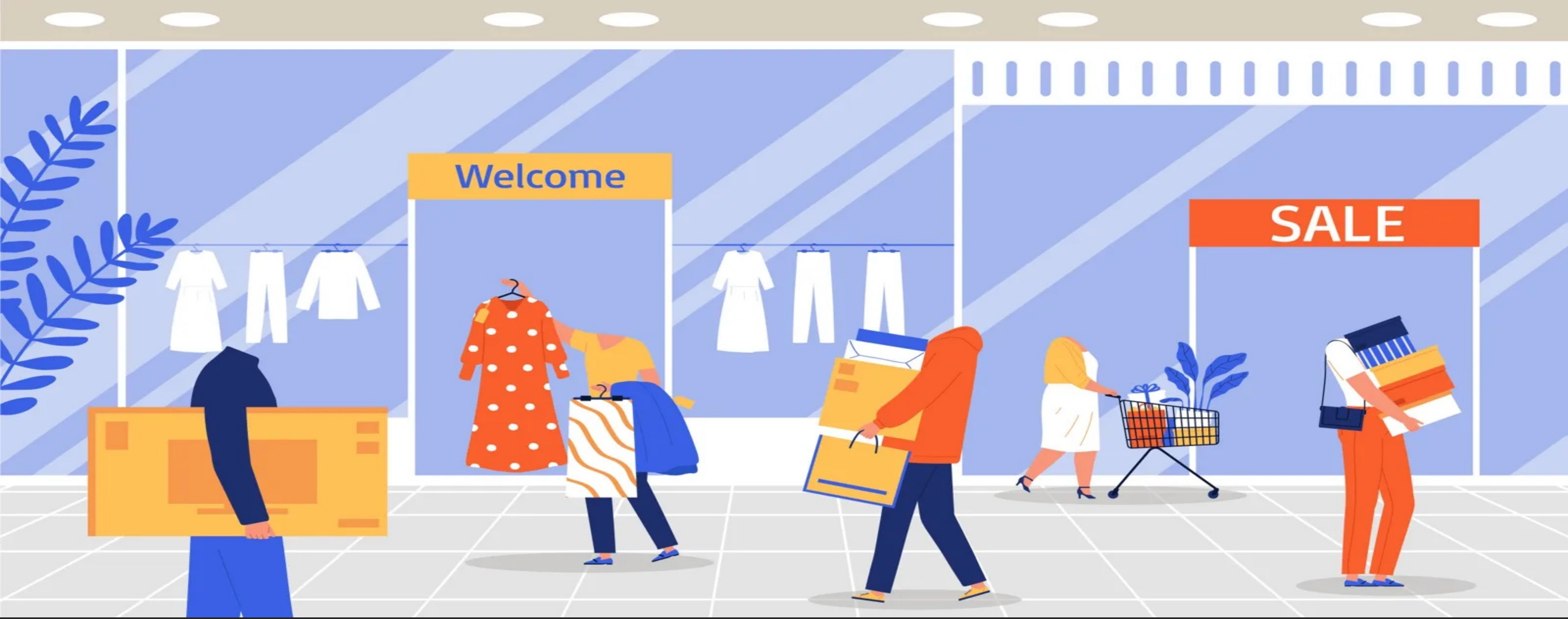


Shopware Boost Day 14.09.2023

Composable frontends in action

+sitegeist

Real Values.

Moin!

- Miriam Müller
- Hamburg
- Senior PHP Developer (focus on backend)
- sitegeist ecommerce solutions
- Shopware since 2015 (4.x)
- mueller@sitegeist.de

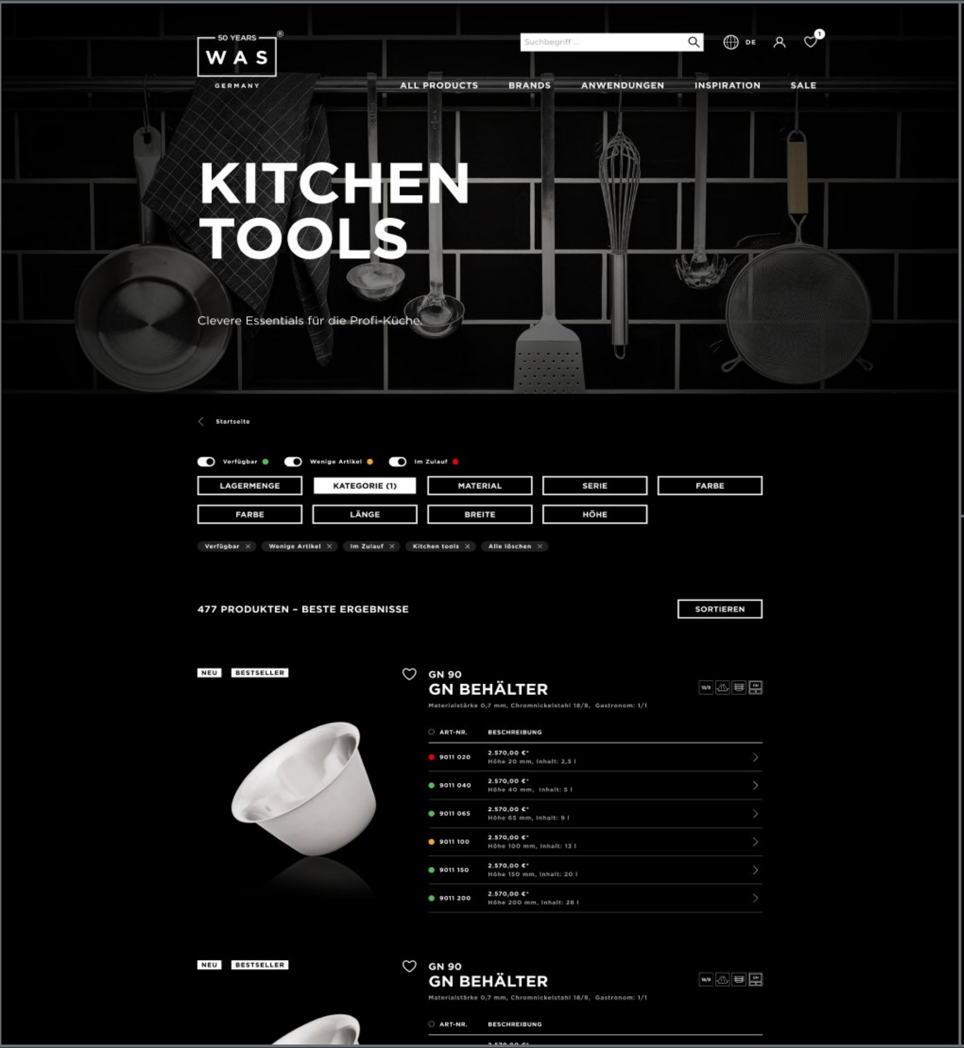
Agenda

- About the project
- Pain points in the past
- Project preparation / design process
- Development process
- DevOps / Hosting
- Learnings
- Conclusion and next steps

About the project

- B2B selfservice portal
- UX/UI kickoff september 2022 (paused march to august 2023)
- The project goal for phase 1 was to display recurring questions about prices, availability and order status online to relieve the customer support. Next phase with order feature
- Individual frontend and strong focus on design details
- Animated header and teaser images and other special features
- Payment not needed for first implementation

Composable frontends in action: About the project



KITCHEN TOOLS

Clevere Essentials für die Profi-Küche.

