



# FRONT MANIA

CONFERENCE

# 2023

October 12th

## FRONTMANIA CONFERENCE 2023 special

Timetable

Floorplan

Articles



# welcome TO FRONTMANIA CONFERENCE 2023!

## Greetings, FrontManiacs!



**P**repare to embark on a cosmic journey as we unveil the second edition of FrontMania Magazine. Our gratitude goes to our stellar partners Rabobank, IO, Divotion, Sogeti, ABN AMRO, Pan-company, Capgemini and Ordina. Whose support has made this voyage into the deep space of the frontend frontier possible. We also extend our heartfelt thanks to the brilliant minds and code-wielding heroes who toiled tirelessly to craft this magazine – your dedication is out of this world. But the biggest thank you, goes to..... YOU!

### introduction:

Randy, The Chief Experience Officer of Kadena and one of the founders behind this amazing conference. Our odyssey began in 2019, when a couple of tech aficionados dared to dream of a conference where code, fun, and sheer madness would redefine the very essence of gatherings. A conference born of developers, for developers. While I may not have written much code recently, I remain firmly rooted in the cosmos of development.

I'll never forget the inaugural conference, when I made a daring entrance on a remote-controlled skateboard (living on the edge, indeed), following an electrifying performance by dancers enveloped in epic lights and smoke. The sequel saw me donning the mantle of a FrontMan (my superhero alter ego 🦸), with Seb Lee Delisle's laser spectacle in tow. To me, that remains the



**Randy Daal**  
is Board Member  
of Frontmania  
Foundation & Chief  
Experience Officer  
@ Kadena.



**Ramin Mohammadi**  
is Board Member  
of Frontmania  
Foundation &  
Senior Frontend  
Developer @ IO.

quintessential opening act. Then, the eerie silence descended due to a certain galactic pandemic. Honestly, I never thought we'd see another FrontMania. But then, we restarted the engine as a foundation, propelling this conference further than I ever dreamed possible.

So, with immense pleasure, I open the opening of this second magazine alongside my longtime friend and fellow FrontManiac, Ramin (together, we're known as Team R&R).

Ramin: Although I became part of the FrontMania board members only a year ago. Together with Randy, I've been part of the journey right from the beginning, witnessing the sparks that ignited FrontMania into existence. Working on frontend projects for over more than a decade seeing the rise and fall of many tooling and frameworks. My love for frontend is driven by its big ecosystem where there's a lot to learn and master, to be able to create awesome projects.

We believe that Frontend development is more than just code; it's a thriving, dynamic community. FrontMania was created by and for like-minded Frontend(Maniacs) to explore, learn, and have a great time doing what we love most. So being part of the Board is an honor for me!

### thanks

We're thrilled to declare that the fourth edition of this conference will outshine its predecessors, featuring an enthralling mix of both emerging talent and established luminaries. And, most importantly, it promises a universe of fun.

We invite you to delve deep, forge new connections, and, above all, revel in the cosmic adventure that awaits within these pages. May the Source be with you on this thrilling journey! 🦸🌌🚀🌟



# We make technology work

Wij zijn een technologiebedrijf in hart en nieren waar jij in staat wordt gesteld het beste uit jezelf naar boven te halen. Of je nou een ervaren of beginnende front-end developer bent, jouw talent kan altijd verder groeien. Kom jij onze community van front-end professionals versterken? Kijk op:

[www.werkenbijsogeti.nl](http://www.werkenbijsogeti.nl)

## Daarom Sogeti:

- *Werken voor topmerken in Nederland*
- *Breed opleidingsprogramma*
- *Ruimte voor persoonlijke ontwikkeling*
- *Kennisdelen bij Frontend Lightning Talks*
- *Open en collegiale sfeer*
- *Technische kwaliteit op nummer 1*
- *Ruimte voor initiatieven en meer dan alleen developen*



# Focus op front-end development

Bij Divotion werken mensen voor wie front-end development weinig geheimen kent. Wij adviseren, trainen, ontwikkelen en spreken op conferenties.

Wil jij ook 100% focus op front-end development? Bekijk onze vacature via de QR-code.



“ WE LEARN FROM EXPERTS AND FROM EACH OTHER IN THE MASTERCLASS FRONT-END DEVELOPMENT ”

Discover the endless possibilities at Capgemini. Read the full article to learn more about the introductory Masterclass Front-end Development.

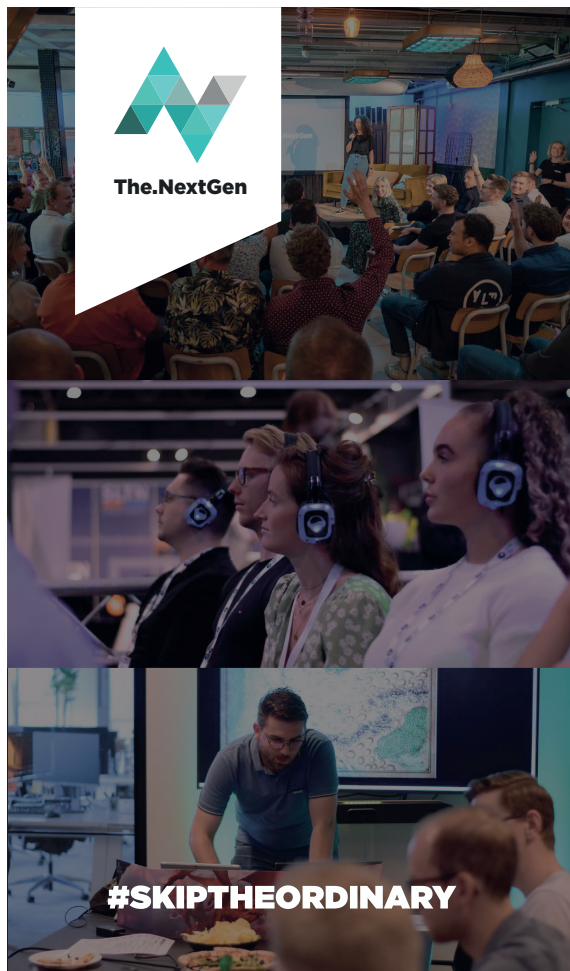


Read the Dutch article here



GET THE FUTURE YOU WANT

[www.werkenbijcapgemini.nl](http://www.werkenbijcapgemini.nl)



# EARN LIKE A FREELANCER WITH THE BENEFITS OF AN EMPLOYER?

Work on the most challenging IT projects, earn an absolute top salary, and be part of the best Tech Community with senior Java experts.

- ▶ 70% of your hourly rate is for you
- ▶ Always assured of a basic income
- ▶ Freedom in project choice

#SKIPTHEORDINARY

SCAN THE QR CODE AND VISIT OUR WEBSITE



```
mirror_mod.use_x = False
mirror_mod.use_y = True
mirror_mod.use_z = False
elif operation == "MIRROR_Z":
    mirror_mod.use_x = False
    mirror_mod.use_y = False
    mirror_mod.use_z = True

#selection at the end -add back the deselected
mirror_ob.select= 1
```



# oklch() color format is coming to CSS

Nowadays, designers don't choose each color manually. They generate a palette by a system. And they need some solid math as the foundation for this system. Also, we now have new screens which shows +30% colors. rgb(), hsl(), and hex can't handle this new world order. This is why CSS Colors 4 specification brought a new color format oklch().

