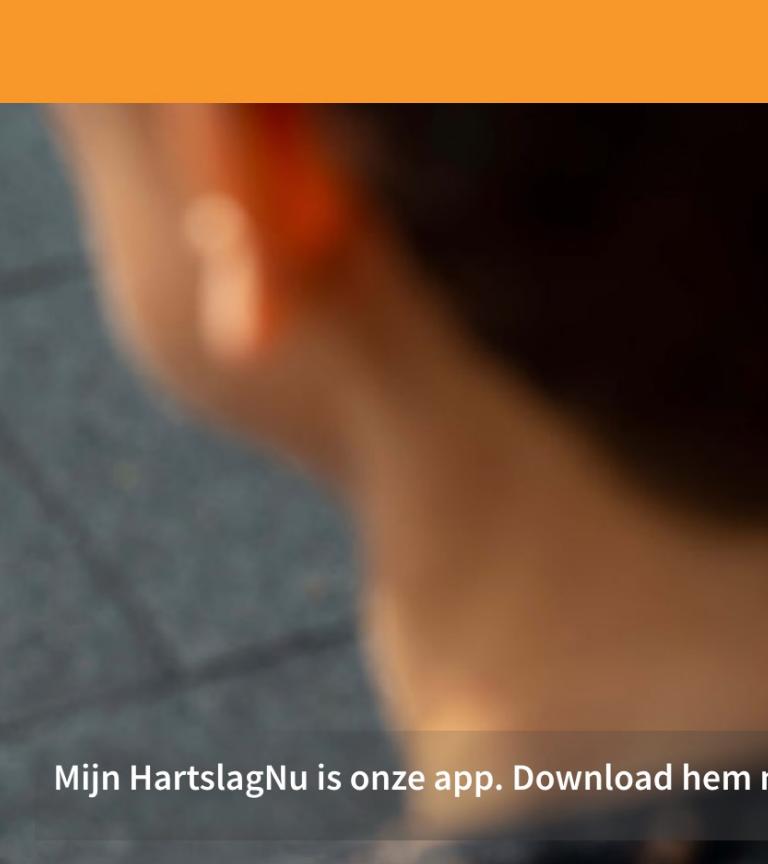
<>

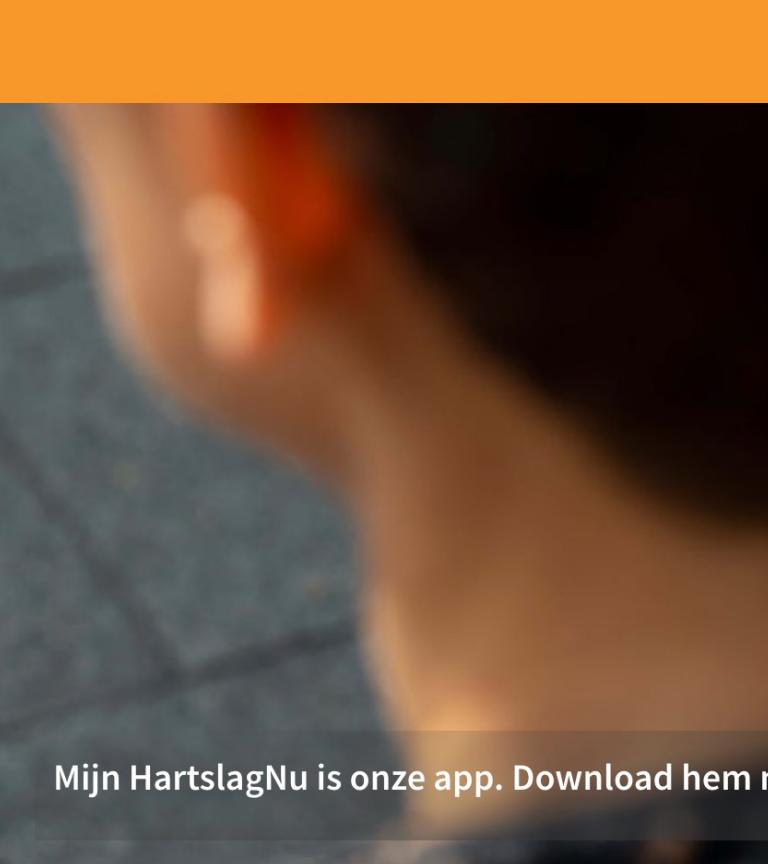
<>

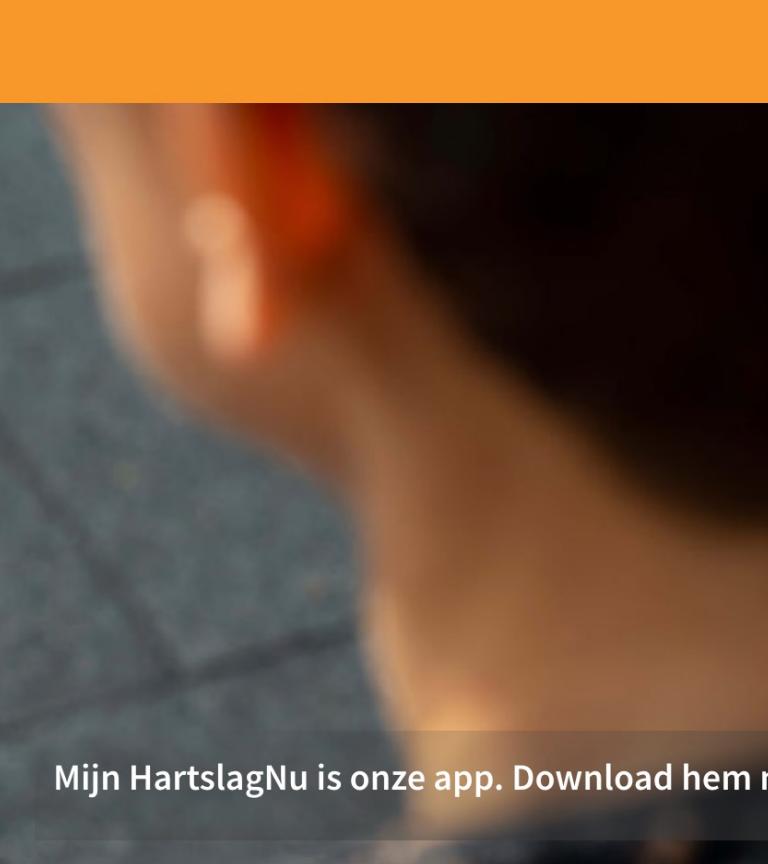
<>

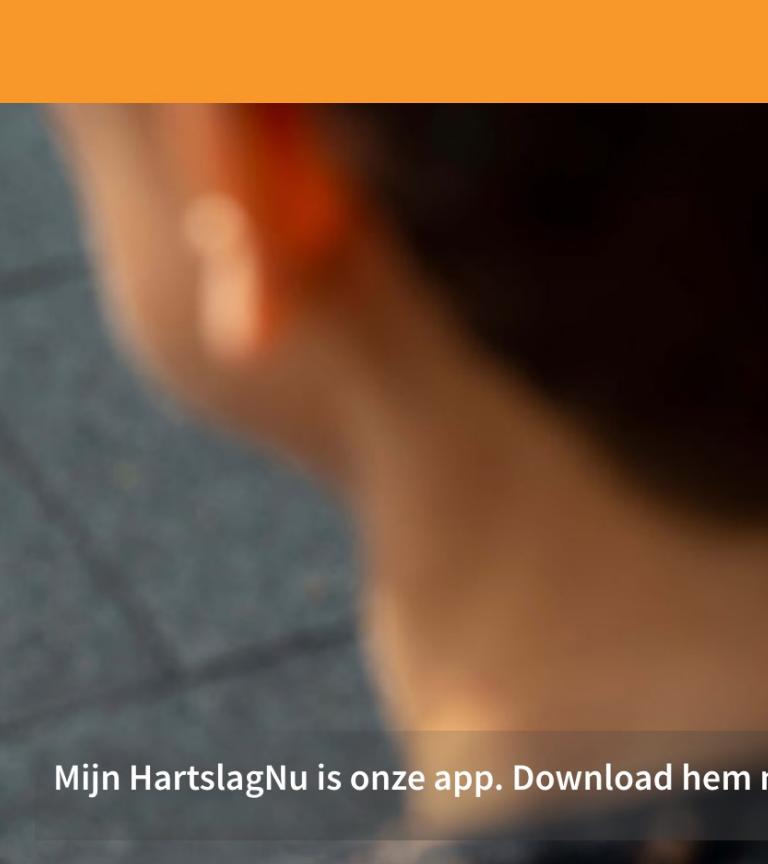
<>

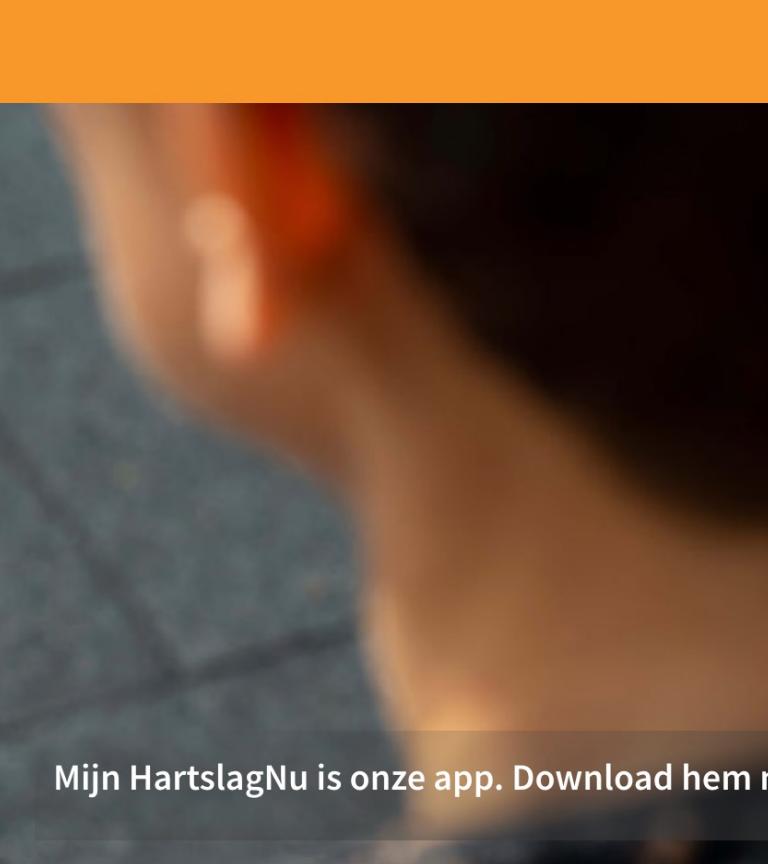
<>

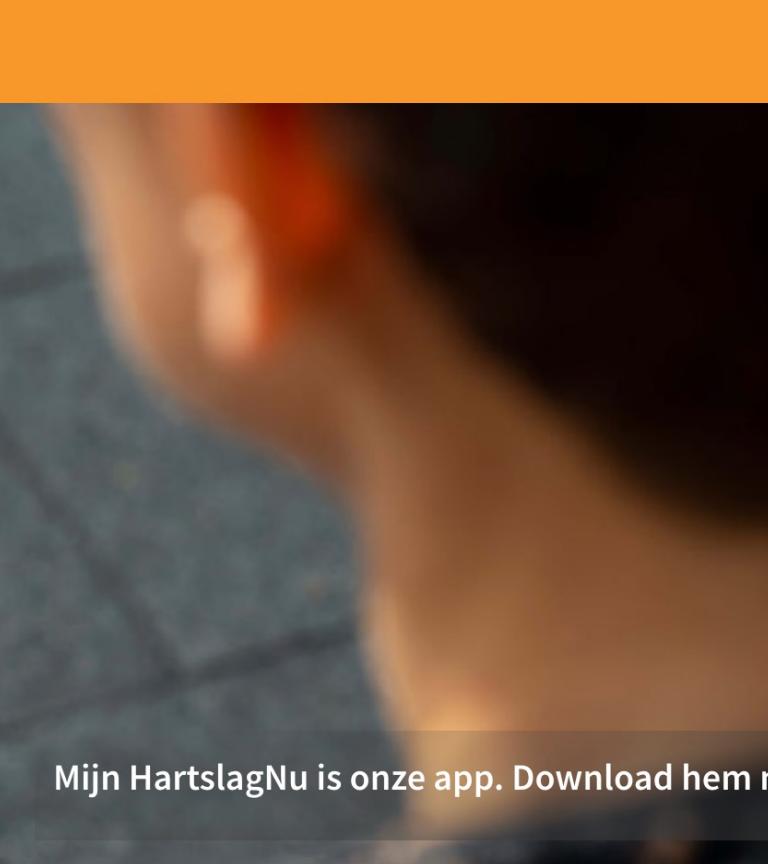
<>

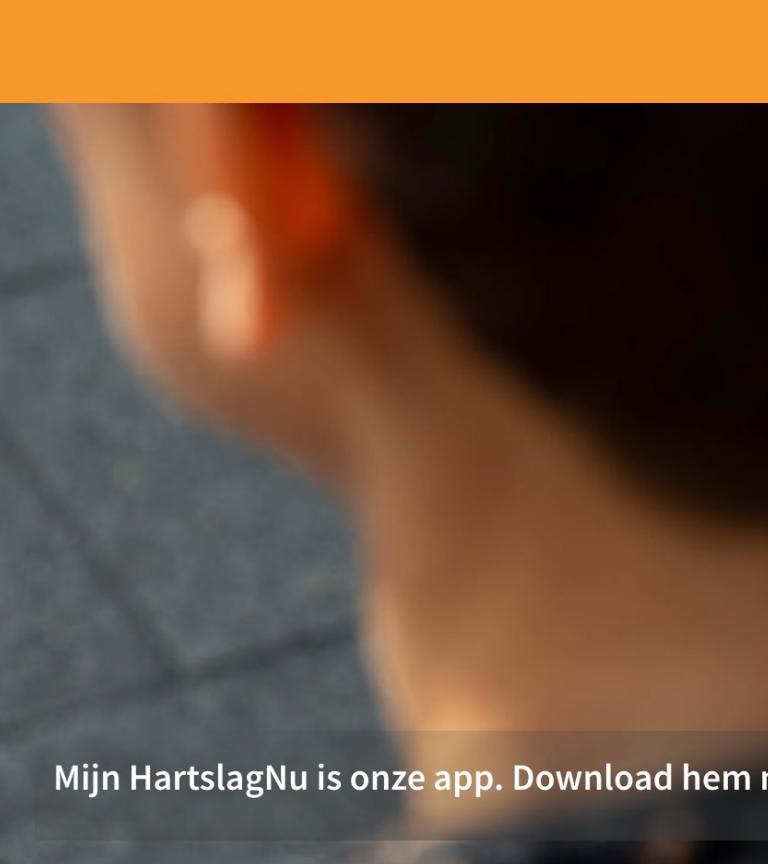
<>

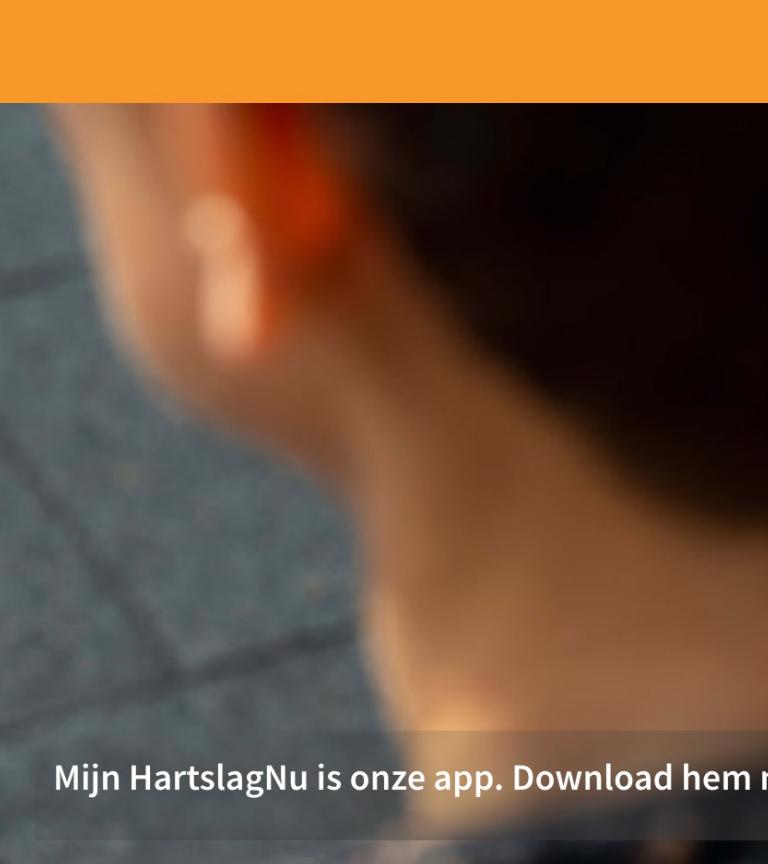
<>

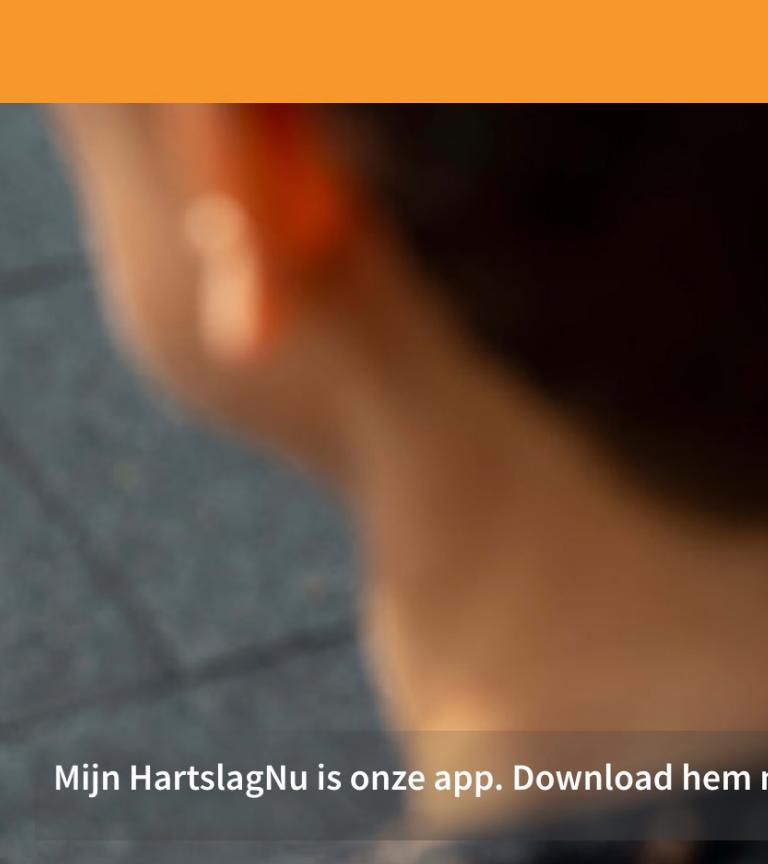
<>

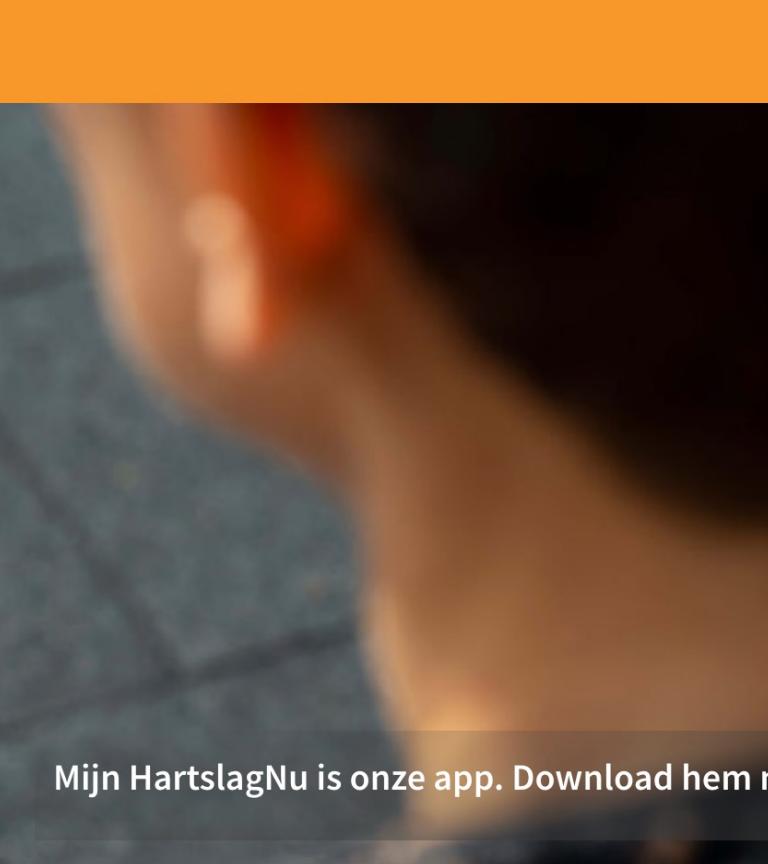
<>

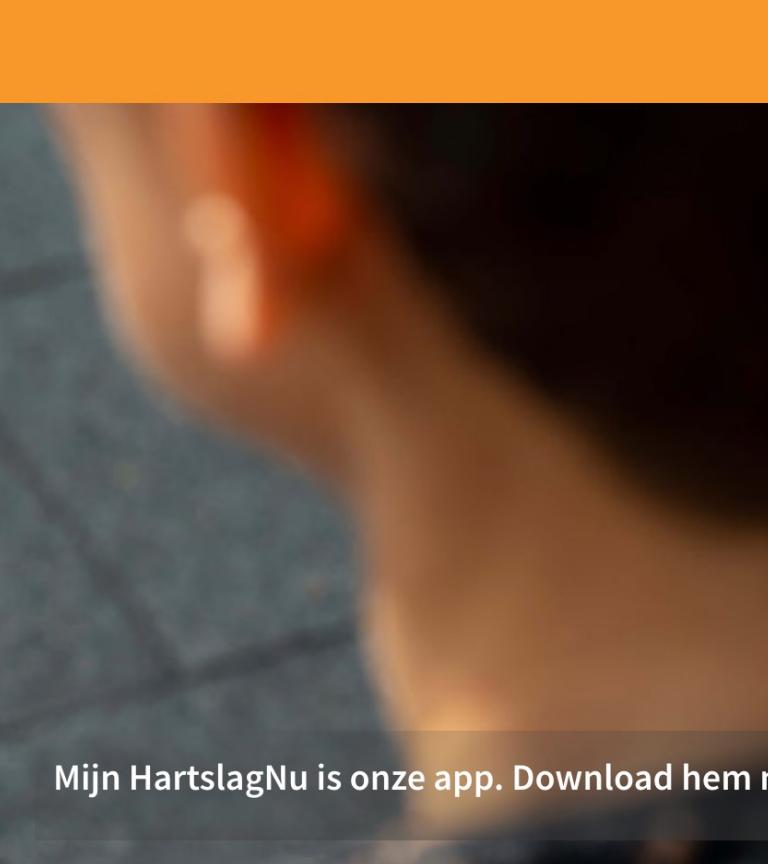
<>

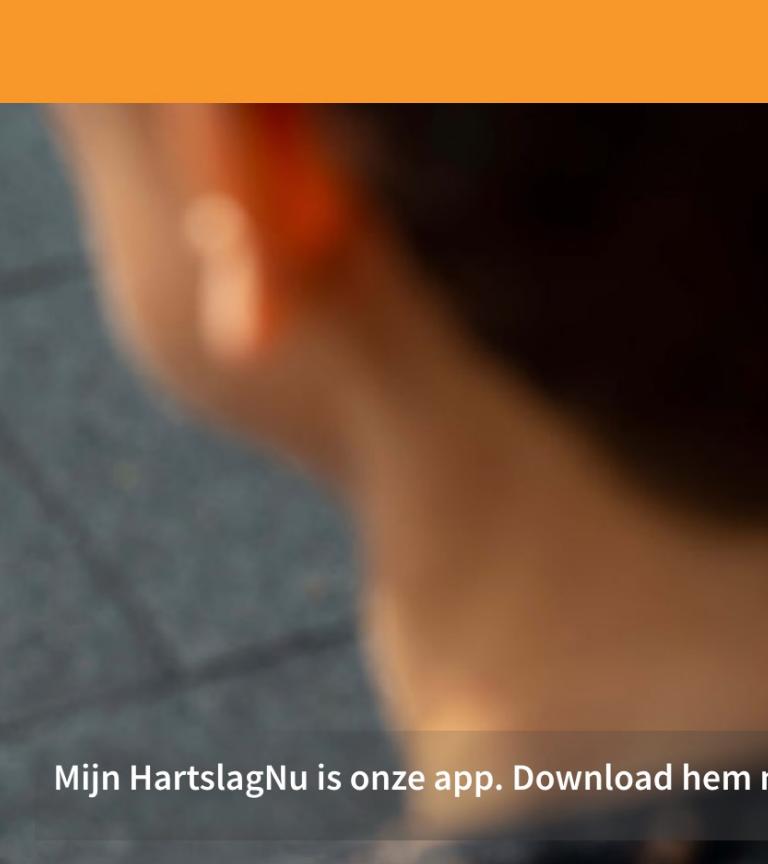
<>

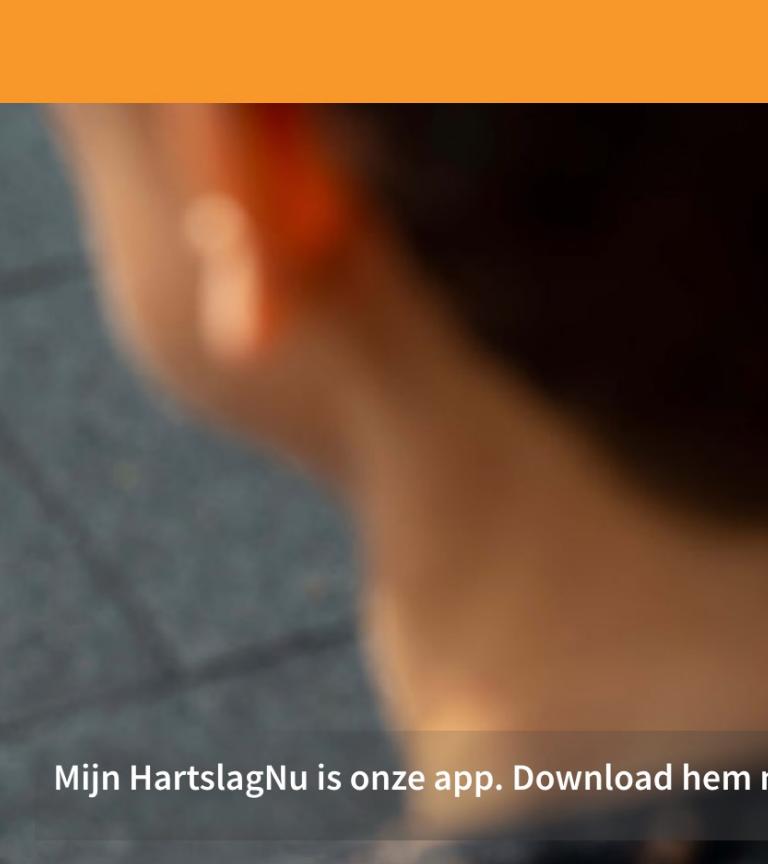
<>

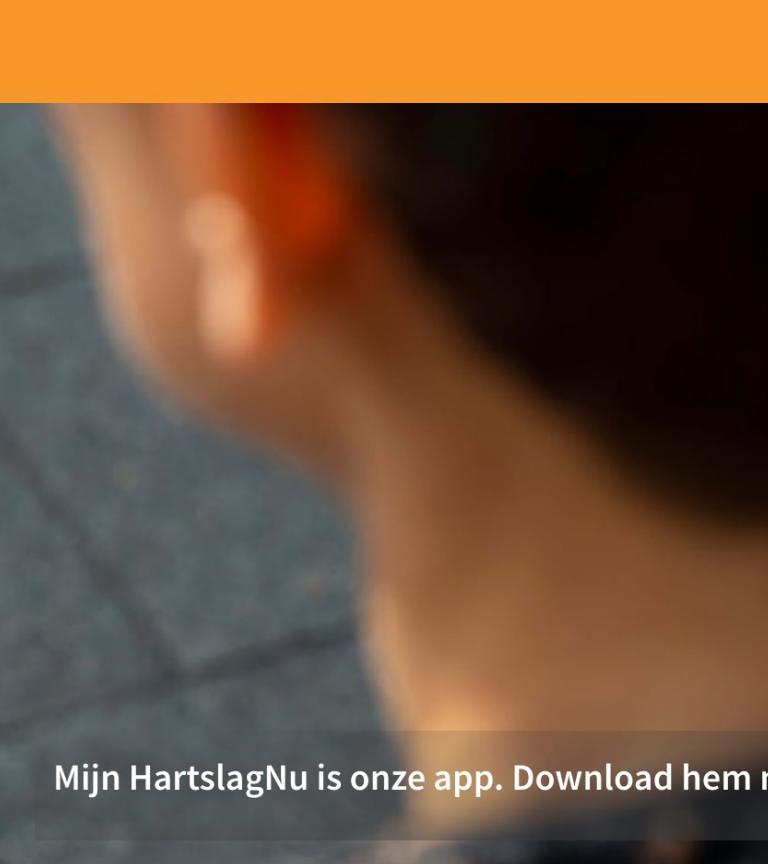
<>

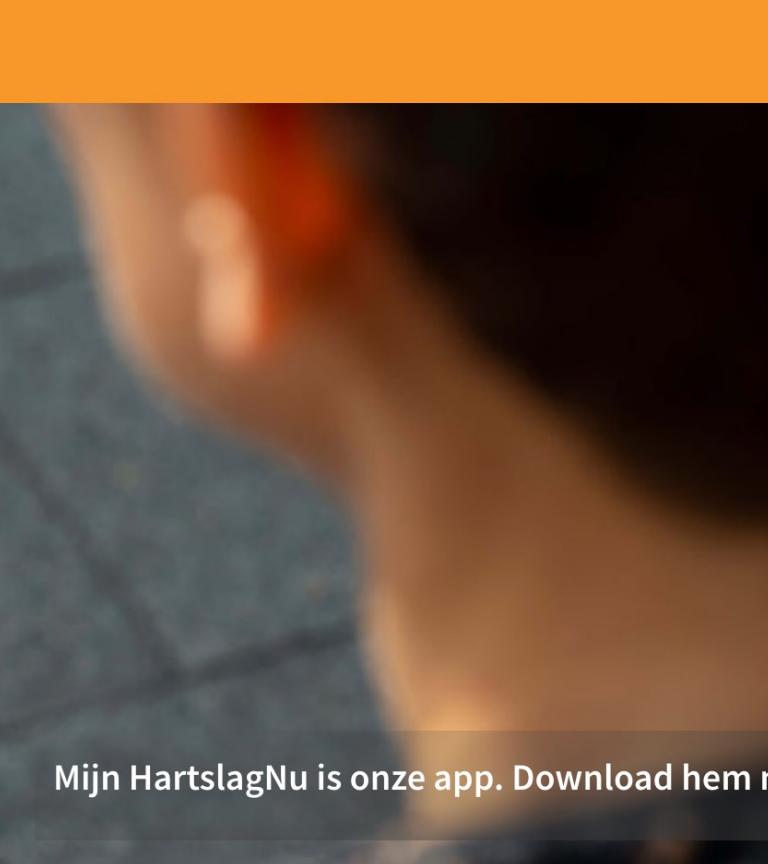
<>

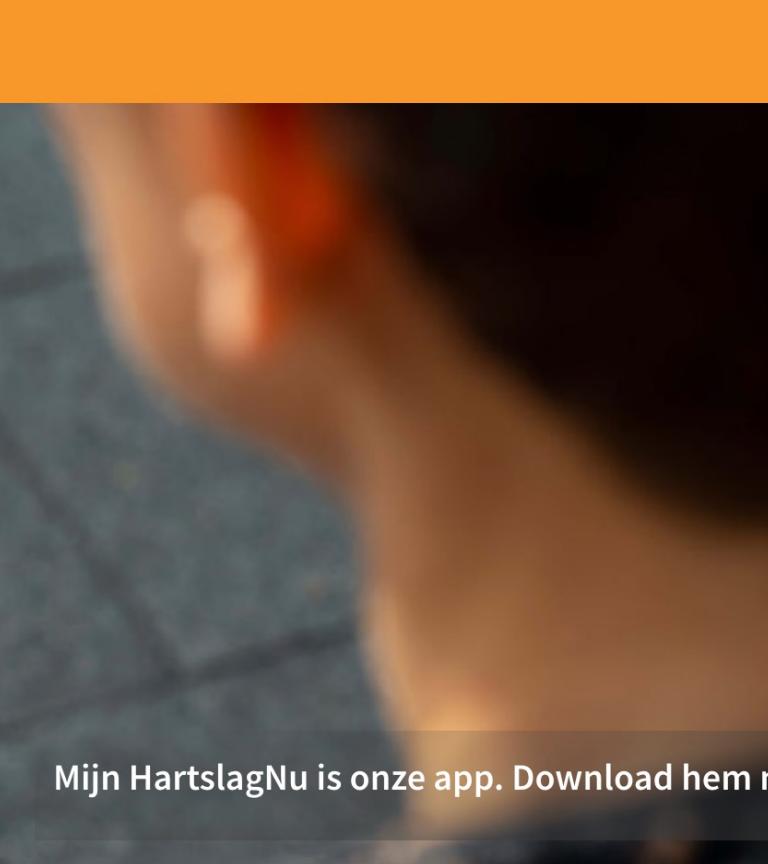
<>

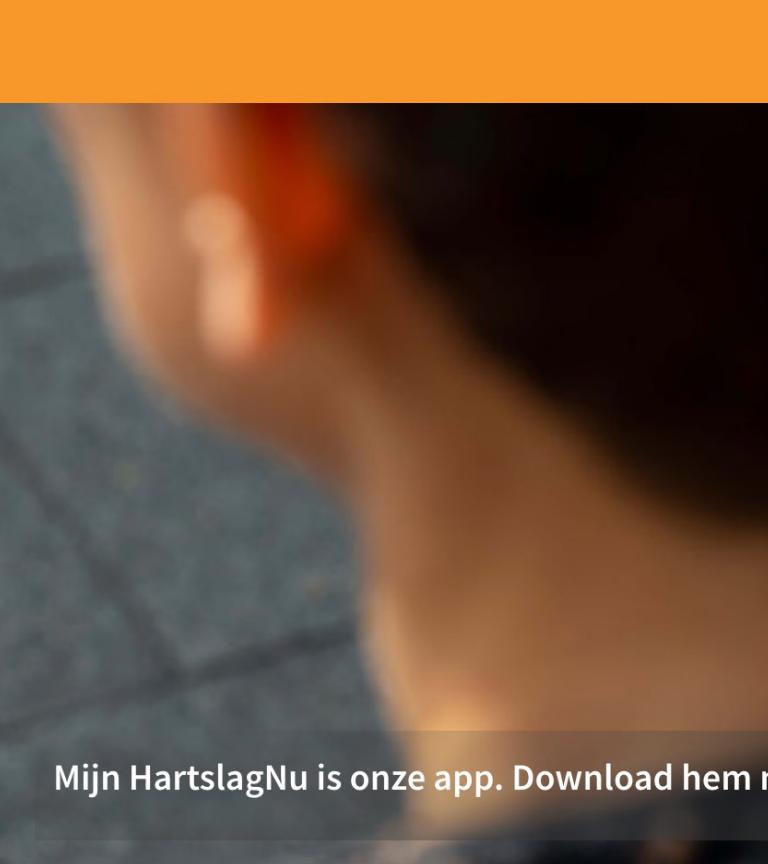
<>

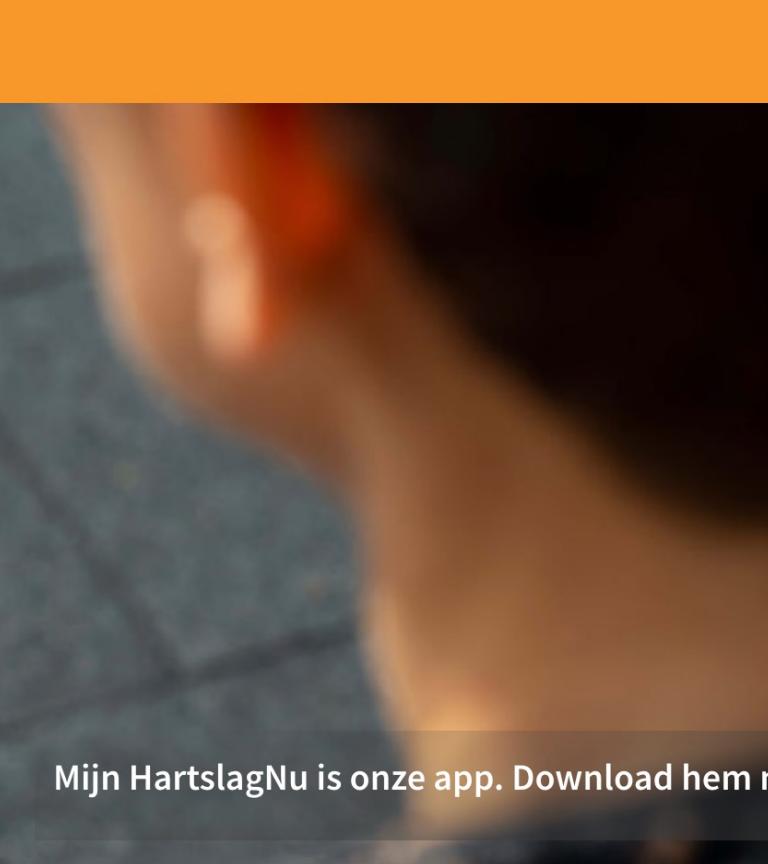
<>

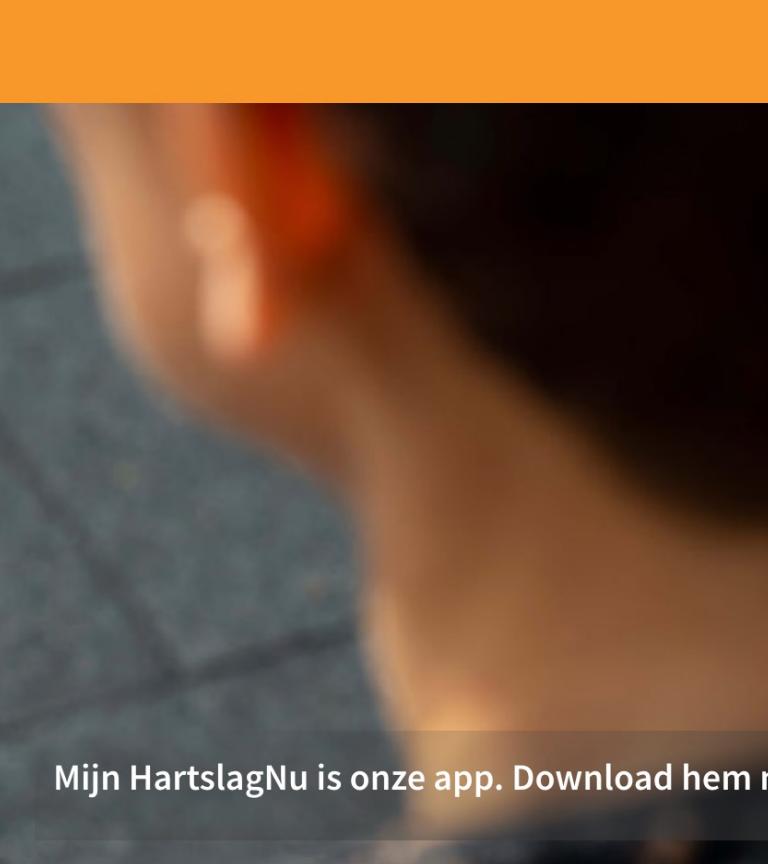
<>

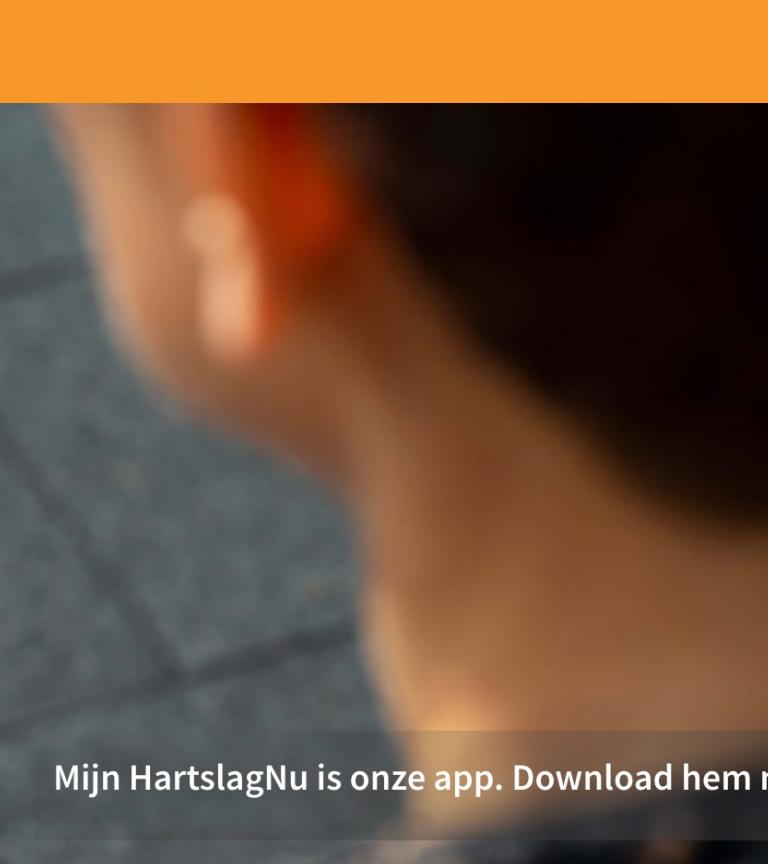