477 PRODUKTEN – BESTE ERGEBNISSE

GN 90
GN BEHÄLTER

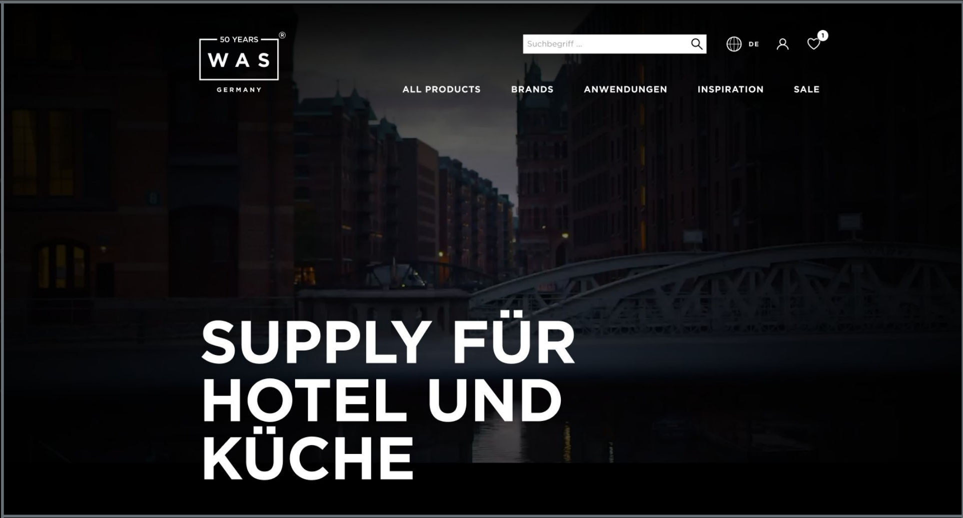
Materialstärke 0,7 mm, Chromnickelstahl 18/8, Gastronom: 1/1

	ART-NR.	BESCHREIBUNG
🔴	9011 020	2.570,00 €* Höhe 20 mm, Inhalt: 2,5 l
🟢	9011 040	2.570,00 €* Höhe 40 mm, Inhalt: 5 l
🟢	9011 065	2.570,00 €* Höhe 65 mm, Inhalt: 9 l
🟠	9011 100	2.570,00 €* Höhe 100 mm, Inhalt: 13 l
🟢	9011 150	2.570,00 €* Höhe 150 mm, Inhalt: 20 l
🟢	9011 200	2.570,00 €* Höhe 200 mm, Inhalt: 28 l

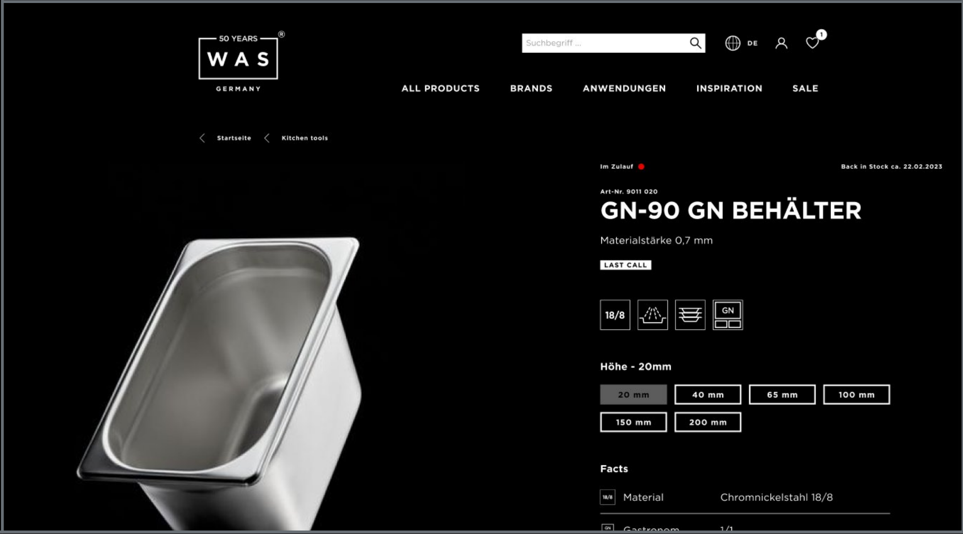
GN 90
GN BEHÄLTER

Materialstärke 0,7 mm, Chromnickelstahl 18/8, Gastronom: 1/1

ART-NR. BESCHREIBUNG



SUPPLY FÜR HOTEL UND KÜCHE



Im Zulauf 🔴 Back in Stock ca. 22.02.2023

Art-Nr. 9011 020

GN-90 GN BEHÄLTER

Materialstärke 0,7 mm

Höhe - 20mm

Facts

Material	Chromnickelstahl 18/8
Gastronom	1/1

+sitegeist

Real Values.

Pain points in the past

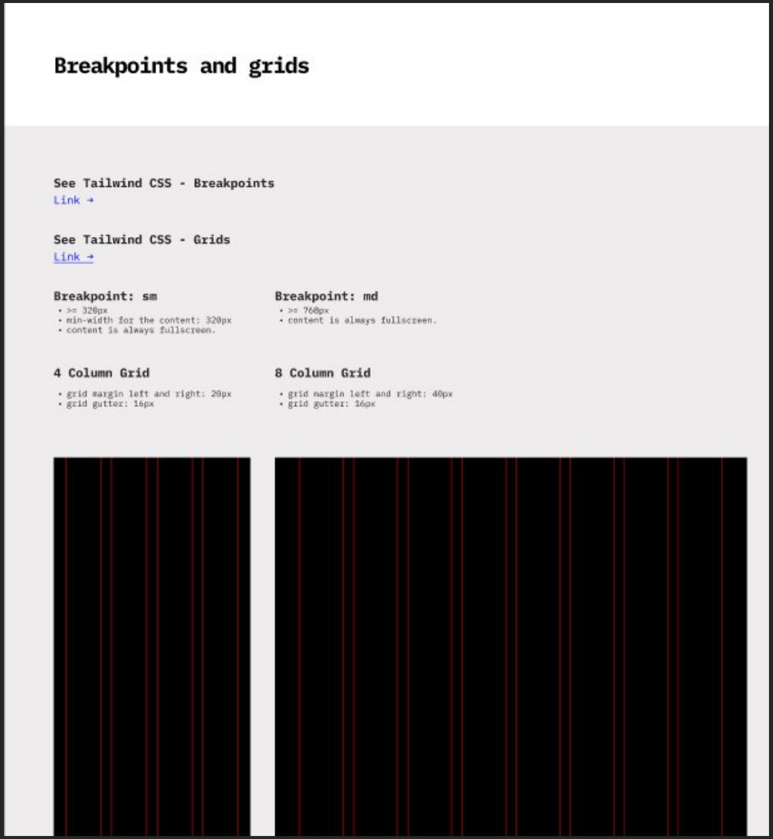
- Adjusting stylesheets in the storefront template becomes increasingly difficult with the amount of customization
- We often end up fixing one thing and creating 5 new errors
- Frustrated frontend developers fighting against the storefront stylesheet
- Dissatisfied UX/UI designers because it was often not implemented as desired
- Unhappy customers because even small adjustments become very expensive over time

Pain points in the past

- Our Neos and TYPO3 teams are working component-based in the frontend already for many years
- Frontend developers preferred to work in the TYPO3 and Neos teams because they could work with more modern technologies

Project preparation / design process

- Clear definition of design tokens like grid, colors, fonts, sizes … at the beginning of the project
- UX/UI team worked with tailwind definitions in figma



Breakpoints and grids

See Tailwind CSS - Breakpoints
Link →

See Tailwind CSS - Grids
Link →

Breakpoint: sm
- >= 320px
- min-width for the content: 320px
- content is always fullscreen.

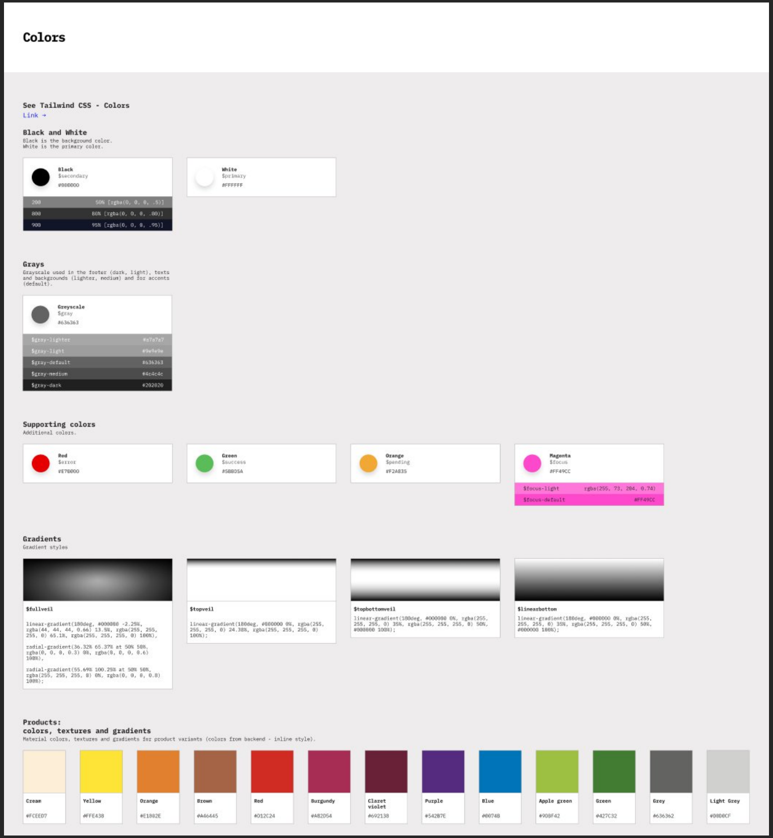
Breakpoint: md
- >= 768px
- content is always fullscreen.