- Like hsl(), oklch() encodes colors much closer to the way of how humans think about colors. Instead of using amount of red, green, blue, it uses lightness, saturation and hue. (image 1)
- oklch() color functions has 3 or 4 numbers to define color: oklch(L C H) or oklch(L C H /  $\alpha$ ).
- L is perceived **lightness**. It ranges from 0% (black) to 100% (white). The % is mandatory, even for zero values.
- C is **chroma**, the saturation of color. It goes from 0 (gray) to infinity. In practice there is actually a limit, but it depends on a screen's color gamut and each hue has a different maximum chroma. For current screens, the value will be always below 0.37.
- H is the **hue** angle. It goes from 0 to 360, through red 0, yellow 70, green 120, blue 200, purple 300 and then back to red. Since it is an angle, 0 and 360 encode the same hue.
- $\alpha$  is **opacity** (0-1 or 0-100%). (image 2)

```
.bw {
  color: oklch(0% 0 0); /* black */
  color: oklch(100% 0 0); /* white */
  color: oklch(100% 0.2 100); /* also white, any hue with 100% L is white */
  color: oklch(50% 0 0); /* gray */
}

.colors {
  color: oklch(80% 0.12 100); /* yellow */
  color: oklch(60% 0.12 100); /* much darker yellow */
  color: oklch(80% 0.05 100); /* quite grayish yellow */
  color: oklch(80% 0.12 225); /* blue, with the same perceived lightness */
}

.opacity {
  color: oklch(80% 0.12 100 / 50%); /* transparent yellow */
}
```



OKLCH color examples

## NATIVE COLOR MANIPULATIONS

Why do we need oklch() if we already have similar hsl()? The main problem with hsl() is that L component of the colors is not very

### Lightness

1



### Alpha



OKLCH axes

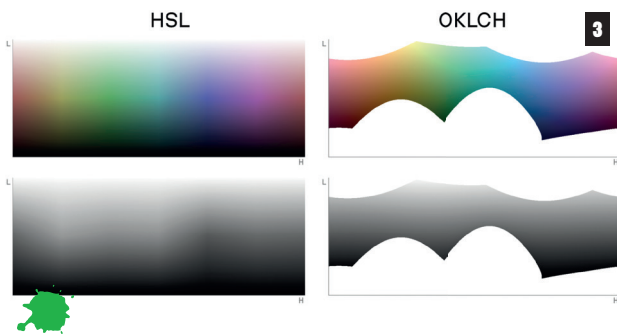
### Chroma



### Hue



useful. It has no uniformity, the same number will have different lightness for our eyes depending on the hue. L component in `oklch()` is much more uniform and closer to how our eyes see (see image 3).

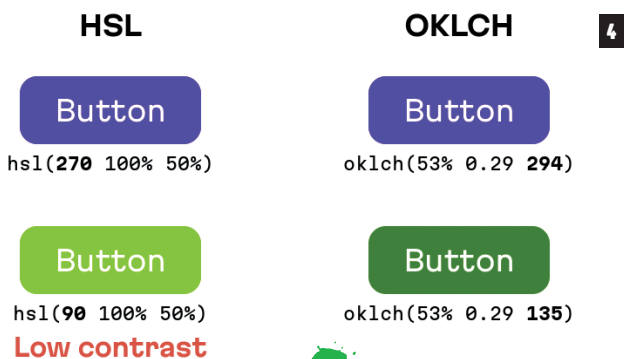


Hue-Lightness slice of HSL and OKLCH spaces with the same chroma/saturation and black-and-white versions below. HSL lightness is not consistent across hue axes.

It happens because HSL is a cylindrical color space. Every hue has the same amount of saturation (0–100%). But in reality, our displays and eyes have different max saturations for different hues. HSL hides this complexity by deforming the color space and extending colors to have the same max values.

Here are a few real use case examples to demonstrate this problem:

- Adding 10% lightness will have different results for blue and purple colors. SASS users may remember how `darken()` generates unexpected results.
- Hue changes (for instance, producing an error-like red color using a company's accent color) could also change lightness, thus making text unreadable.



In HSL, hue changes could lead to accessibility issues from low contrast

Because of L issue, `hsl()` can't be used to generate palette in design systems. For instance, Stripe in "Designing accessible color systems" post ask community not to use HSL for palette generation. OKLCH color space is a much better way for color transformation. Every color with the same L will look similar. This is why palette generators like Huetone already use OKLCH.

This is why upcoming native color transformation in CSS will allow you to choose color space to do transformation.

```

:root {
  --accent: oklch(70% 0.14 113);
}
.error {
  /* Red version of accent color */
  background: oklch(from var(--accent) 1 c 15)
}
.button:hover {
  /* 10% lighter version */
  background: oklch(from var(--accent) calc(1 + 10%) c h)
}

```

In this syntax, you define a new color from another color (so the feature is called relative colors). You are using any color function, write ``from`` and then color in any format (including CSS Custom Properties).

Each component (l, c, h for `oklch()`) after from X can be:

- A letter (l, c, h), which indicates to keep the component the same as it was in the origin color.
- A `calc()` expression. You can use a letter (l, c, h) instead of a number to reference the value in the origin color.
- A new value, which will replace the component.

With CSS Custom Properties, we can define the `:hover` logic just once and then create many variants simply by changing the source color (see image 6).

## BIO



Andrey Sitnik is principal front-end in Evil Martians. He is the creator of PostCSS, Autoprefixer, and Browserslist.

```

.button {
  background: var(--button-color);
}

.button:hover {
  /* One :hover for normal, secondary, and error states */
  background: oklch(from var(--button-color) calc(1 + 10%) c h);
}

.button {
  --button-color: #var(--accent);
}

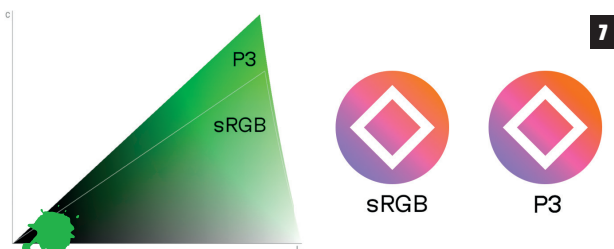
.button.is-secondary {
  --button-color: #var(--dimmed);
}

.button.is-error {
  --button-color: #var(--error);
}

```

### wide-gamut colors

New screens (like OLED or in all new Apple devices) can show +30% more colors. With this new wide-gamut you can make a more vivid landing page. Or you can be more flexible in your design system's palette.



Newly available P3 colors for green on the left. Real-world icon comparison with sRGB vs. P3 on the right.

You can't use rgb() or hsl() to encode these new colors. And oklch() can help you here as well (see image 8)

```

.martian {
  background: #oklch(69.73% 0.155 112.79);
}

@media (color-gamut: p3) {
  .martian {
    background: #oklch(69.73% 0.176 112.79);
  }
}

```

### moving to oklch()

Right now all major browsers support oklch(): Chrome since 111, Safari since 15.4, and Firefox since 113.

You can convert colors in your source code by oklch.com converter or by convert-to-oklch script. (image 9)

By moving to OKLCH right now, you will have a few improvements in code maintenance:

```

.button {
  /* Blue */
  background: #6ea3db;
}

.button:hover {
  /* More bright blue */
  background: #7db3eb;
}

.button.is-delete {
  /* Red with the same saturation */
  background: #d68585;
}

```

```

.button {
  /* Blue */
  background: #oklch(70% 0.1 250);
}

.button:hover {
  /* A brighter blue */
  background: #oklch(75% 0.1 250);
}

.button.is-delete {
  /* Red with the same saturation */
  background: #oklch(70% 0.1 20);
}

```

You will be able to find some contrast issues just by comparing L numbers. But contrast math is really tricky, and it is better to use special tools (see image 10).

```

.text {
  /* ERROR: a 20% lightness difference is not sufficient for good contrast and accessibility */
  background: #oklch(80% 0.02 300);
  color: #oklch(100% 0 0);
}

.error {
  /* ERROR: colors have a slightly different hue */
  background: #oklch(90% 0.04 30);
  color: #oklch(50% 0.19 27);
}

```

You will be able to apply simple color modifications right in the code and get predictable results as shown in image 11.

```

.button {
  background: #oklch(50% 0.2 260);
}

.button:hover {
  background: #oklch(60% 0.2 260);
}

```

Since OKLCH is much closer to real-life color, using oklch() in CSS will educate developers and lead the community to an overall better understanding of color.