<>

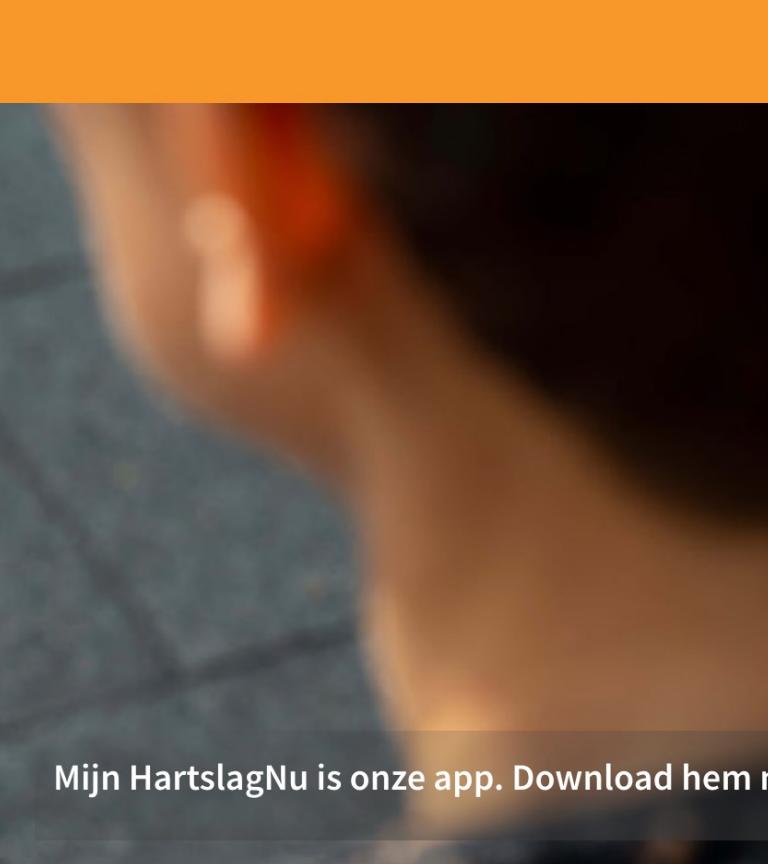
<>

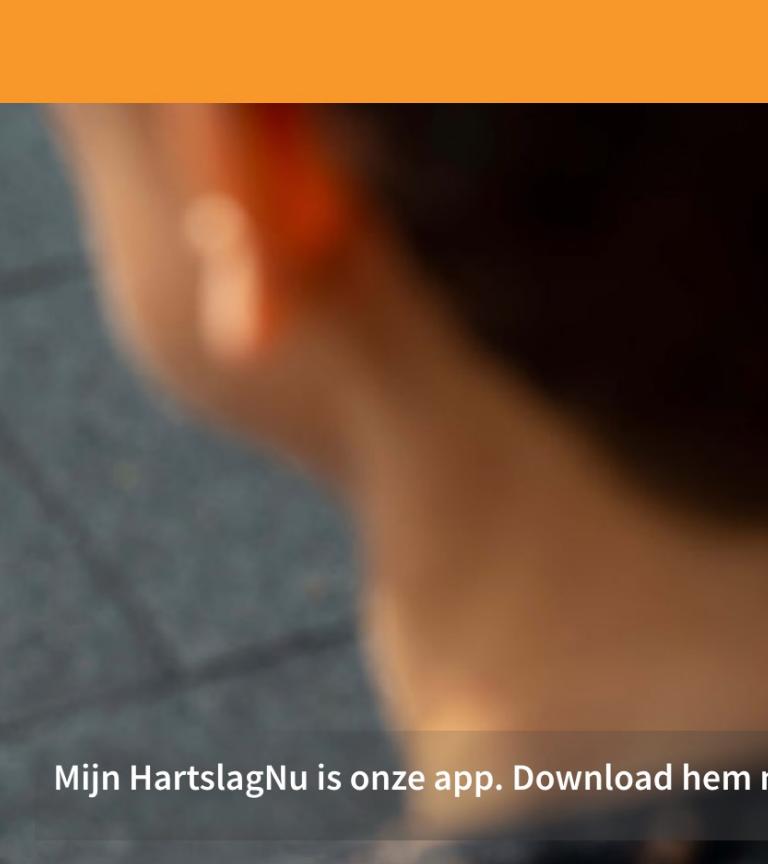
<>

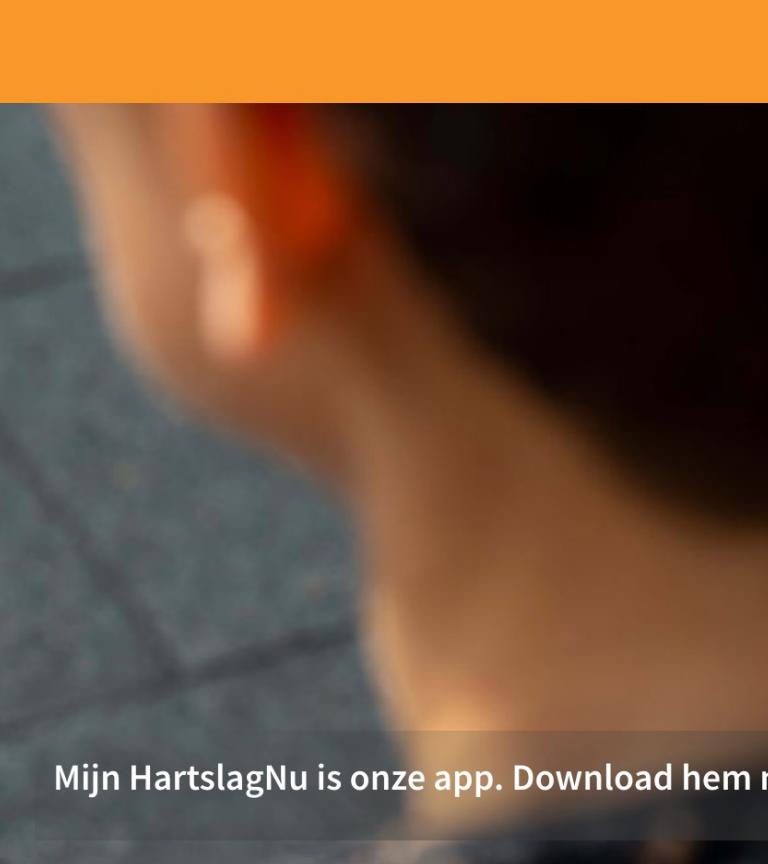
<>

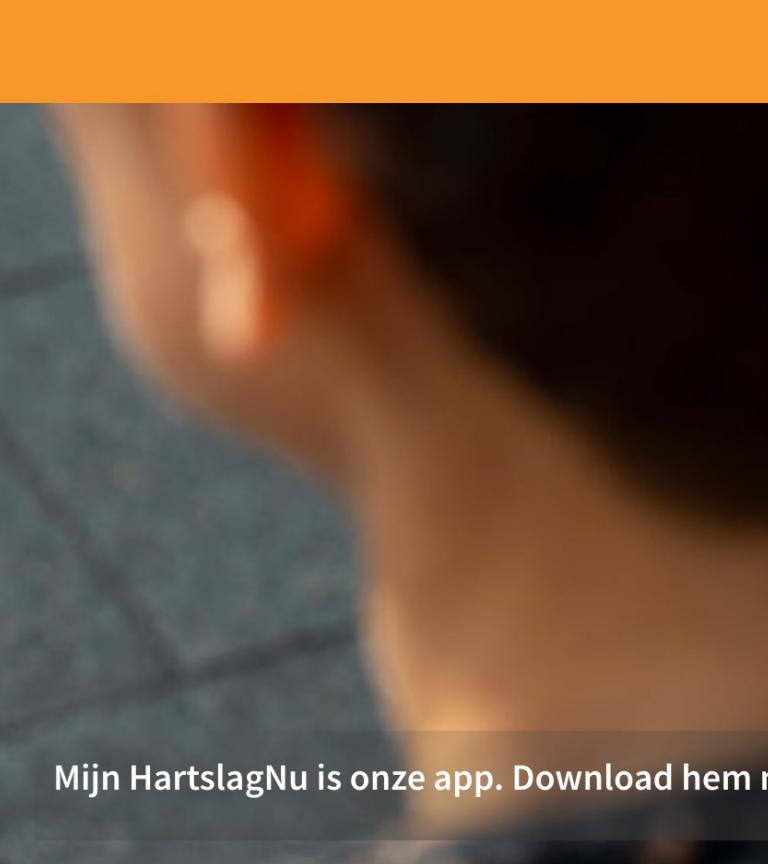
<>

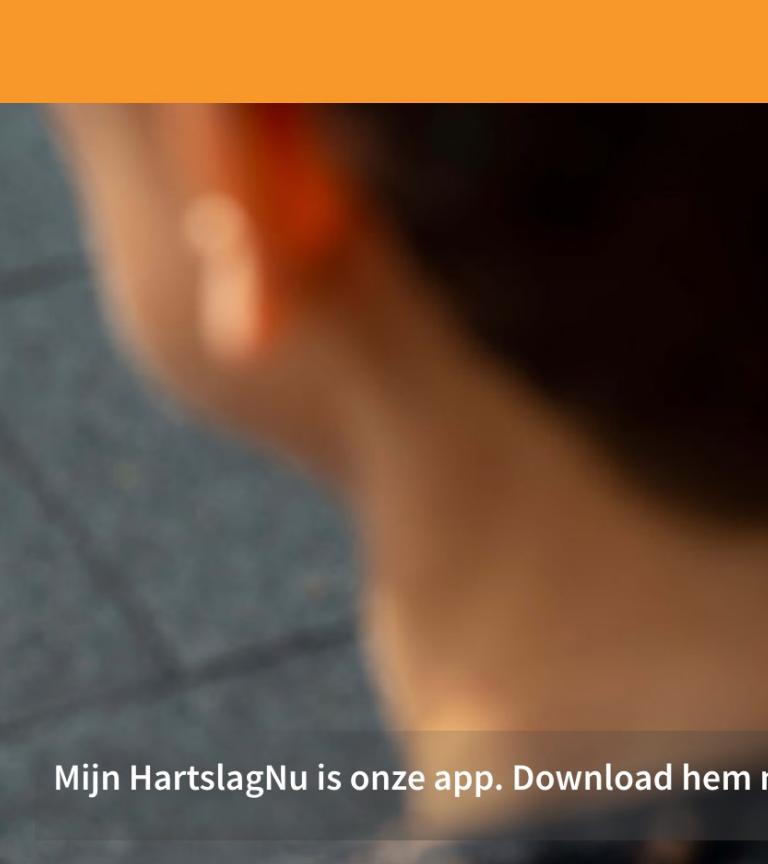
<>

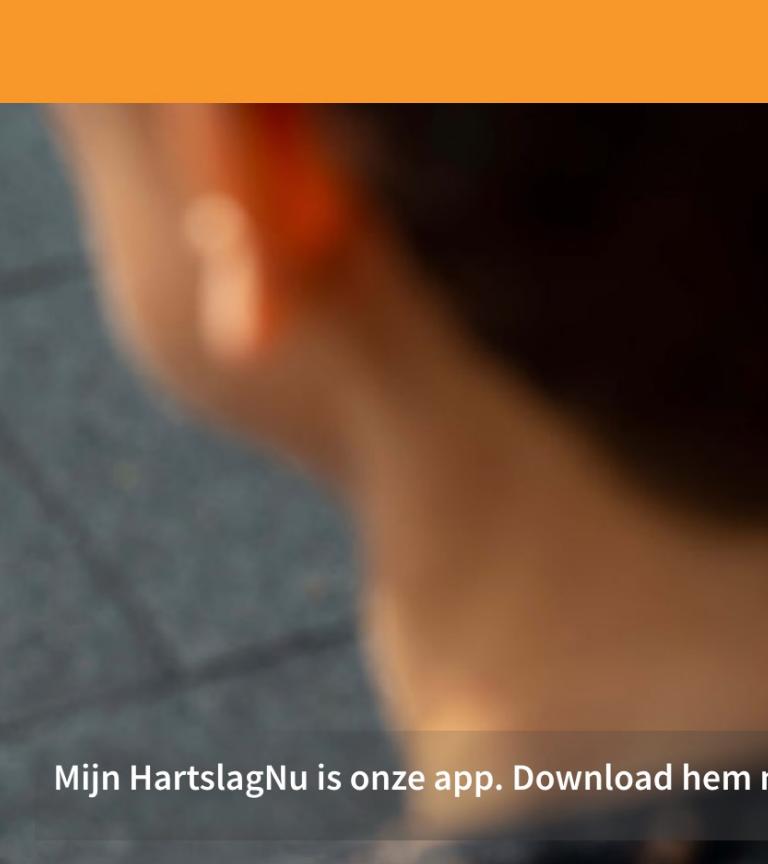
<>

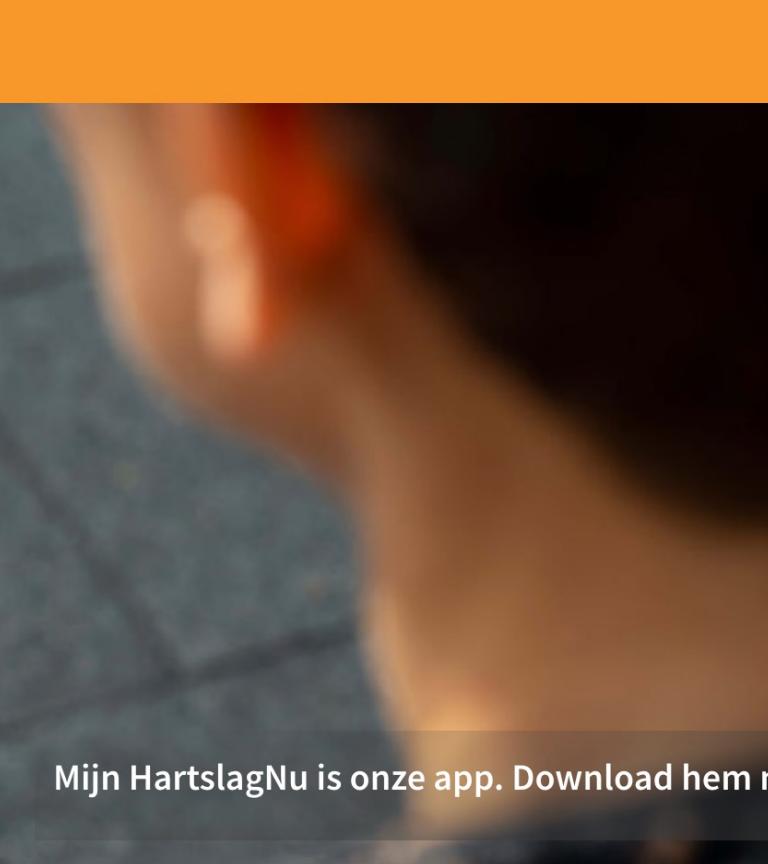
<>

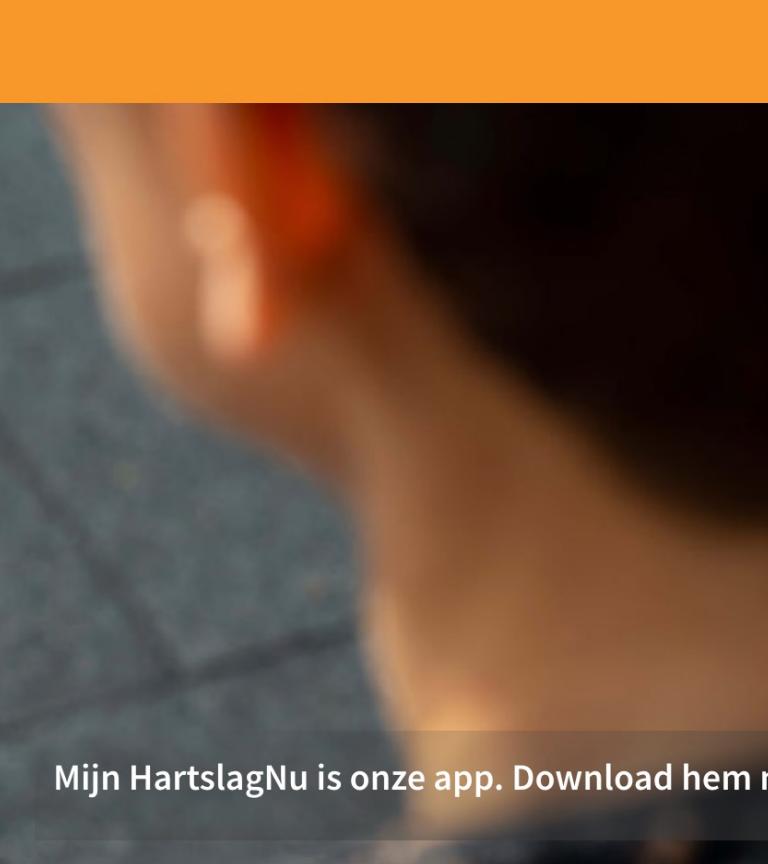
<>

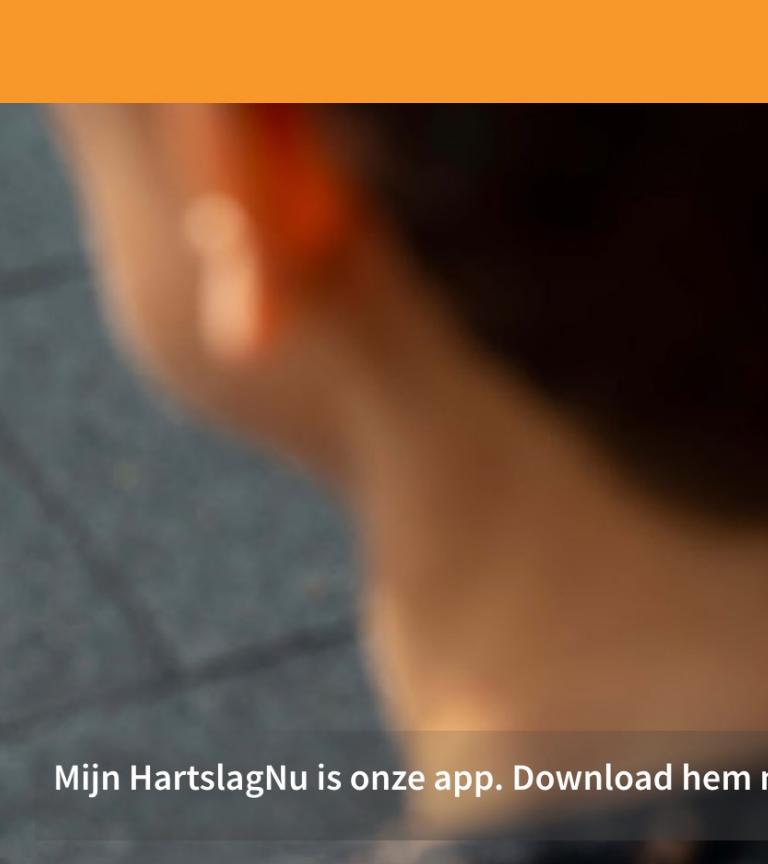
<>

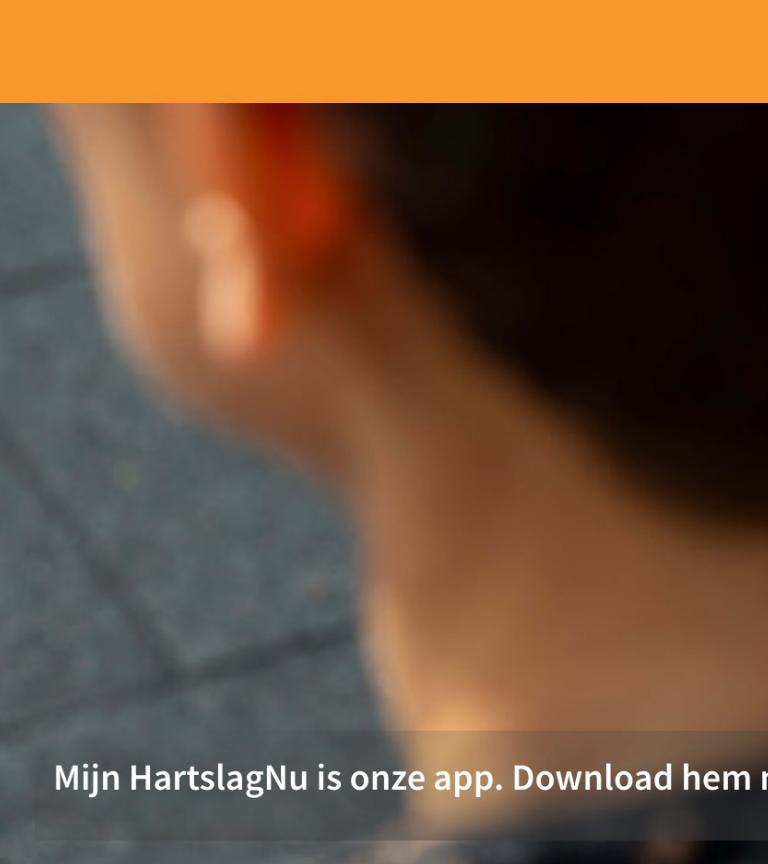
<>

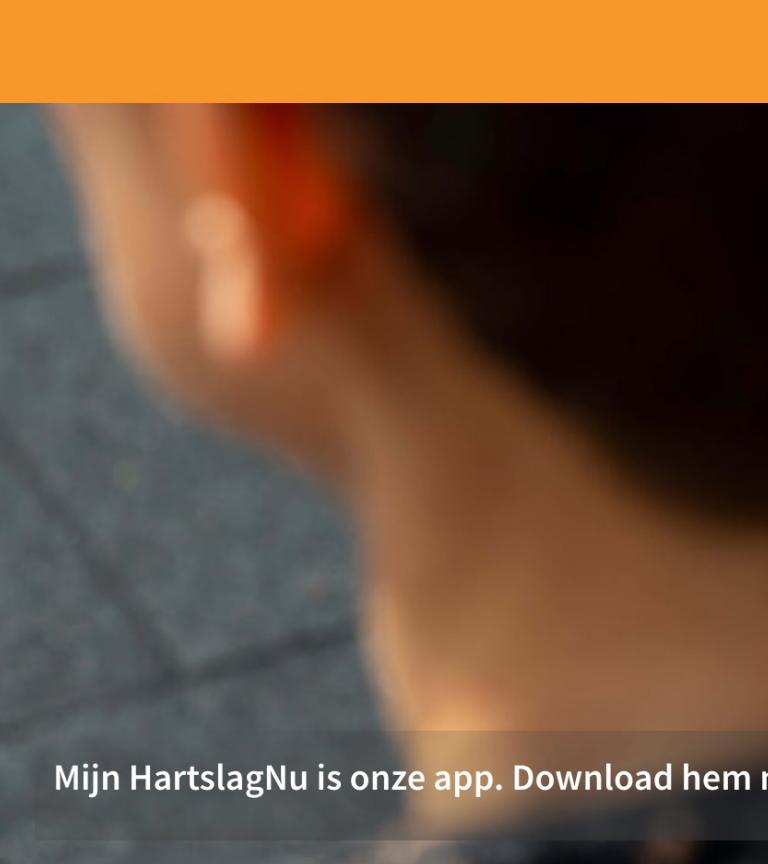
<>

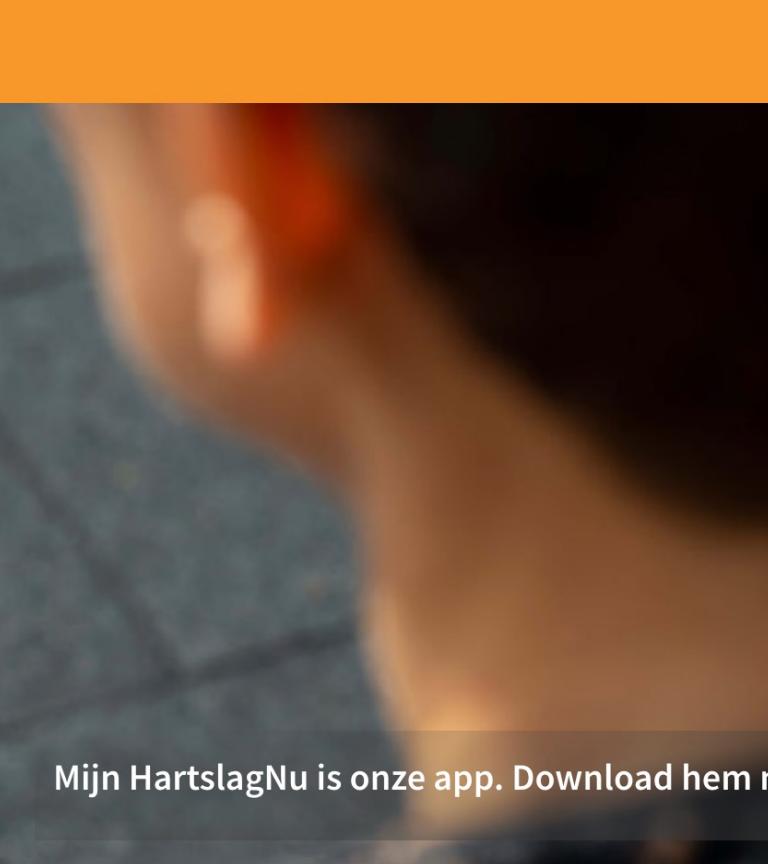
<>

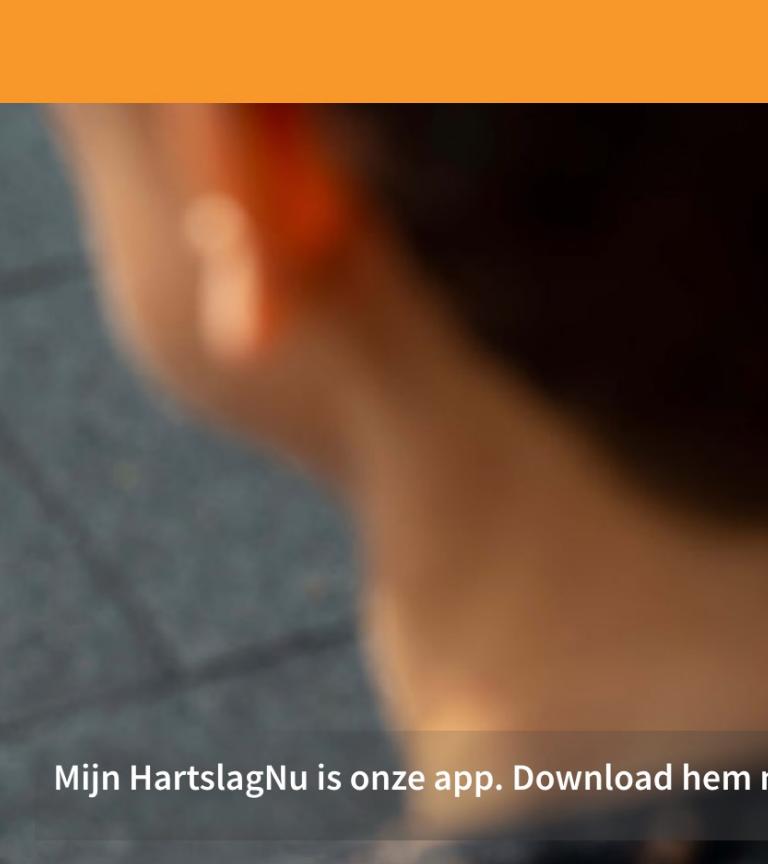
<>

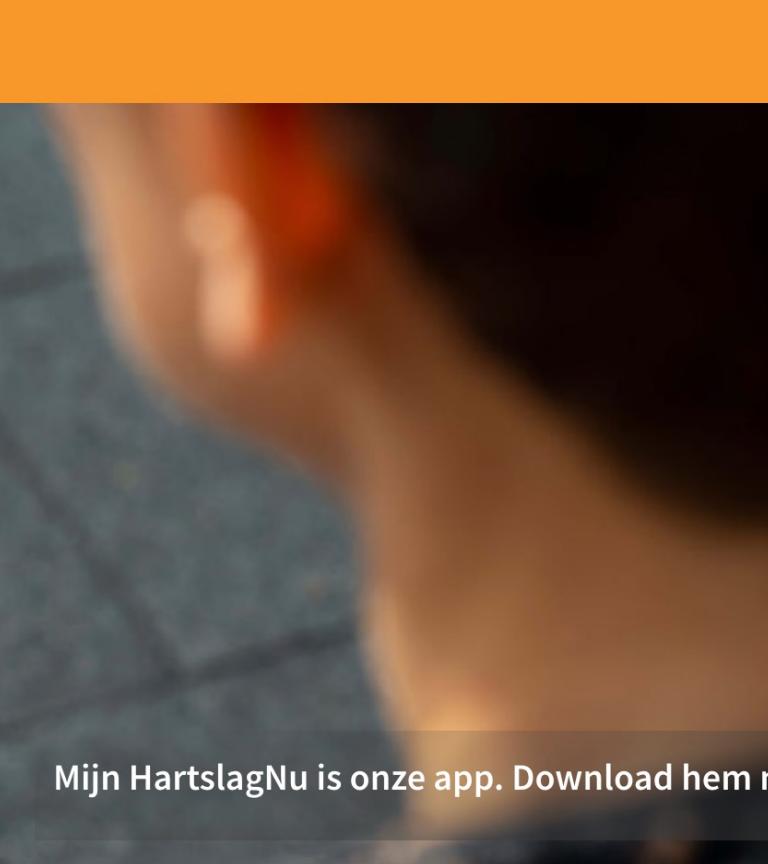
<>

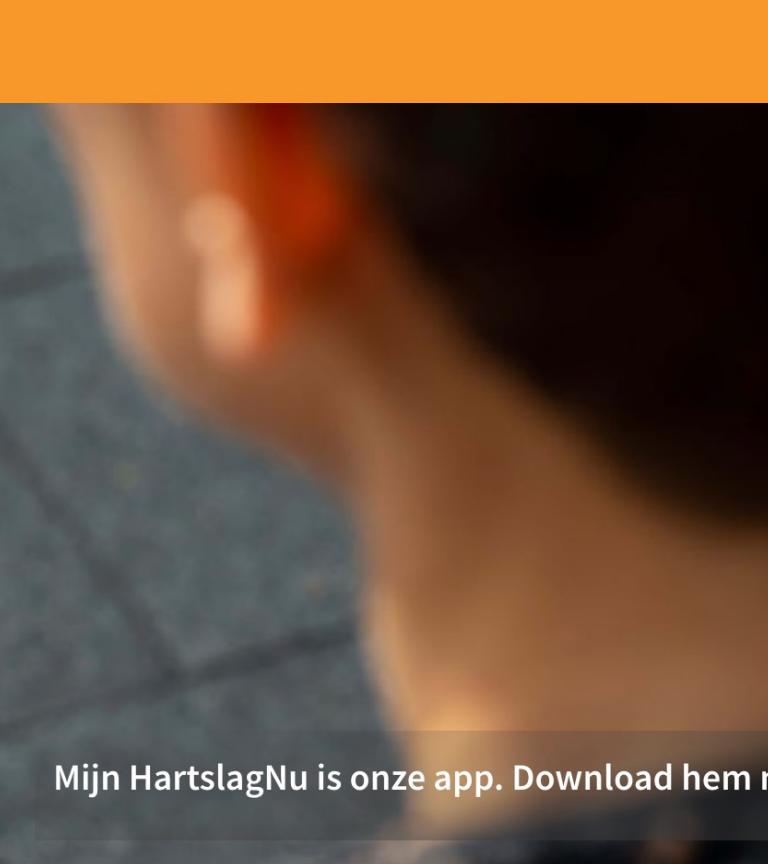
<>

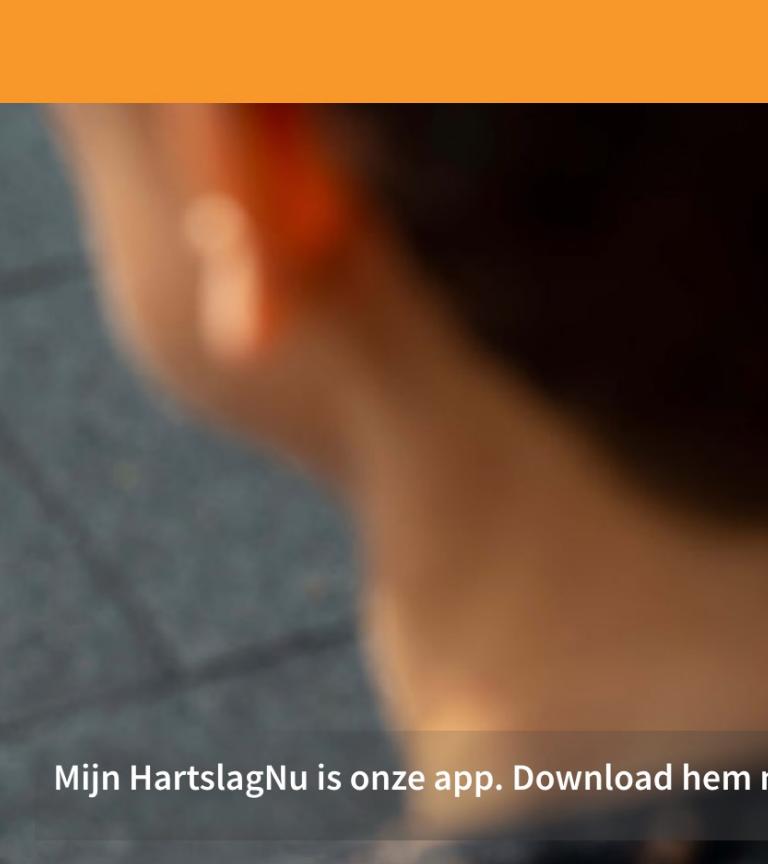
<>

<>

<>

<>

<>

<>



Just Vervaart

+316

Wachtend

Vrijwilliger

## Niet Storen: Aan



Je wordt niet opgeroepen

🕒 Tijdlijn

👤 Mijn profiel

+AED's