4 Column Grid
- grid margin left and right: 20px
- grid gutter: 16px

8 Column Grid
- grid margin left and right: 40px
- grid gutter: 16px

Project preparation / design process

- Clear definition of design tokens like grid, colors, fonts, sizes … at the beginning of the project
- UX/UI team worked with tailwind definitions in figma



Colors

See Tailwind CSS - Colors
Link →

Black and White
Black is the background color.
White is the primary color.

Black
$secondary
#000000

200	50% [rgba(0, 0, 0, .5)]
800	80% [rgba(0, 0, 0, .80)]
900	95% [rgba(0, 0, 0, .95)]

White
$primary
#FFFFFF

Grays
Grayscale used in the footer (dark, light), texts and backgrounds (lighter, medium) and for accents (default).

Greyscale
$gray
#636363

$gray-lighter	#a7a7a7
$gray-light	#9e9e9e
$gray-default	#636363
$gray-medium	#4c4c4c
$gray-dark	#202020

Supporting colors
Additional colors.

Red
$error
#E70000

Green
$success
#5BBD5A

Orange
$pending
#F2A835

Magenta
$focus
#FF49CC

$focus-light	rgba(255, 73, 204, 0.74)
$focus-default	#FF49CC

Gradients
Gradient styles

$fullveil

```
linear-gradient(180deg, #000000 -2.25%, rgba(44, 44, 44, 0.66) 13.5%, rgba(255, 255, 255, 0) 65.1%, rgba(255, 255, 255, 0) 100%),

radial-gradient(36.32% 65.37% at 50% 50%, rgba(0, 0, 0, 0.3) 0%, rgba(0, 0, 0, 0.6) 100%),

radial-gradient(55.69% 100.25% at 50% 50%, rgba(255, 255, 255, 0) 0%, rgba(0, 0, 0, 0.8) 100%);
```

$topveil

```
linear-gradient(180deg, #000000 0%, rgba(255, 255, 255, 0) 24.38%, rgba(255, 255, 255, 0) 100%);
```

$topbottomveil

```
linear-gradient(180deg, #000000 0%, rgba(255, 255, 255, 0) 35%, rgba(255, 255, 255, 0) 50%, #000000 100%);
```

$linearbottom

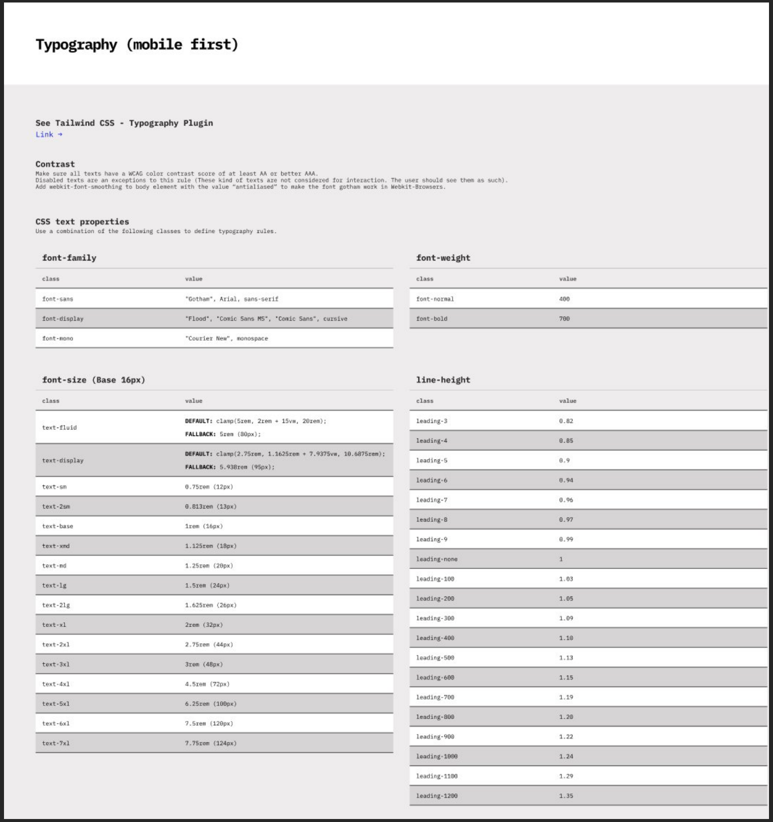
```
linear-gradient(180deg, #000000 0%, rgba(255, 255, 255, 0) 35%, rgba(255, 255, 255, 0) 50%, #000000 100%);
```

Products:
colors, textures and gradients
Material colors, textures and gradients for product variants (colors from backend - inline style).

Cream	#FCEED7
Yellow	#FFE438
Orange	#E1802E
Brown	#A46445
Red	#D12C24
Burgundy	#A82D54
Claret violet	#692138
Purple	#542B7E
Blue	#0074B
Apple green	#9DBF42
Green	#427C32
Grey	#636362
Light Grey	#D0D0CF

Project preparation / design process

- Clear definition of design tokens like grid, colors, fonts, sizes … at the beginning of the project
- UX/UI team worked with tailwind definitions in figma



Typography (mobile first)

See Tailwind CSS - Typography Plugin
Link →

Contrast
Make sure all texts have a WCAG color contrast score of at least AA or better AAA.
Disabled texts are an exceptions to this rule (These kind of texts are not considered for interaction. The user should see them as such).
Add webkit-font-smoothing to body element with the value "antialiased" to make the font gotham work in Webkit-Browsers.

CSS text properties
Use a combination of the following classes to define typography rules.

font-family

class	value
font-sans	"Gotham", Arial, sans-serif
font-display	"Flood", "Comic Sans MS", "Comic Sans", cursive
font-mono	"Courier New", monospace

font-weight

class	value
font-normal	400
font-bold	700

font-size (Base 16px)

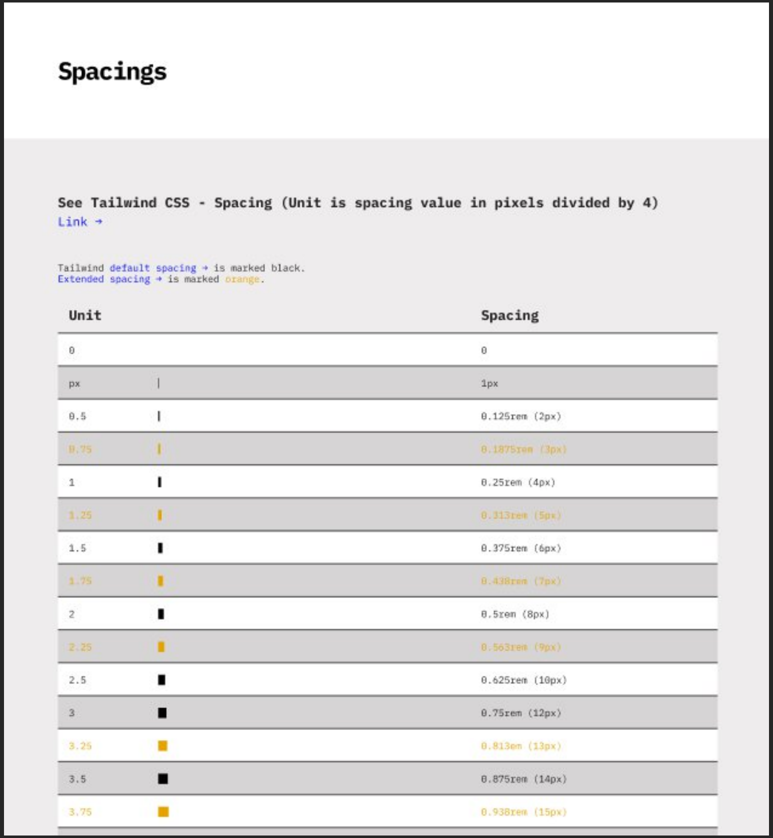
class	value
text-fluid	**DEFAULT:** clamp(5rem, 2rem + 15vw, 20rem); **FALLBACK:** 5rem (80px);
text-display	**DEFAULT:** clamp(2.75rem, 1.1625rem + 7.9375vw, 10.6875rem); **FALLBACK:** 5.938rem (95px);
text-sm	0.75rem (12px)
text-2sm	0.813rem (13px)
text-base	1rem (16px)
text-xmd	1.125rem (18px)
text-md	1.25rem (20px)
text-lg	1.5rem (24px)
text-2lg	1.625rem (26px)
text-xl	2rem (32px)
text-2xl	2.75rem (44px)
text-3xl	3rem (48px)
text-4xl	4.5rem (72px)
text-5xl	6.25rem (100px)
text-6xl	7.5rem (120px)
text-7xl	7.75rem (124px)

line-height

class	value
leading-3	0.82
leading-4	0.85
leading-5	0.9
leading-6	0.94
leading-7	0.96
leading-8	0.97
leading-9	0.99
leading-none	1
leading-100	1.03
leading-200	1.05
leading-300	1.09
leading-400	1.10
leading-500	1.13
leading-600	1.15
leading-700	1.19
leading-800	1.20
leading-900	1.22
leading-1000	1.24
leading-1100	1.29
leading-1200	1.35