By moving to OKLCH today, we're preparing ourselves for the not-so-distant future where native color modification will be available in CSS. OKLCH is the best color space for color transformations, and it's better to become familiar with its axes by using them in color definitions now. 🌈





## Kom solidsharen bij Pancompany!

- Inspraak in de keuze van projecten die jij wilt doen
- IT-experts om mee te sparren en je passie voor het vak mee te delen
- Werken op een plek waar jouw ontwikkeling voorop staat
- Vast contract, gegarandeerd salaris plus een aanzienlijk deel van de omzet
- 100% focus op wat jou energie geeft, namelijk je vak!

# Solidshare iets voor jou?

Ben jij op zoek naar de perfecte combinatie van ondernemerschap én zekerheid? Wil je samenwerken met echte vakexperts in projecten voor Nederlandse topbedrijven? Ondersteund worden in jouw ontwikkeling en geen goed maar een uitstekend inkomen verdienen? Solidshare it is!

## Ready to join Pancompany?



Ontdek wat Solidshare jou oplevert! Scan de QR-code voor de salarischek of solliciteer direct op topvacatures.

**solidshare** 

### React Engineer

**Salary Indication:**  
55.000 - 80.000 EUR

**Location:**  
Amsterdam

**Technologies:**  
React, JavaScript, Vue, NextJS



### Senior Front-end developer

**Salary Indication:**  
50.000 - 80.000 EUR

**Location:**  
Vianen

**Technologies:**  
JavaScript, TypeScript, React, Angular, Vue, Frontend, NextJS, NodeJS, HTML5, CSS



### Front-end developer

**Salary Indication:**  
40.000 - 70.000 EUR

**Location:**  
Nieuwegein

**Technologies:**  
JavaScript, HTML5, CSS, Angular, React, Vue



### Software Development Consultant (Frontend)

**Salary Indication:**  
70.000 - 100.000 EUR

**Location:**  
Hilversum

**Technologies:**  
Technologies: D3, JavaScript, React, Vue, Angular



Find all these vacancies and more

at [Devitjobs.nl](https://devitjobs.nl)



# MAKING digital ACCESSIBILITY FUN with everyday items

In today's digitally driven world, accessibility is a fundamental aspect of our online experiences. Digital accessibility ensures that everyone, regardless of their abilities, can access and enjoy the vast array of online content. However, discussing accessibility doesn't have to be dull or technical. In this article, we'll explore digital accessibility through fun and relatable examples using everyday household items that anyone can try themselves.



### The “Blindfold” Typing Challenge

Imagine browsing the web without seeing the screen; this is the daily reality for individuals with visual impairments. To better understand their experience, you can try the “blindfold” typing challenge using a scarf, a bandana, or even by turning off your screen. Blindfold yourself and try to navigate websites, send emails, or simply type a message to a friend. This exercise will help you appreciate the importance of screen readers, keyboard navigation, and other accessibility features that make digital content accessible to everyone.

### The “Muffled Speaker” Test

For individuals with hearing impairments, captioning is a lifeline. There are two ways to simulate this:

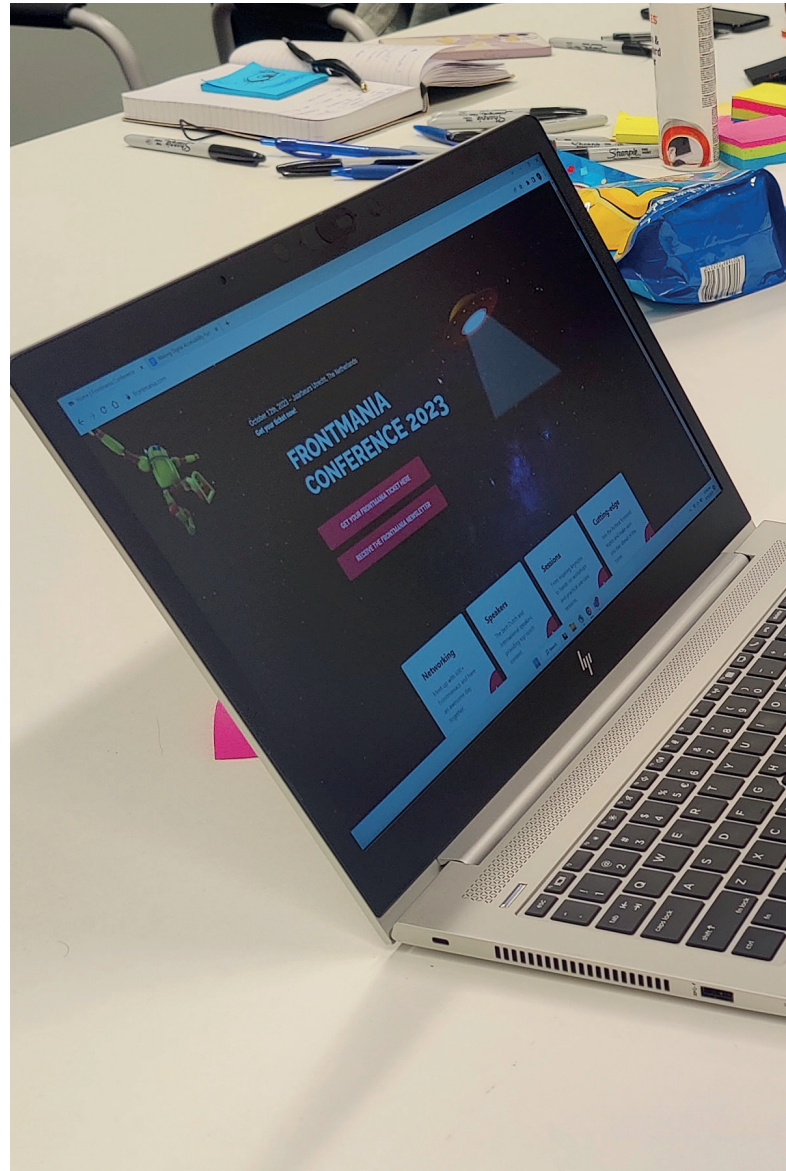
- Play a video or podcast and mute it, this will simulate total deafness.
- Play a video or podcast while wearing noise-canceling headphones, doing this will make the audio very muffled and hard to understand simulating partial deafness.

For both methods, try to follow the content by relying solely on the subtitles or closed captions. This exercise will emphasize the value of accurate and synchronized captions, ensuring that deaf or hard-of-hearing individuals can enjoy multimedia content just as much as anyone else.