? Open HartslagSamen

? Vragenlijsten

gMaps Mijn omgeving

— Overzicht

— Resultaten

❤️ App

✉️ Over HartslagNu

✉️ Contact

✖️ Uitloggen



491

Vrijwilligers in de straal 2000  
meter

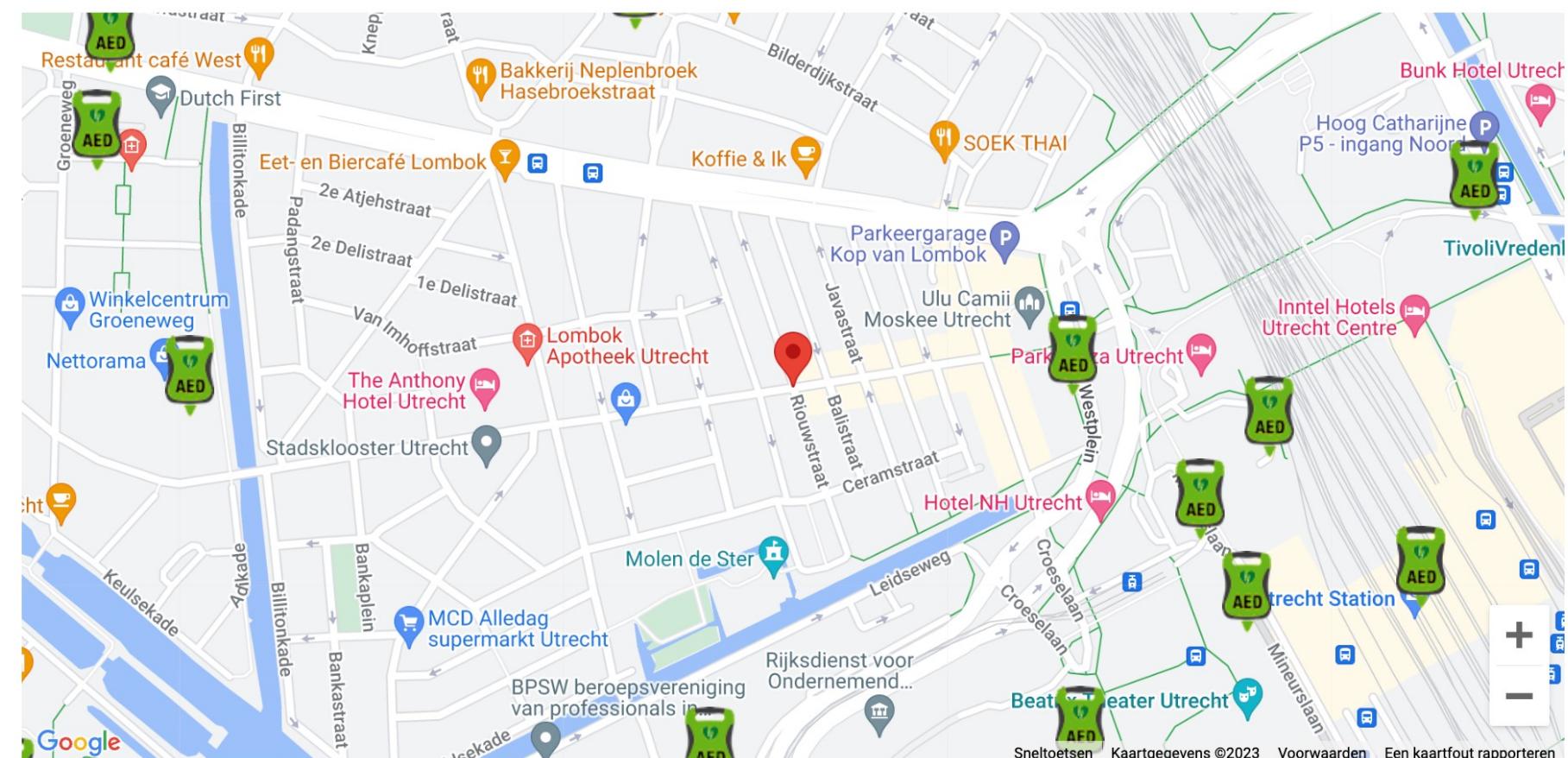
87

Actieve AED in de straal 2000  
meter

5

Inactieve AED's in de straal 2000  
meter

## AED's in de buurt



# API ‘hacking’

[https://webapi.heartsafeliving.com/api/Aed/GetAllInRange?  
latitude=52.091371&  
longitude=5.100946&  
range=2000](https://webapi.heartsafeliving.com/api/Aed/GetAllInRange?latitude=52.091371&longitude=5.100946&range=2000)

```
"Distance": 0.01885190251672252,
"AedId": 7082,
"AedType": "Medtronic - LifePak CR Plus",
"AedModel": null,
"Description": "Buitengevel hoofdingang Stadhuis Utrecht slotcode 2803",
"PurchaseDate": "2022-08-02T00:00:00",
"Address1": "Korte Minrebroederstraat",
"Address2": null,
"Address3": null,