Project preparation / design process

- Clear definition of design tokens like grid, colors, fonts, sizes … at the beginning of the project
- UX/UI team worked with tailwind definitions in figma
- additional classes different to the tailwind default are highlighted



Spacings

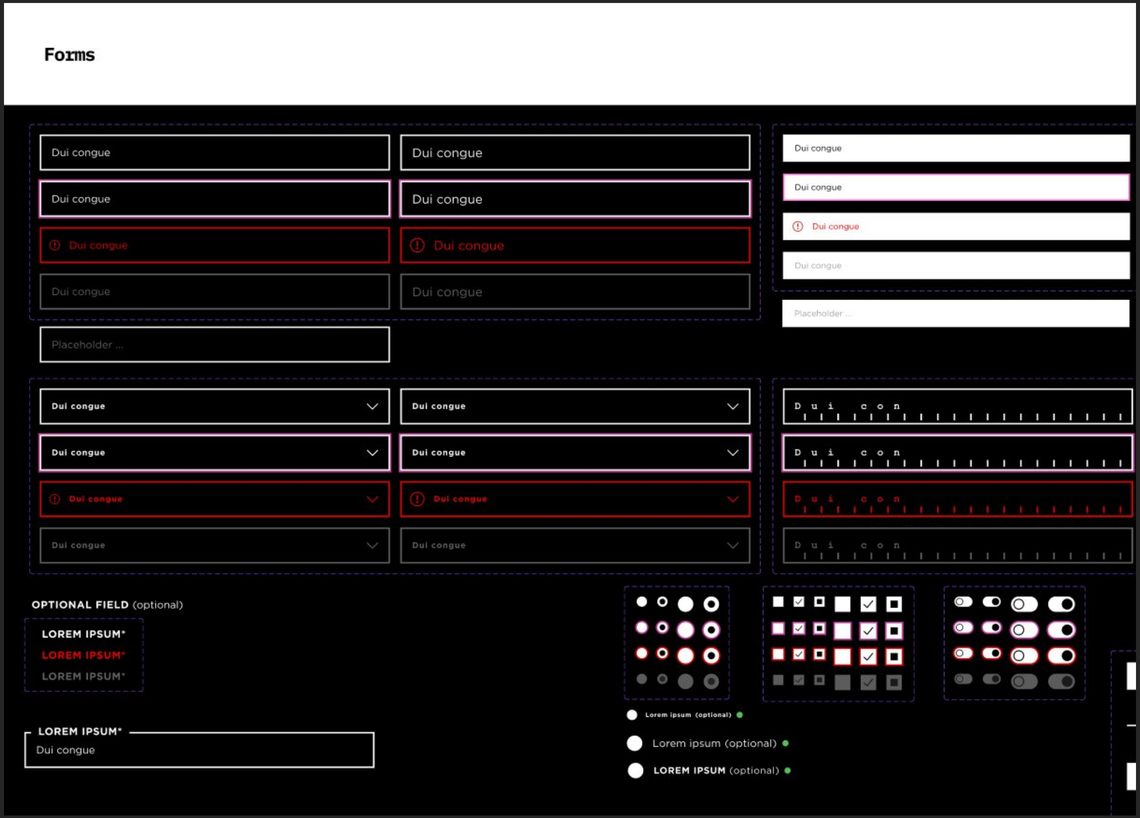
See Tailwind CSS - Spacing (Unit is spacing value in pixels divided by 4)
Link →

Tailwind default spacing → is marked black.
Extended spacing → is marked orange.

Unit		Spacing
0		0
px	▮	1px
0.5	▮	0.125rem (2px)
0.75	▮	0.1875rem (3px)
1	▮	0.25rem (4px)
1.25	▮	0.313rem (5px)
1.5	▮	0.375rem (6px)
1.75	▮	0.438rem (7px)
2	▮	0.5rem (8px)
2.25	▮	0.563rem (9px)
2.5	▮	0.625rem (10px)
3	■	0.75rem (12px)
3.25	■	0.813em (13px)
3.5	■	0.875rem (14px)
3.75	■	0.938rem (15px)