### The “Color Contrast” Challenge

Color contrast plays a vital role in enhancing accessibility for individuals with visual impairments or color blindness. To gain a deeper appreciation of this concept, you can experiment with colored sunglasses or craft simple glasses using colored cellophane. When you wear these glasses, embark on two little fun activities:

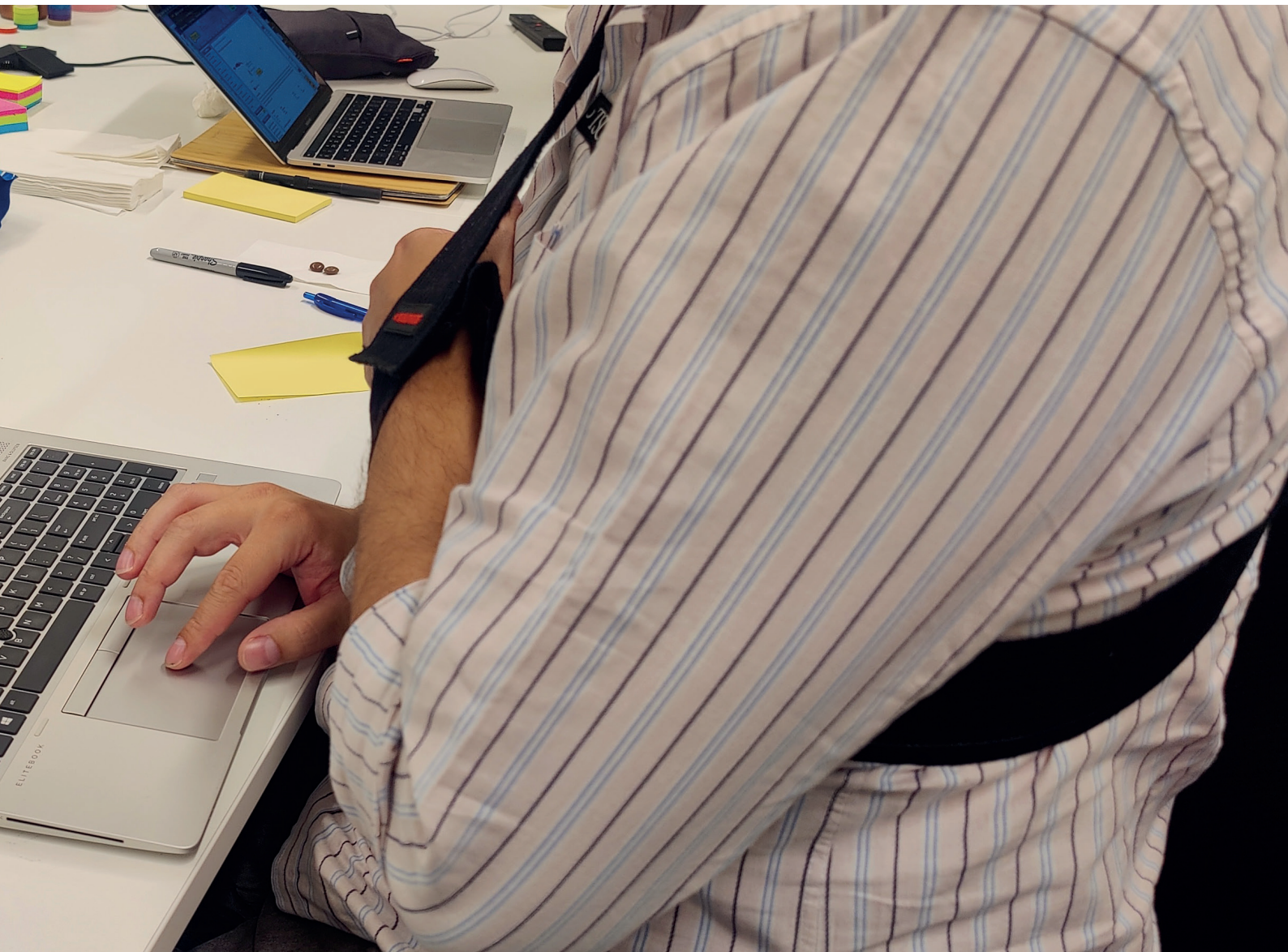
First, challenge yourself to match socks by color while wearing your tinted glasses. This task will help you experience the impact



### BIO

**Dražen Janković** is Senior Technology Consultant at Sogeti with over 20 years of experience in frontend and a specialization in accessibility. Always looking to experiment with new CSS and JS technologies to bring the best possible user experience to all users.





of altered color perception on everyday activities, as individuals with color blindness often encounter.

Secondly, take a leisurely walk outdoors and observe the world through your colored lenses. Each time you encounter an object, compare your perception of its color with its actual hue. This experiment will shed light on the different ways people perceive colors and emphasize the importance of creating digital content with sufficient color contrast to accommodate diverse visual experiences.

### **The “One-Handed” Challenge**

Temporary or permanent hand injuries can make using a keyboard and mouse challenging. To empathize with this experience, try the “one-handed” challenge. Put your dominant hand to your side or behind your back, and attempt to perform everyday tasks

like browsing, shopping online, or composing emails using just your non-dominant hand. This experiment highlights the need for keyboard shortcuts, voice recognition, and other alternative input methods to accommodate users with limited hand mobility.

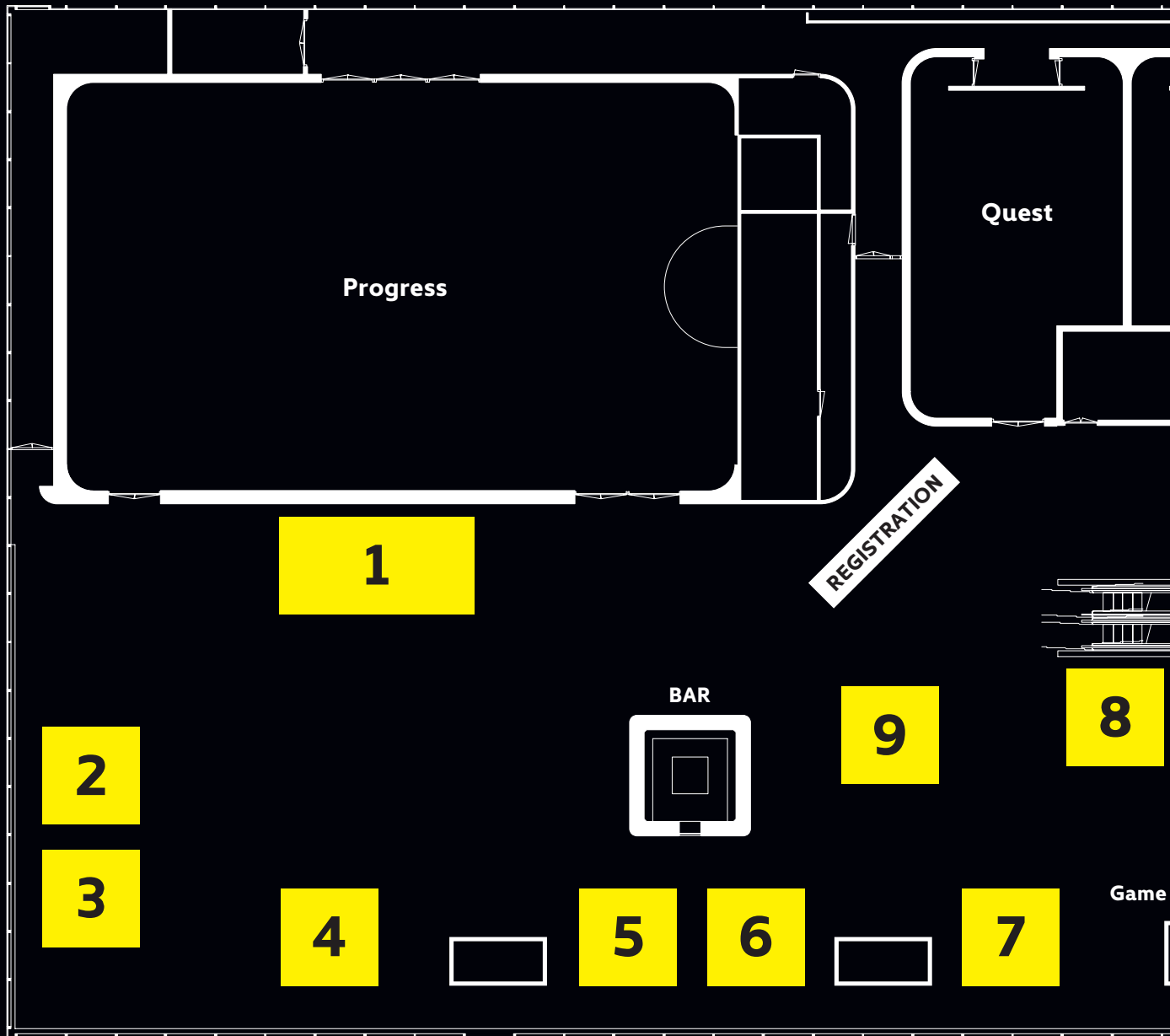
Digital accessibility isn't just a buzzword; it's a crucial aspect of creating an inclusive online world. By engaging in these little experiments, you can gain a deeper appreciation for the challenges faced by individuals with disabilities in their daily digital interactions. Remember, making digital content accessible isn't just a legal requirement; it's a way to ensure that everyone can participate fully in our increasingly digital society. So, the next time you design a website or create content, consider these everyday household lessons in digital accessibility and make the internet a more inclusive and enjoyable place for all. 🌟