"HouseNumber": 2,
"HouseNumberEx": "",
"City": "UTRECHT",
"Zip": "3512GG",
"CountryCode": "NL",
"State": null,
"MobilityType": null,
"AedAvailableWeekDays": [
{
    "AvailableWeekDay": 0,
    "AedId": 7082,
    "FromHour": 0,
    "FromMin": 0,
    "ToHour": 23,
    "ToMin": 59,
    "Aed": {
        "AedId": 7082,
        "AedType": "Medtronic - LifePak CR Plus",
        "AedModel": null,
        "Description": "Buitengevel hoofdingang Stadhuis Utrecht slotcode 2803",
        "PurchaseDate": "2022-08-02T00:00:00",
        "Address1": "Korte Minrebroederstraat",
        "Address2": null,
        "Address3": null,
        "HouseNumber": 2,
        "HouseNumberEx": "",
        "City": "UTRECHT",
        "Zip": "3512GG",
        "CountryCode": "NL",
        "State": null,
        "MobilityType": null,
        "BatteryBestBefore": "2024-10-21T00:00:00",
        "Security": null,
        "SecurityCode": null,
        "SerialNumberHsl": "45405131",
        "SerialNumber": null,
        "ElectrodesBestBefore": "2024-06-01T00:00:00",
        "FormattedElectrodesBestBefore": "01-06-2024",
        "FormattedBatteryBestBefore": "21-10-2024",
        "AedMedias": [],
        "Volunteer": null,
        "DeviceStatus": "Active",
        "SpottedBy": null,
        "NextReminder": "2024-03-01T00:00:00",
        "IsPublic": true,
        "IsPhAvailability": true,
        "PlateformKeyCreate": null,
        "FullAddress": null,
        "PlateformKeyModify": "1E334D80A0C7B48CBA38CA1F14F5556D2DB4898D04B45E13BB19FC7CE4A7EE53",
        "Latitude": 52.091839,
        "Longitude": 5.119793,
        "UserId": null
    }
}
```

# Alle AED's

Toen ik dit had gevonden kreeg ik een idee hoe ChatGPT me kon helpen om info over alle AED's op te vragen.