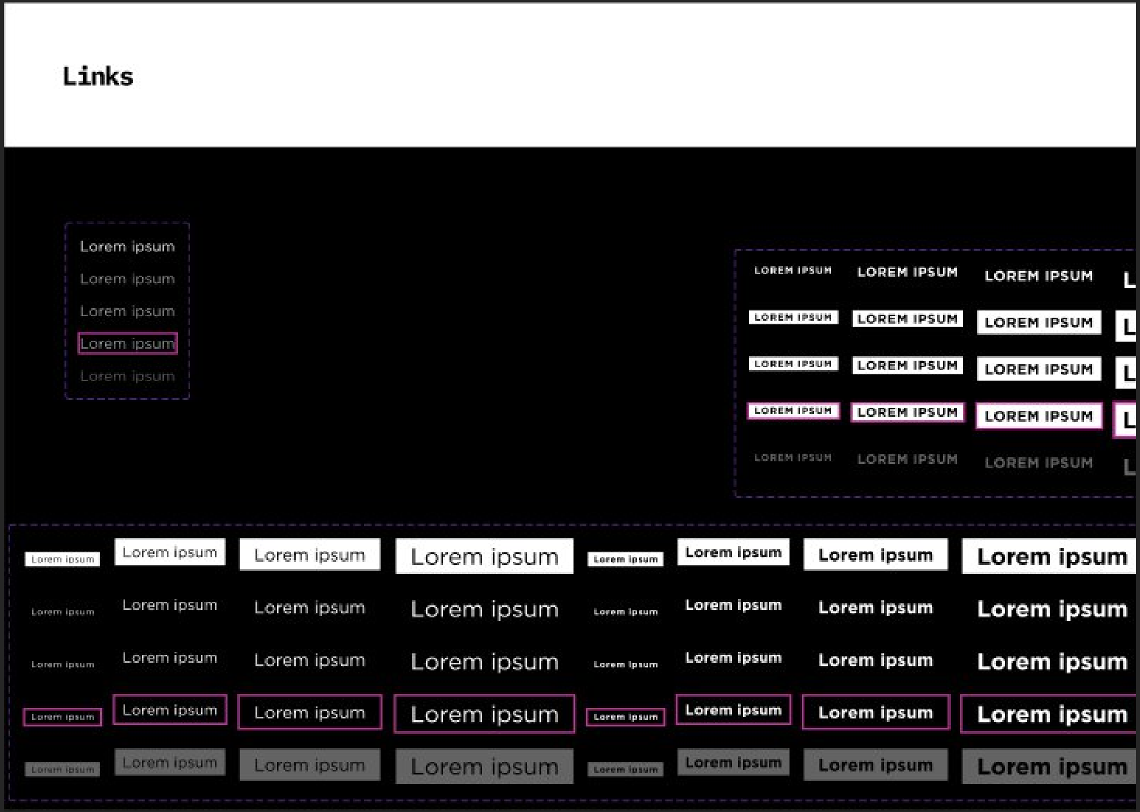
Project preparation / design process

- Components first!
- Started with small components like button, links, inputs …



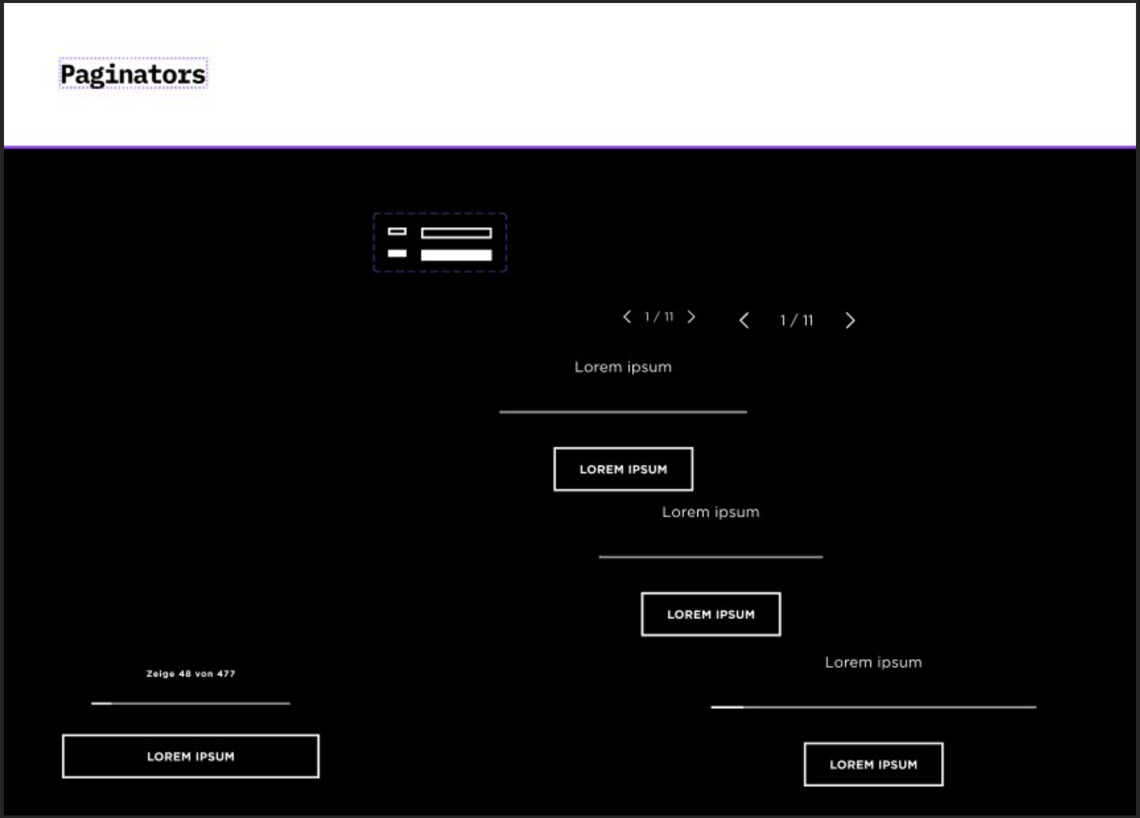
Project preparation / design process

- Components first!
- Started with small components like button, links, inputs ...



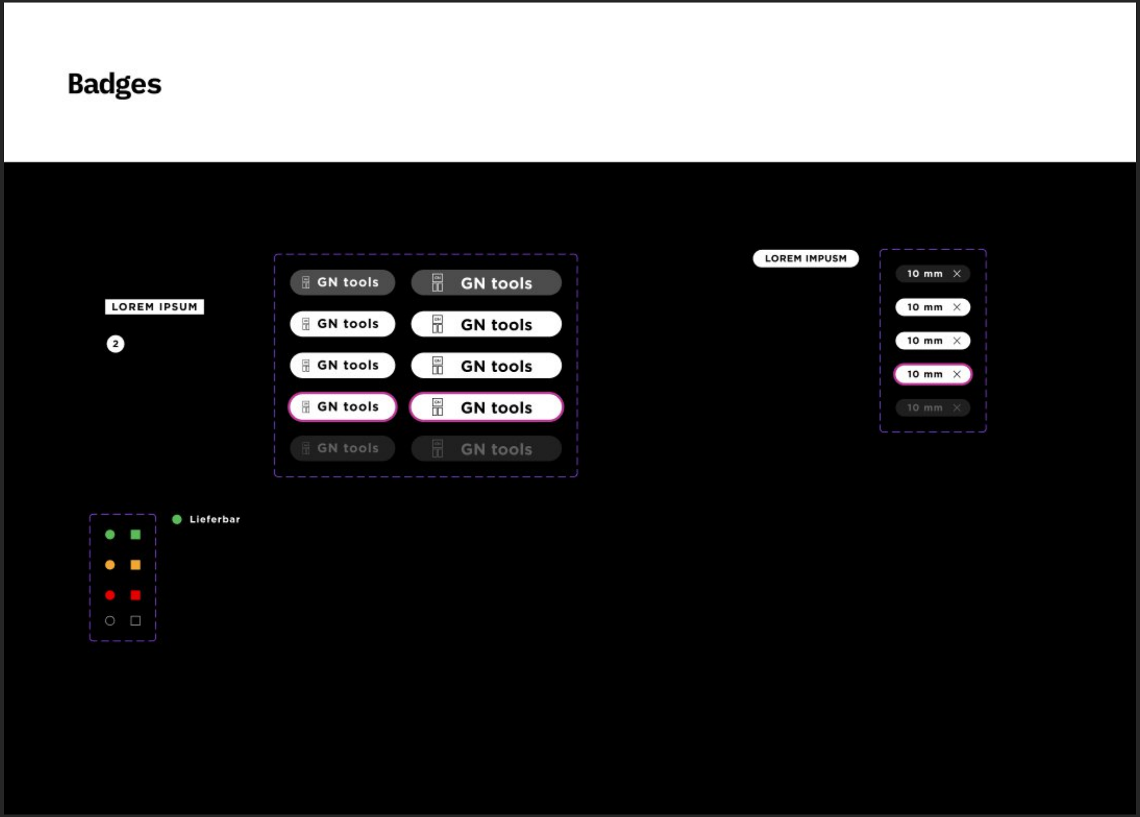
Project preparation / design process

- Components first!
- Started with small components like button, links, inputs ...



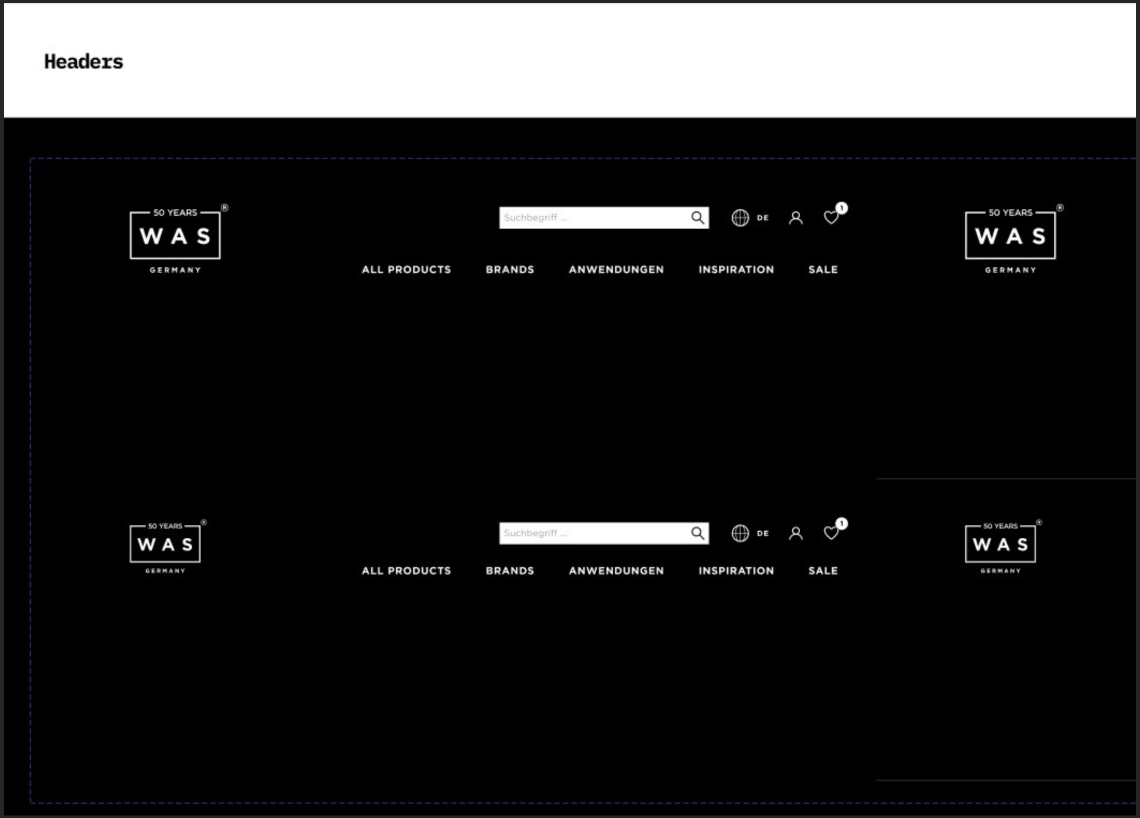
Project preparation / design process

- Components first!
- Started with small components like button, links, inputs ...