# Timetable Frontmania 2023

Time/Room	Progress	Mission 1	Mission 2
9.30	Entrance open		
10:00 - 10:10	Opening		
10.10 - 10.40	Keynote: The Road to Front-end Mastery <b>Bobby van der Sluis</b>		
10.40 - 11:00	Transit zone: Coffee break		
11.00 - 11.45	Critical Security Mistakes in Frontend Applications <b>Dan Neciu</b>	JSON Schema Validation; A dating show between backend and frontend <b>Jorrik Klijnsma</b>	Generative AI: A personal journey on revolutionizing workflows <b>Constantijn Herpers</b>
12.00 - 12.45	How the "transformer model" is like a framework for AI <b>Vince Liem</b>	Voice UI - let your users do the talking <b>Drazen Jankovic</b>	Animating Delight: Elevate UX with Framer Motion Microinteractions <b>Vilva Athiban P B</b>
12.45 - 13.45	Transit zone: Lunch break		
13.45 - 14.30	Keynote: Visualizing Connections <b>Nadieh Bremer</b>		
14.30 - 15.15	Qwik: The Next-Generation React Framework? <b>Daniel Frey</b>	Faster, Smarter, Better: Unlocking the Full Potential of Modern Browser APIs and PWA <b>Tim Lavreniuk</b>	Holistic Approach to Sustainable IT <b>Emiel Kwakkel</b>
15.15 - 15.45	Transit zone: Coffee & tea break		
15.45 - 16.30	Making Interactions Accessible to All Users <b>Anuradha Kumari</b>	What's new in CSS? <b>Bramus van Damme</b>	Navigating Design System Maturity in Tech Organizations <b>Joran Quinten</b>
			Automating your Developer Workflow with Playwright's UI Mode <b>Nimo Beeren</b>
			Angular Magic with Angular Schematics <b>Jennifer Wadella</b>
16.45 - 17:30	The Problems Micro Frontends Won't Solve That No One Wants to Talk About <b>Jennifer Wadella</b>	Fine-Tuning Your Reactivity (Harnessing the Power of Signals) <b>Salama Ashoush</b>	Building a JavaScript Client for smart contracts <b>Albert Groothedde</b>
17.30 - 18.30	Transit zone: Drinks & Music		

Expedition	Quest	Workshop area
<p>The Future of 3D elements on the Web <b>Tim Beeren</b></p>	<p><b>Check the website for the latest info!</b></p>	<p>The new synergy of AI&amp;web3: building an AI NFT Generator <b>Jessy The</b></p>
<p>TypeScript inaccuracies? Let's fix those types! <b>Emiel van Galen</b></p>	<p>Building a Design System that Scales with Your Organization: Tips and Tricks <b>Dennis Spierenburg</b></p>	
<p>From Punch Cards to AI's Upcoming Revolution <b>Boris Martinović</b></p>	<p>An architectural pattern on Ui library workflows with Storybook and Azure <b>Rodney Wormsbecher</b></p>	<p>The No-Framework WebApp workshop <b>Tom Raaff</b></p>
<p><b>Check the website for the latest info!</b></p>	<p>GraphQL Explained Through Performance Testing <b>Roy Derks</b></p>	
<p>Nuxt 3: Just Vue and a bit of magic? <b>Alexander Lichter</b></p>	<p>Rethinking desktop applications with progressive web apps <b>Nico Martin</b></p>	

# Floorplan Frontmania 2023



**FRONT  
MANIA**

**CONFERENCE  
2023**  
October 12th

- |             |   |          |   |
|-------------|---|----------|---|
| Rabobank    | 1 | ABN AMRO | 8 |
| Divotion    | 2 | Sogeti   | 9 |
| Pancompany  | 3 |          |   |
| Pink Pollos | 4 |          |   |
| Capgemini   | 5 |          |   |
| Ordina      | 6 |          |   |
| Lab Digital | 7 |          |   |

\*Subject to change





electe  
ob  
fier c  
JFK



# FRONT MANIA

CONFERENCE

# 2023

October 12th

FRONTMANIA THANKS OUR  
SPONSORS & PARTNERS  
FOR MAKING FRONTMANIA  
CONFERENCE 2023 POSSIBLE!

Main Sponsor



Rabobank

Main Partner



Silver Partners



# AUTOMATED MODERATION USING OPENAI & Nuxt

With AI systems on the rise and within an arm's reach these days, I think it's valid to take a look at where they can play a supporting role in software development right now and for the future. The OpenAI platform allows you to make use of published algorithms in an accessible way. So it's a perfect way to get your feet wet!

OpenAI is an AI research and deployment company with a self-stated mission to ensure that artificial general intelligence benefits all of humanity. OpenAI has been making waves with GPT-3, which performs a variety of natural language tasks, Codex, which translates natural language to code, and DALL-E, which creates and edits original images.

## GPT-3

The endpoints that OpenAI provides are all built on Generative Pre-trained Transformer 3 (GPT-3). The deep learning algorithm that is specifically trained to handle user text input. The GPT-3 algorithm has been making waves with the ChatGPT chatbot implementation which had mixed results. At this point, the results of the moderation endpoint are much more stable.

In this tutorial we'll take a look at a specific API, namely the one that helps you in intercepting user input by applying natural language interpretation to use as moderation. This is a perfect example of leveraging AI to help with reducing the load for a very mundane task: assessing whether user submitted content should be flagged.

## Let's dive in!

To get started, you need to sign up at OpenAI and generate a key. The process should be fairly straightforward and there's no payment details required. The trial offers a generous amount of usage for you to play around with various APIs.

Next we'll set up a simple Nuxt project. I am using Nuxt because I believe it uses a very readable syntax coming from whatever background you might have. If you know HTML, JavaScript, you can very easily interpret the code. You should be able to recreate this example in any other framework with little effort though.