Ik heb een API die op basis van coordinaten info ophaalt binnen een straal van 2 kilometer. Schrijf een Python script dat info voor heel Nederland ophaalt. Leg je aanpak uit.



Het ophalen van informatie voor heel Nederland met een straal van 2 kilometer kan behoorlijk veel data opleveren, afhankelijk van de dichtheid van je punten. Hier is een algemene aanpak die je kunt volgen, samen met een basis Python script:

## Aanpak:

1. Bepaal de coördinaten van de grenzen van Nederland. De ruwe coördinaten zijn ongeveer:  
Noord: 53.5, Zuid: 50.8, West: 3.3, Oost: 7.2.
2. Maak een grid van coördinatenpunten die heel Nederland dekt, met tussenruimtes van ongeveer 2 kilometer.
3. Roep voor elk van deze coördinatenpunten je API aan en verzamel de gegevens.
4. Sla de gegevens op in een gewenst formaat.

## Script:

python

Copy code

```
import requests

def ophaal_info_van_api(latitude, longitude):
    # Je API URL en parameters, aan te passen naar jouw behoeften
    url = "https://jouwapiurl.com/data"
    params = {
        'latitude': latitude,
        'longitude': longitude,
        'radius': 2.0
    }
    response = requests.get(url, params=params)
```



Hulp bij een hartstilstand is nooit ver weg, behalve in grensgemeenten

Als elke minuut telt: vaker vrijwilligers bij reanimatie



Jouw regionale nieuws

Niet iedereen heeft AED in de buurt, toch is er wel vaak genoeg hulp

Relatief weinig AED's in Westerwolde, toch toename van opkomst bij noodgeval



Opkomst AED-vrijwilligers in Flevoland leidt tot snellere reacties bij hartstilstanden



NOS

NOS Nieuws • Vrijdag 20 oktober, 06:00

## Geen geld uit de muur: geldautomaten te vaak in storing

Geldautomaten waren vorige maand veel vaker in storing dan de banken hadden beloofd. Gemiddeld was 1 op de 20 automaten van Geldmaat in september defect, om precies te zijn 5,7 procent van de automaten. Dat blijkt uit gegevens die de NOS samen met regionale omroepen verzamelde.



AFRONDING

# Samenvattend

1. AI ken je helpen om code te schrijven voor tal van onderzoeken. Google Dorking, Wikidata, Open Street Map en Python.
2. Experimenteer, maar vergeet niet de uitkomsten van je onderzoeken te verifiëren met andere bronnen.
3. Met bijna ieder onderzoek leer ik weer nieuwe technieken en ontdek nieuwe mogelijkheden. Het is vooral een kwestie van doen!

# VRAGEN?

HE

Vertel kort iets over een mand.



Een mand is een container gemaakt van natuurlijke materialen die wordt gebruikt om voorwerpen te vervoeren of op te slaan.



HE

Veel korter.



Een mand is een container voor voorwerpen.



HE

Nog korter



Een mand is een container.



HE

Veel korter



Een mand.



HE

Nog korter



Mand.