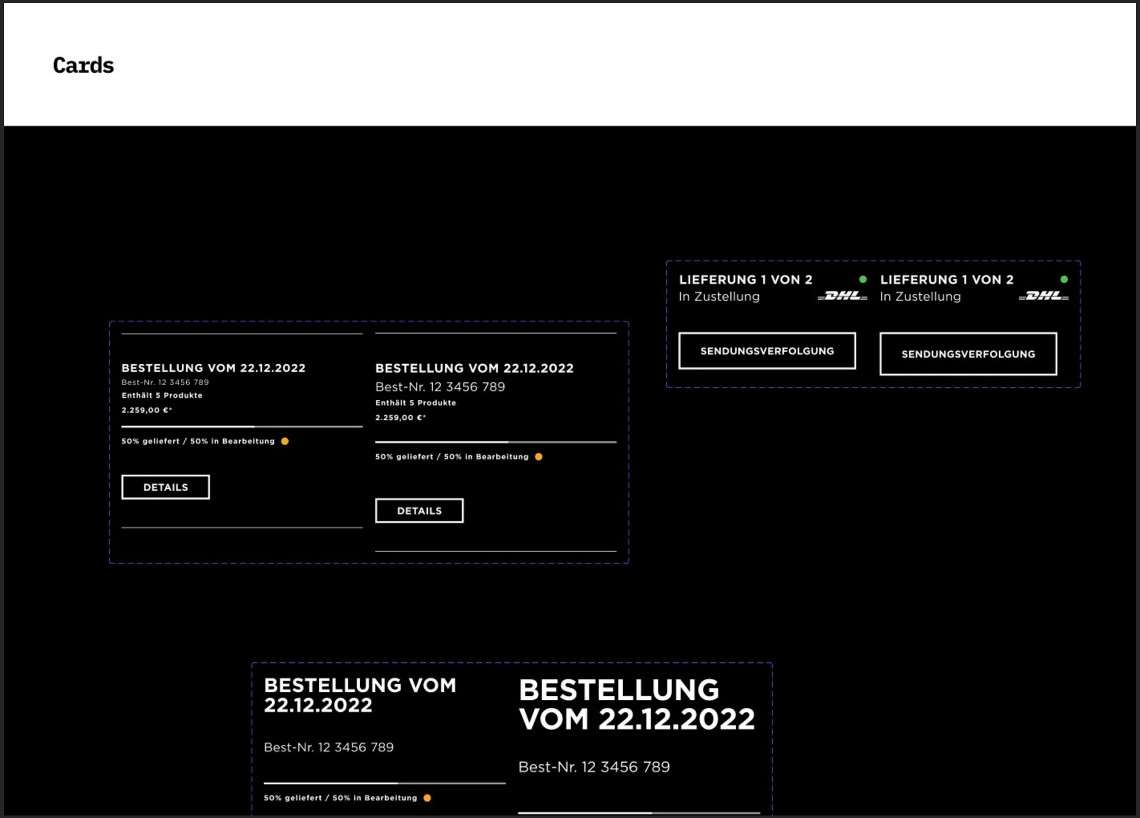
Project preparation / design process

- Components first!
- Grouped smaller components together to bigger components like header, footer, cards …



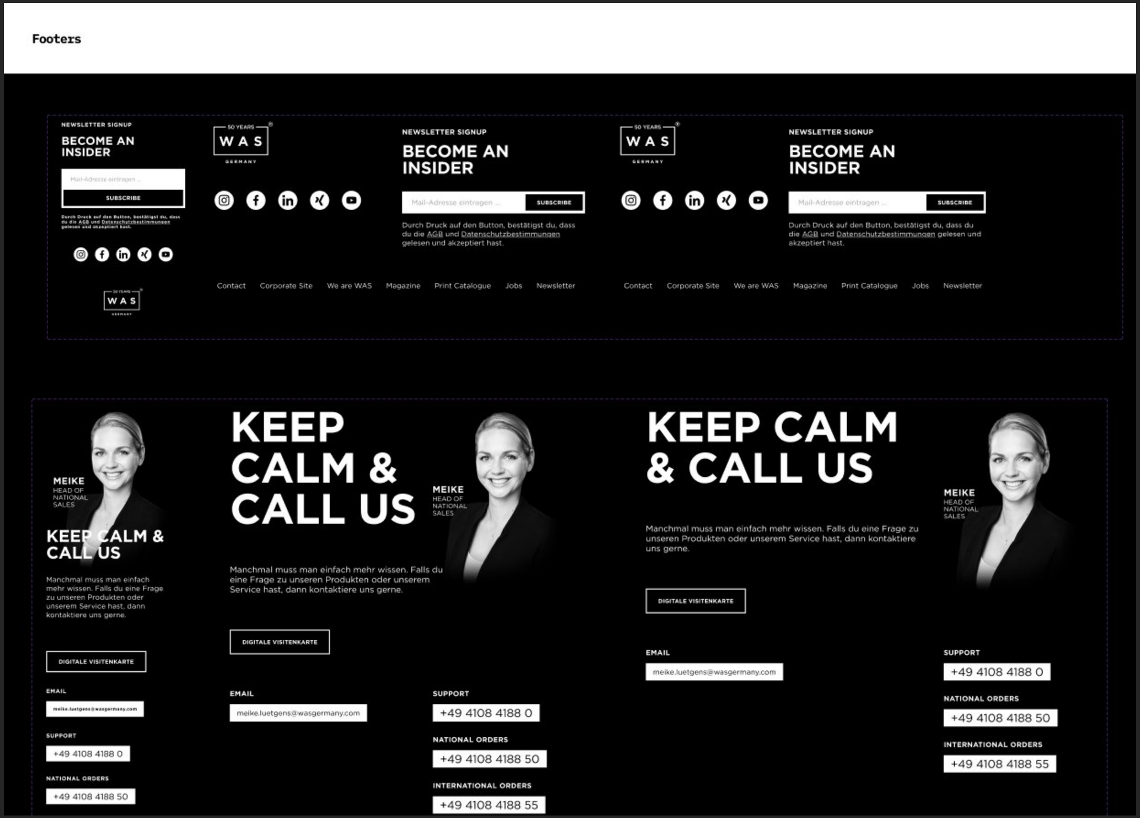
Project preparation / design process

- Components first!
- Grouped smaller components together to bigger components like header, footer, cards ...



Project preparation / design process

- Components first!
- Grouped smaller components together to bigger components like header, footer, cards …

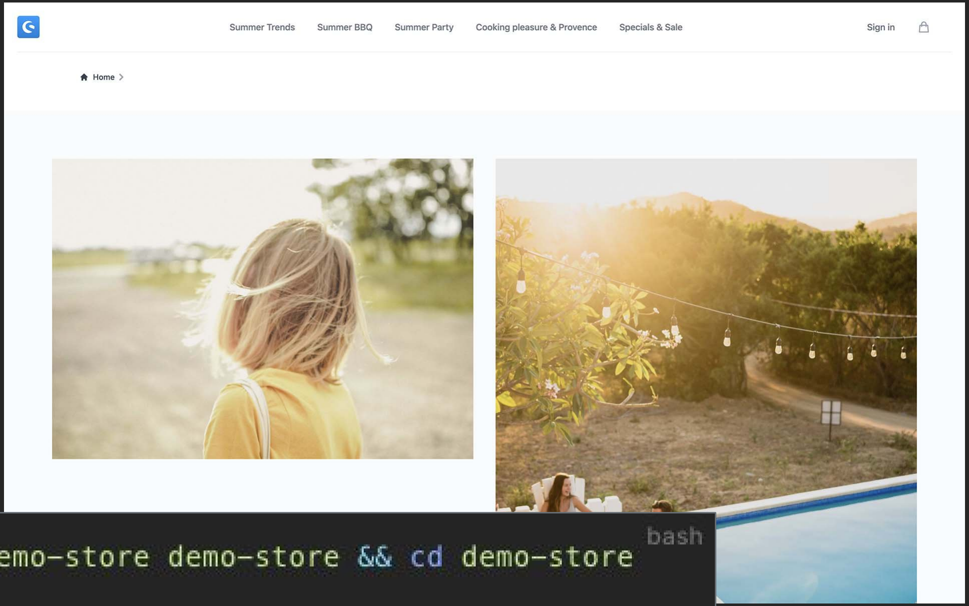


Durch Druck auf den Button, bestätigst du, dass du die AGB und Datenschutzbestimmungen gelesen und akzeptiert hast.

Manchmal muss man einfach mehr wissen. Falls du eine Frage zu unseren Produkten oder unserem Service hast, dann kontaktiere uns gerne.