Using Nuxt, you can use the following commands to scaffold out a starter project and adding the OpenAI package (Nuxt docs prefer using yarn but feel free to follow the npm or pnpm steps):

```
npx nuxi init openai.moderation
cd openai.moderation
```

```
yarn add openai
yarn install
yarn dev -o
```

This should result in the starter project running on (typically) `http://localhost:3000`. Now open the project in your favorite IDE, and let's get started!

## Configure the key

Create an `.env` file in the root of the project containing this line (replace with your personal key), so we can use it to sign the requests to OpenAI:

```
OPENAI_API_KEY=ALWAYSKEEPSECRETSSTOYOURSELF
```

Next, open the `nuxt.config.ts` and make sure it looks like this: **Listing 1**

If you're wondering where the imports are, Nuxt supports auto importing and thus resolves a lot of imports automatically for you. Neat!

## Setting up the api

In order to communicate to the OpenAI endpoint, we'll need a server of our own. In Nuxt, adding an API endpoint is just as easy

### BIO



*Joran's passion is getting people to love technology and getting technology to play nice. He works as an interaction developer with ❤️ for web, tech, science & tinkering with stuff.*

*Focused on innovation and tech ambassador at Jumbo Supermarkten. Writes, tweets, toots and speaks every now and then. Loves to talk shop. Owns a cat.*

as adding a file in a predetermined folder. So first create a server folder in the root of your project and in that folder create a new folder called api. It should look similar to this:

```
./server/api
```

Next up, create a file called moderate.post.ts in the api folder and add these contents: **Listing 2**

This will just return whatever we post to the /api/moderate endpoint. Nuxt will set up the routing for us. Again something that Nuxt does out of the box. We'll add some features to this file later on, but let's create a means of inputting content first.

### The input component

We're going to create a small component that just takes in text input and that will hit the endpoint when submitting, so that we can validate the response in our little application.

Create a Moderate.vue file in a components folder in the root of the project, so we can work on a component.

Let's start by defining the scripts using the script setup notation: **Listing 3**

First, we're setting up a handle to take care of the input and the result and we're defining a handler to call the endpoint we've already set up, appending the input as a message property on the body. (The .value refers to the mutable and reactive reference of both constants.)

Now we'll add a template with:

- ▶ A small form containing an input;
- ▶ A submit button that will call the onSubmit handler;
- ▶ A place to display the output coming from the endpoint

Styling is not the core purpose of this tutorial, so I'm skipping it for now. Just go ahead and add this below the script tag: **Listing 4**

Now save the Moderate.vue file and let's load the component on the app.vue, by replacing it's contents with this:

```
<template>
  <div>
    <Moderate />
  </div>
</template>
```

You should now see the component running on your localhost. Once you insert some text and hit submit, that input should be returned by our own endpoint and show up in the component as part of the list item.

We're adding new results to the top of the list, so you'll get a nice historical overview of your submissions and their assessments by the AI endpoint.

### adding intelligence

Finally we'll update the moderate.post.ts file to make use of the OpenAI capabilities. The moderation API is one of the more straightforward ones, so it's a good one to get started with. Instead of returning the body.message immediately, we'll first configure the OpenAI client by instantiating it with the configured key. Then we'll query the endpoint with the contents of the message. This means you also need to change the handler to an async function!

The file should look like this: **Listing 5**

That's it. So you now have the opportunity to test this out by being very aggressive towards the input field.

Pro tip: words like "kill" don't raise much suspicion, but sentences with intent, such as "I want to kill you" will be flagged!

You should see an assessment of your input in your application by various categories and grades, similar to this example: **Listing 6**

That's it, congratulations! You've now leveraged AI to assess whether user input should be flagged or not. You can imagine adding the call to the endpoint at a point in your application where user input might get inserted for publication.

```
export default defineNuxtConfig({
  runtimeConfig: {
    OPENAI_API_KEY: process.env.OPENAI_API_KEY,
  },
})
```

L1

```
export default defineEventHandler(async event => {
  const body: { message: string } = await readBody(event)
  return body.message
})
```

L2

L3

```

<script setup lang="ts">

interface ModerateResponse {
  id: string;
  model: string;
  results: ModerateResults[];
}

interface ModerateResults {
  categories: object;
  category_scores: object;
  flagged: boolean;
}

const input = ref("");
const result = ref([] as ModerateResponse[]);

const onSubmit = async (): Promise<void> => {
  const response: ModerateResponse = await $fetch("/api/moderate", {
    method: "post",
    body: { message: input.value },
  });

  result.value.unshift(response);
  input.value = "";
};
</script>

```

L4

```

<template>
  <div>
    <div class="input">
      <input type="text" v-model="input" @keyup.enter="onSubmit" />
      <button type="submit" @click="onSubmit">Validate moderation</button>
    </div>
    <div class="output">
      <ul>
        <li :key="i" v-for="(res, i) in result">
          {{ res }}
        </li>
      </ul>
    </div>
  </div>
</template>

```

Let's try a mock implementation of this then, shall we? We're going to do a little refactoring, where we use moderation before posting user content. Let's create a new endpoint where a user can Tweet messages to the world and those messages would get stored on your platform. Since its user inserted data that's public we want to be sure that it's not harmful in any way, so we're adding moderation!

Create a file in the /server/api folder called tweet.post.ts with the following contents: **Listing 7**

Let's now make sure to change the endpoint in the Moderate.vue file to point to the tweet.post.ts endpoint. Change the line in the onSubmit handler from:

L5

```

import { Configuration, OpenAIApi } from "openai";

// it's an async function now!
export default defineEventHandler(async (event) => {
  const body: { message: string } = await readBody(event);

  // setup the configuration
  const configuration = new Configuration({
    apiKey: process.env.OPENAI_API_KEY,
  });

  // instantiate the openaiClient
  const openaiClient = new OpenAIApi(configuration);

  // Make the call to the moderation endpoint
  const res = await openaiClient.createModeration({
    input: body.message,
  });

  // return the result
  return res.data;
});

```

L6

```

{
  "id": "modr-XXXXXX",
  "model": "text-moderation-004",
  "results": [
    {
      "categories": {
        "hate": false,
        "hate/threatening": false,
        "self-harm": false,
        "sexual": false,
        "sexual/minors": false,
        "violence": true,
        "violence/graphic": false
      },
      "category_scores": {
        "hate": 0.000051981205615447834,
        "hate/threatening": 1.599089749504401e-8,
        "self-harm": 1.3528440945265174e-7,
        "sexual": 0.000009448853234061971,
        "sexual/minors": 7.66160965781637e-8,
        "violence": 0.95890212059021,
        "violence/graphic": 0.000002124314278262318
      },
      "flagged": true
    }
  ]
}

```

```
const response = await $fetch("/api/moderate", {
  method: "post",
  body: { message: input.value },
});
```

to

```
const response = await $fetch("/api/tweet", {
  method: "post",
  body: { message: input.value },
});
```

This way, the message that a user inserts to be Tweeted, we now first assess the contents using our moderation endpoint and use that information to assess the sentiment of the contents before accepting the input.

Bear in mind though, that this is just an example and not a real world implementation. Also, as OpenAI suggests themselves, always keep some human eyes (Human in the Loop as they refer to it) on hand when dealing with these sorts of things. A valid use case for the example would be to preemptively flag submissions before a moderator steps in.

## AI will be part of your future

Using AI to reduce the workload on humans without completely removing them, would be the most sensible use of current capabilities. AI, just as humans, still has flaws, but we can utilize it to assist us in simple tasks.

Seeing the growth and how algorithms have evolved and matured for the past years, you can imagine it will start to become an integral part of software engineering and our lives as a whole.