Development process

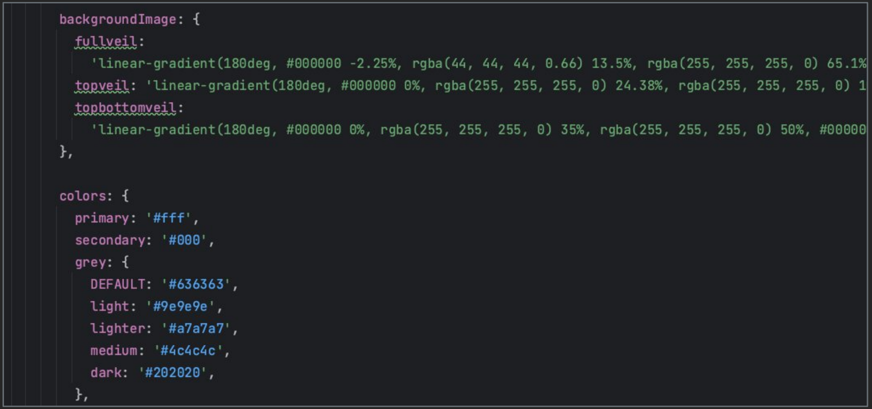
- Project setup with shopware composable frontends demo-store template



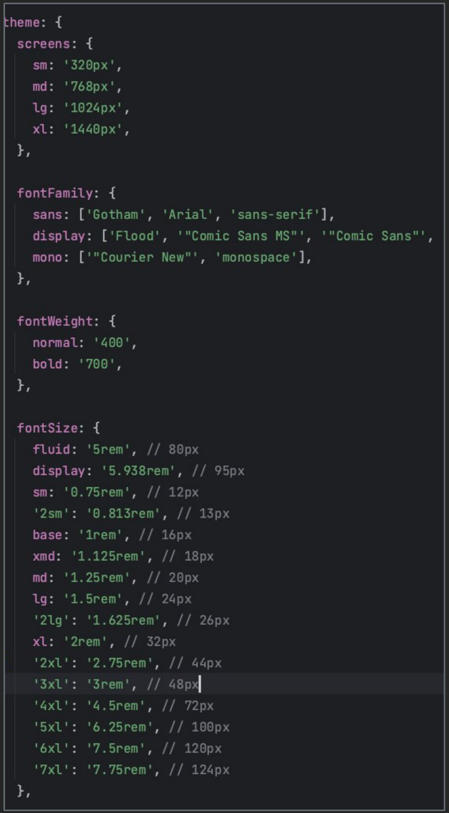
```bash
npx tiged shopware/frontends/templates/vue-demo-store demo-store && cd demo-store
npm i && npm run dev
```

Development process

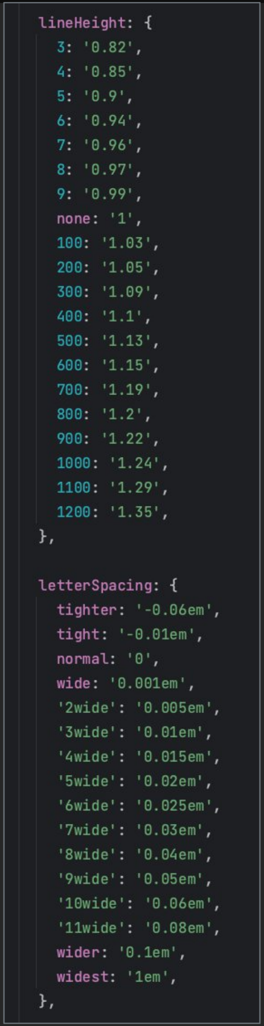
- Tailwind configuration with defined design tokens (tailwind.config.ts)
- UX/UI and frontend developer worked close together
- We took extra time to carefully define the design tokens



```
backgroundImage: {
  fullveil:
    'linear-gradient(180deg, #000000 -2.25%, rgba(44, 44, 44, 0.66) 13.5%, rgba(255, 255, 255, 0) 65.1%
  topveil: 'linear-gradient(180deg, #000000 0%, rgba(255, 255, 255, 0) 24.38%, rgba(255, 255, 255, 0) 1
  topbottomveil:
    'linear-gradient(180deg, #000000 0%, rgba(255, 255, 255, 0) 35%, rgba(255, 255, 255, 0) 50%, #00000
},

colors: {
  primary: '#fff',
  secondary: '#000',
  grey: {
    DEFAULT: '#636363',
    light: '#9e9e9e',
    lighter: '#a7a7a7',
    medium: '#4c4c4c',
    dark: '#202020',
  },
```



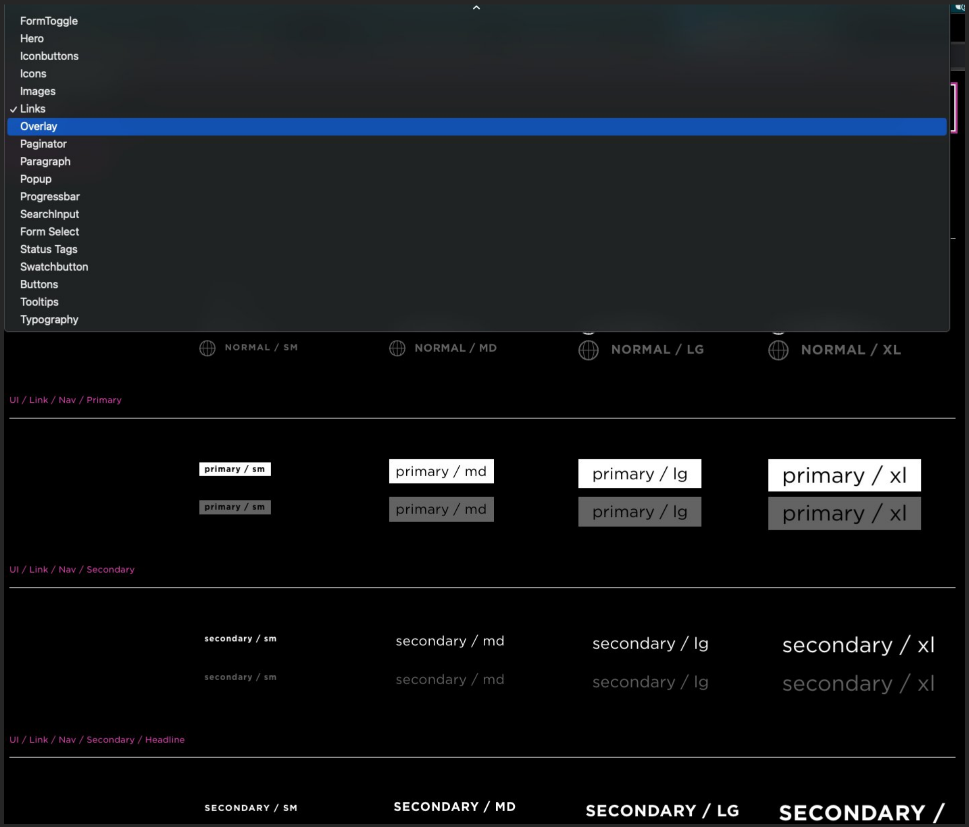
```
theme: {
  screens: {
    sm: '320px',
    md: '768px',
    lg: '1024px',
    xl: '1440px',
  },

  fontFamily: {
    sans: ['Gotham', 'Arial', 'sans-serif'],
    display: ['Flood', '"Comic Sans MS"', '"Comic Sans"',
    mono: ['"Courier New"', 'monospace'],
  },

  fontWeight: {
    normal: '400',
    bold: '700',
  },

  fontSize: {
    fluid: '5rem', // 80px
    display: '5.938rem', // 95px
    sm: '0.75rem', // 12px
    '2sm': '0.813rem', // 13px
    base: '1rem', // 16px
    xmd: '1.125rem', // 18px
    md: '1.25rem', // 20px
    lg: '1.5rem', // 24px
    '2lg': '1.625rem', // 26px
    xl: '2rem', // 32px
    '2xl': '2.75rem', // 44px
    '3xl': '3rem', // 48px
    '4xl': '4.5rem', // 72px
    '5xl': '6.25rem', // 100px
    '6xl': '7.5rem', // 120px
    '7xl': '7.75rem', // 124px
  },
```



```
lineHeight: {
  3: '0.82',
  4: '0.85',
  5: '0.9',
  6: '0.94',
  7: '0.96',
  8: '0.97',
  9: '0.99',
  none: '1',
  100: '1.03',
  200: '1.05',
  300: '1.09',
  400: '1.1',
  500: '1.13',
  600: '1.15',
  700: '1.19',
  800: '1.2',
  900: '1.22',
  1000: '1.24',
  1100: '1.29',
  1200: '1.35',
},

letterSpacing: {
  tighter: '-0.06em',
  tight: '-0.01em',
  normal: '0',
  wide: '0.001em',
  '2wide': '0.005em',
  '3wide': '0.01em',
  '4wide': '0.015em',
  '5wide': '0.02em',
  '6wide': '0.025em',
  '7wide': '0.03em',
  '8wide': '0.04em',
  '9wide': '0.05em',
  '10wide': '0.06em',
  '11wide': '0.08em',
  wider: '0.1em',
  widest: '1em',
},
```