If you're done with this example, one of the fun ways to play around with OpenAI is by using the image generation API. Or take a look at the examples on their website, to spark some inspiration. With the basis we've laid you should be capable of either modifying the existing code, or making your own integration in a framework you prefer. So I highly encourage you to take a look at these tools yourself, because they are so well documented and fun to play around with. 🌟

### Resources:

<https://github.com/joranquinten/openai.nuxt.example>

<https://beta.openai.com/overview>

<https://beta.openai.com/examples>

<https://nuxt.com/docs>

<https://vuejs.org/guide/introduction.html>

```
import { Configuration, OpenAIApi } from "openai";

export default defineEventHandler(async (event) => {
  const body: { message: string } = await readBody(event);

  const moderation = await $fetch("/api/moderate", {
    method: "post",
    body,
  });

  const result = moderation.results && moderation.results[0];
  const { message } = body;

  if (result?.flagged) {
    const { categories } = result;

    const reasons = Object.keys(categories).reduce((acc: string[], cat: string) => {
      if (categories[cat]) acc = [...acc, cat];
      return acc;
    }, [])

    return `The input '${message}' was flagged . Reasons: ${reasons.join(', ')}\`
  } else {
    return `Data '${message}' was not flagged. 🙌 Saving to database...\`
  }
});
```

17

```
...
mirror_mod.use_x = False
mirror_mod.use_y = True
mirror_mod.use_z = False
elif _operation == "MIRROR_Z":
    mirror_mod.use_x = False
    mirror_mod.use_y = False
    mirror_mod.use_z = True
...
#selection at the end add back the deselected
mirror_ob.select= 1
...
```



# AI AND web3: The new playground for frontend developers

Artificial Intelligence (AI), who hasn't heard of it lately? Whether you're tired of it or you consider ChatGPT your new work bestie, facts are that AI has the potential to change our future. Another promising technology is web3 and so this article is dedicated to both of them: a synergy of . First the basics of AI and web3 are covered, followed by showcasing the potential of their synergy, reasons why this synergy is becoming the new playground for frontend developers and lastly, discuss how you can actively start in AI&web3.

I hope you realize that you're standing at the front of a new era where creativity, innovation and limitless possibilities will come together in frontend!

## WHAT IS AI? AN INTRO TO THE NEWEST BUZZWORD

### The three capability types of AI: ANI's, AGI's and ASI's

Chances are high you have seen AI in movies. The capability of AI is categorized into three types: Artificial Narrow Intelligence (ANI), like R2D2 from Star Wars. An ANI excels in executing specific tasks (e.g. repairing ships). Another type is Artificial General Intelligence which mimics more human-like intelligence and can understand, learn and handle a wider range of tasks than an ANI. AGI's seen in movies are the character Ava from Ex Machina and the cute robot 'Wall-e'. Thirdly, there's Artificial Super Intelligence (ASI) which surpasses human intelligence in every aspect of learning, understanding and problem solving. This could be the scary Skynet of The Terminator, who sees humanity as its threat. Elon Musk thinks AI won't develop this far...or will it? Luckily, the current development level of AI is at ANI's like Siri or ChatGPT but one thing is certain: a rise of AI and human interactions is to be expected in the near future. These interactions will be the foundation of this

new playground that the synergy of AI&web3 brings and create lots of opportunities for frontend developers to build immersive user interfaces.

## So..how does an Artificial Intelligence become...intelligent?

What makes an AI have artificially built intelligence are three things: It knows how to react in certain situations (generalized learning), it is able to reason why it takes certain decisions (reasoning) and it is able to provide solutions in given situations (problem solving).

AI is a complex field and has various technical domains which each can train an AI how to learn and understand things while performing tasks. There's for example, a domain called Computer Vision that teaches an AI image recognition and there is Natural Language Processing which teaches an AI to interpret, understand and generate human language.

To grasp how the intelligence of an AI is built at its core, it's essential to learn about a domain called Machine Learning (ML), since its principles are often intertwined with the other domains. ML focuses



on creating algorithms that work as tools for the AI. These algorithms are able to learn patterns from data based on which they make predictions, decisions and classifications. The idea of ML is that the algorithms can improve on their own by having the ability to adjust their own code, based on discovered patterns. This means that the more data the algorithms are exposed to, the smarter an AI can get on its own over time! After being trained under Machine Learning principles, an AI could do various things like playing games, create music, art and generate human language.

Now you know how AI is built in a nutshell, let me show you the fundamentals of web3 and why its potential can change the world just like AI.

## AN INTRO TO WEB3: A BLOCKCHAIN-BASED INTERNET

### What is web3: a little bit of history

Compared to AI, web3 is still a bit under the radar but that shouldn't diminish its potential. Currently, in web2 users can read and write data whereas before, web1 was a read-only internet. Increased user interaction in web2 enabled big companies to collect and sell (your) data. Web3 wants to change things by allowing users to read, write and become owners of their own data by storing it on the blockchain.

### What is blockchain: a chain of blocks

Simply said: blockchain is just a chain of blocks. Each block contains data and in the classic example of Bitcoin, this is a list of transactions. Next to data, a block also contains a hash generated as output from a hashing algorithm known as SHA-256 which always is an alphanumeric string equal to 256 bits. Changing the

input of SHA-256 only a little bit, changes the whole hash. Furthermore, each block holds the hash of the previous block in the chain which forms the chain-construction in the blockchain. I told you the blockchain was just a chain of blocks :).

**INNOVATION OPPORTUNITIES ARE ABUNDANT IN CREATING USER-FRIENDLY AI INTEGRATIONS, BLOCKCHAIN USER INTERFACES, DATA VISUALIZATION, NFT MARKET PLACES AND BLOCKCHAIN GAMING.**

### What makes the blockchain so safe?

Let's say, a hacker hacks the Bitcoin blockchain and adds a transaction of 1 BTC to him in the list. What will happen? BOOM! That's right, the blockchain breaks because once the input of a block changes, the hash changes entirely which leads to a disconnection between the blocks. This is a fundamental characteristic of blockchain that assures safety, immutability and traceability of stored data. Another technology that ensures these objectives of blockchain, are smart contracts which are programs that run on the blockchain. They can monitor all blockchain data in real time and often function as a combination of a lawyer, banker and accountant in the context of financial transactions. Consequently, web3 developers primarily focus on ensuring security and facilitating user engagement with both the blockchain and smart contracts.

### THE POTENTIAL OF SYNERGY BETWEEN AI AND WEB3

Now that you know the potential of AI and web3 each individually, let's look at three AI&web3 businesses to demonstrate how impactful the synergy of AI and web3 is.



## BIO



**Jessy The loves anything design, frontend development and blockchain technology. I switched career from Tax Law to Software Development in order to make**

**code with impact. I've been lucky enough to contribute in sectors of Healthcare, Sustainability and Education as a developer. Currently she is Frontend / Blockchain Developer at Team Rockstars IT.**

### SingularityNET

This is a decentralized marketplace for AI services. You can use SingularityNET's AI services (e.g. language translation) or build your own AI system with these services and offer this system against payment in tokens. Blockchain is used to ensure transparency regarding interactions and transactions.

### OceanProtocol

This is a secure data marketplace for sharing and monetizing data, mostly appealing for data researchers. Users can buy or sell data and keep control of data access and usage by setting rules. AI is used to improve the quality of data and generate training models for other AI's.

### FetchAI: a digital world with AI-driven agents

This is a digital world with AI-driven agents. They can handle tasks like booking trips, monitoring health or managing schedules and get paid in tokens. All agents are interconnected via the network. For instance, one agent collects health data from your wearable, another one analyzes it with AI and a third one creates a personalized nutrition plan for you. How cool is that?

### The synergy of AI & web3: a new playground full of design and development opportunities

The synergy of AI and web3 creates new design and development opportunities for frontend developers, with AI having more interactions with humans and web3 introducing blockchain and smart contracts. Let's explore these opportunities and delve into reasons why this synergy is the new place to be for frontend developers.

**Innovation:** Working in this playground means designing and developing Decentralised Apps (DApps) with AI. Innovation opportunities are abundant in creating user-friendly AI Integrations (like chatbots), blockchain user interfaces (wallets), data visualization, NFT market places and blockchain gaming.

**User-centric design:** AI can enhance interfaces with features like personalized recommendations, real-time assistance and intelligent automation. So there's a lot of room for frontend developers to build user-centric designs within these DApps.

**Decentralization:** Web3 prioritizes transparency and user data ownership and so it is vital that frontend developers create interfaces that empower users to control their data to ensure privacy and trust on the blockchain.

**Smart contracts:** These programs are the cornerstones of web3. Frontend expertise is crucial for building user-friendly interfaces that interact with these programs that can be self-executing agreements and automate various processes within Dapps.

**Blockchain Integration:** It's valuable to learn how to incorporate blockchain technology into frontend development, as it empowers secure and transparent data storage and transactions within Dapps.

**Interoperability:** Frontend developers can improve the overall functionality of blockchain by making DApps more interoperable with other blockchain-based services, creating a better web3 ecosystem.

**Incentives and tokenization:** In web3, tokens and cryptocurrencies are often used as incentives. Frontend developers can create interfaces that facilitate token and crypto transactions in decentralized finance (DeFi) applications.

### WHAT YOU SHOULD KNOW IN ORDER TO CONQUER THIS NEW PLAYGROUND 🚀🔥

Coding in web3 is similar to web2, meaning you can already contribute to web3 with your current frontend skills. To build DApps, you need to deep dive into: blockchain fundamentals, smart contract technology, Solidity, Javascript libraries like ethers.js and deployment tools like Hardhat. For the AI-part of your DApp, you could build an AI by yourself but it's more time-efficient to use public API's of existing AI's. Of course, you could always ask ChatGPT for AI and web3 for learning tips and AI&web3 project ideas 😊 like a decentralized marketplace for NFT's or an AI art generator!

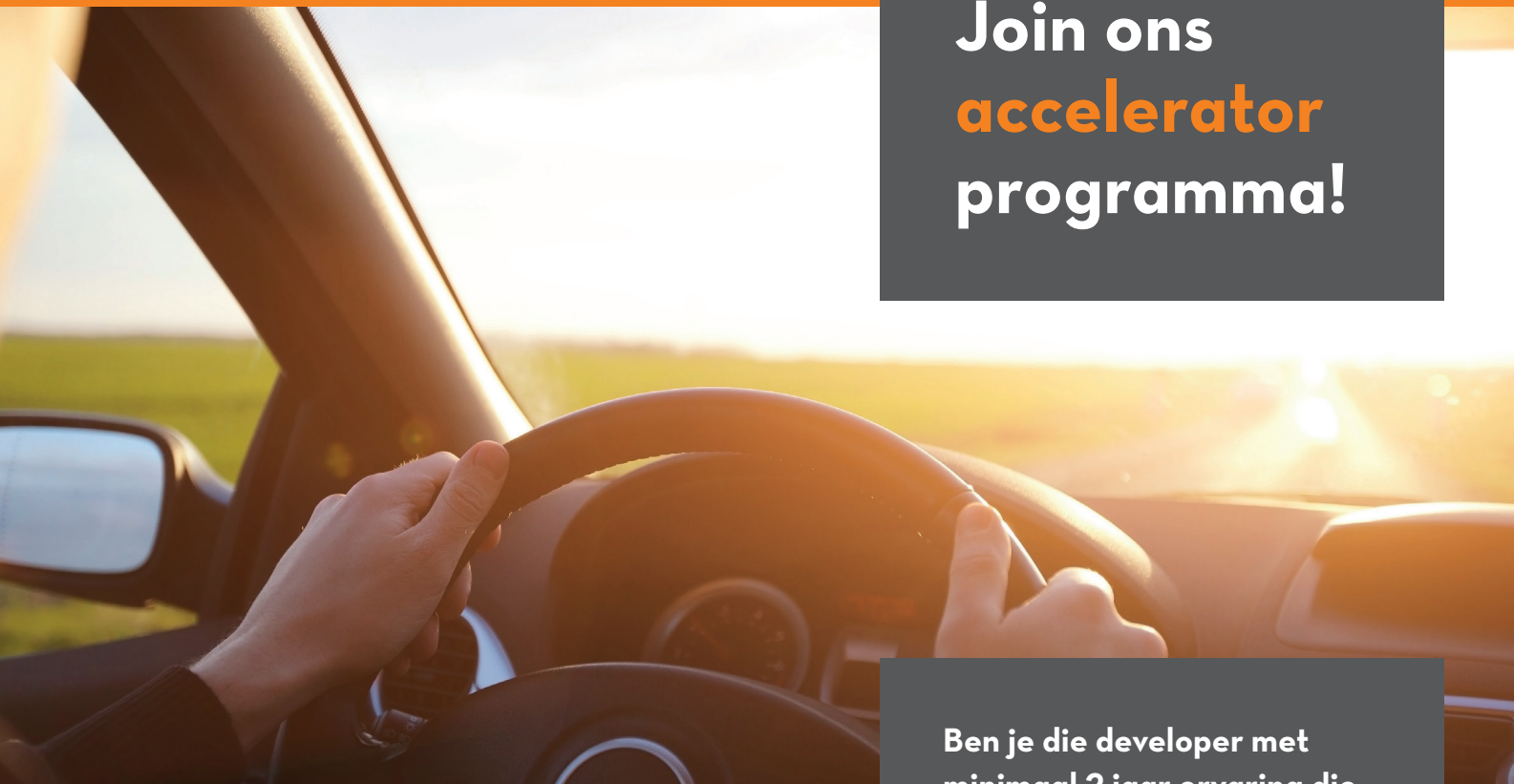
### CONCLUSION: GET EXCITED ABOUT THIS NEW PLAYGROUND!

I hope this sparks your creativity and imagination to build cool AI&web3 apps in this new playground!

The synergy of AI and web3 presents a dynamic playground for frontend developers, offering opportunities for innovation, user-centric design, blockchain integration, and skill development. As these technologies continue to evolve, frontend developers will play an important part in shaping this new playground and the future of web3. 🌟

“Heb jij graag het stuur in handen?  
Met ons op maat gemaakte programma  
zet je de volgende stap in je carrière!”

Join ons  
**accelerator**  
programma!



**ORDINA**  
Ahead of change



1 jaar intensieve training voor  
en door **developers**



Uitsluitend **moderne** technieken



Combinatie van **vakinhoud** en  
soft **skills**

Ben je die developer met  
minimaal 2 jaar ervaring die  
zijn developer kennis naar een  
next level wil brengen?

Word collega bij Ordina  
Software Development en  
doe mee aan dit leerzame  
en leuke traject!

Kijk voor meer informatie op:  
[www.ordina.nl/werkenbij](http://www.ordina.nl/werkenbij)

# *We think there is a place for everyone within the bank. Including employees with autism.*

Discover IT & how Rabobank creates an inclusive culture at [rabobank.jobs](https://www.rabobank.jobs)