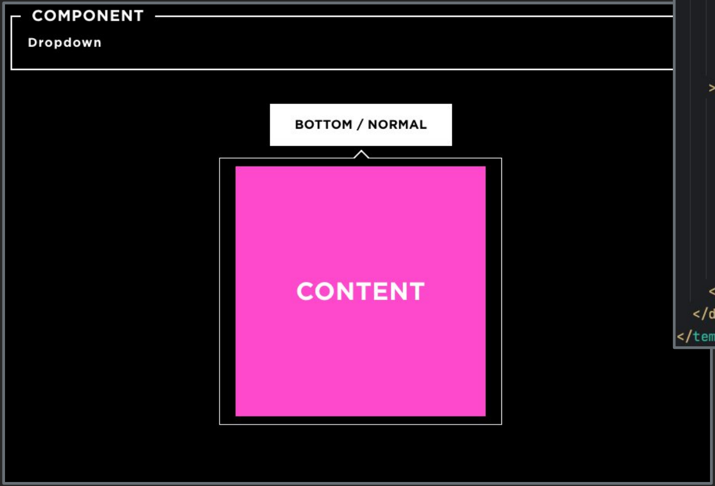
Development process

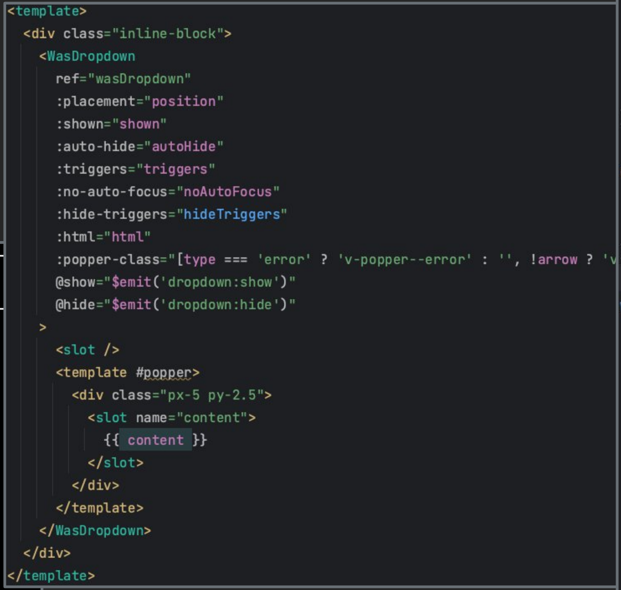
- Setup simple component styleguide as vue.js page
- Storybook was not vue.js 3 ready in the beginning



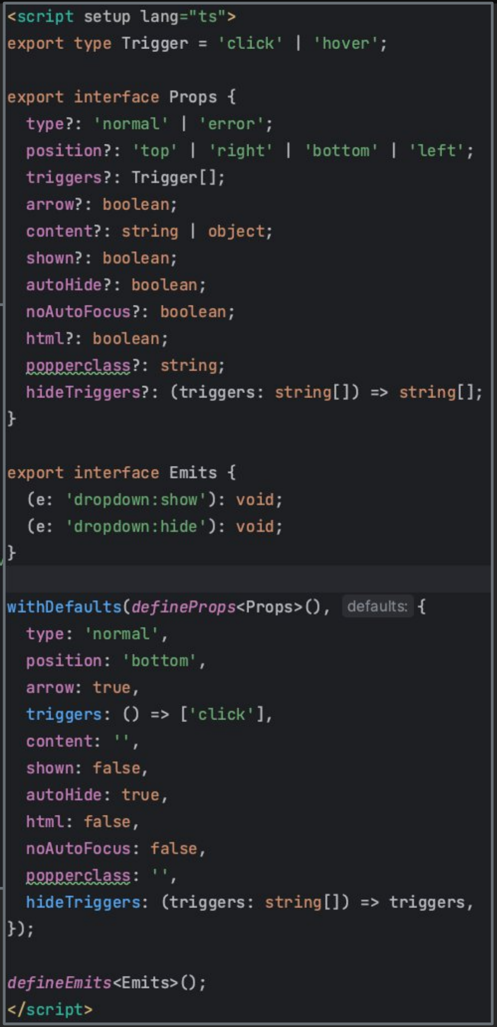
Development process

- Component: Dropdown





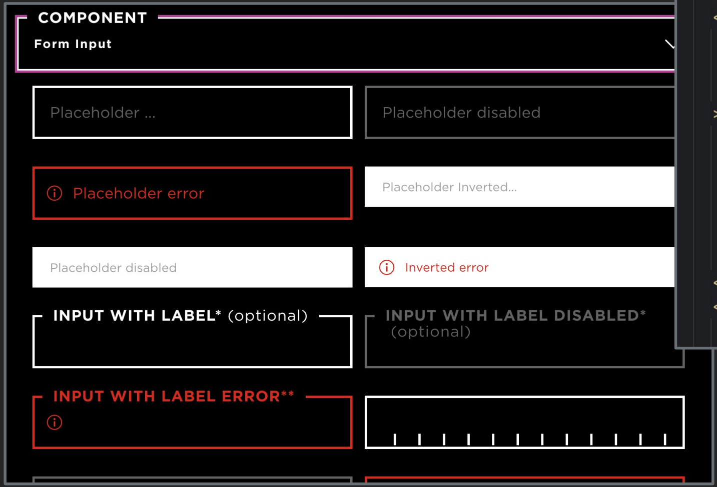
```
<template>
  <div class="inline-block">
    <WasDropdown
      ref="wasDropdown"
      :placement="position"
      :shown="shown"
      :auto-hide="autoHide"
      :triggers="triggers"
      :no-auto-focus="noAutoFocus"
      :hide-triggers="hideTriggers"
      :html="html"
      :popper-class="[type === 'error' ? 'v-popper--error' : '', !arrow ? 'v
      @show="$emit('dropdown:show')"
      @hide="$emit('dropdown:hide')"
    >
      <slot />
      <template #popper>
        <div class="px-5 py-2.5">
          <slot name="content">
            {{ content }}
          </slot>
        </div>
      </template>
    </WasDropdown>
  </div>
</template>
```

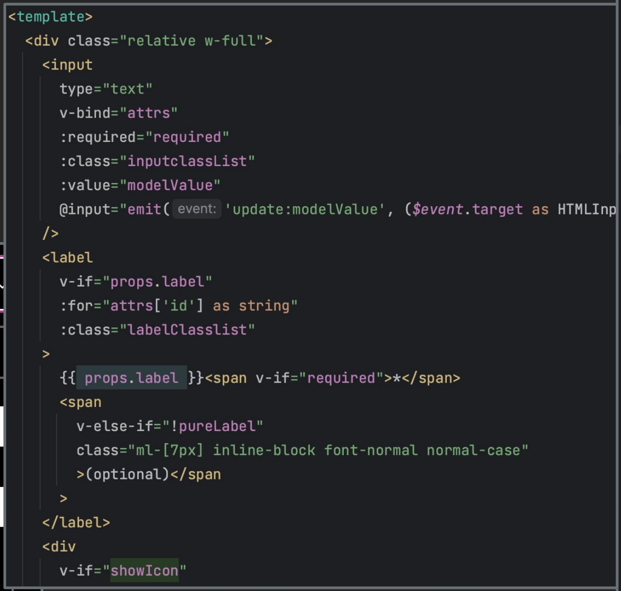


```
<script setup lang="ts">
export type Trigger = 'click' | 'hover';

export interface Props {
  type?: 'normal' | 'error';
  position?: 'top' | 'right' | 'bottom' | 'left';
  triggers?: Trigger[];
  arrow?: boolean;
  content?: string | object;
  shown?: boolean;
  autoHide?: boolean;
  noAutoFocus?: boolean;
  html?: boolean;
  popperclass?: string;
  hideTriggers?: (triggers: string[]) => string[];
}

export interface Emits {
  (e: 'dropdown:show'): void;
  (e: 'dropdown:hide'): void;
}

withDefaults(defineProps<Props>(), defaults: {
  type: 'normal',
  position: 'bottom',
  arrow: true,
  triggers: () => ['click'],
  content: '',
  shown: false,
  autoHide: true,
  html: false,
  noAutoFocus: false,
  popperclass: '',
  hideTriggers: (triggers: string[]) => triggers,
});

defineEmits<Emits>();
</script>
```

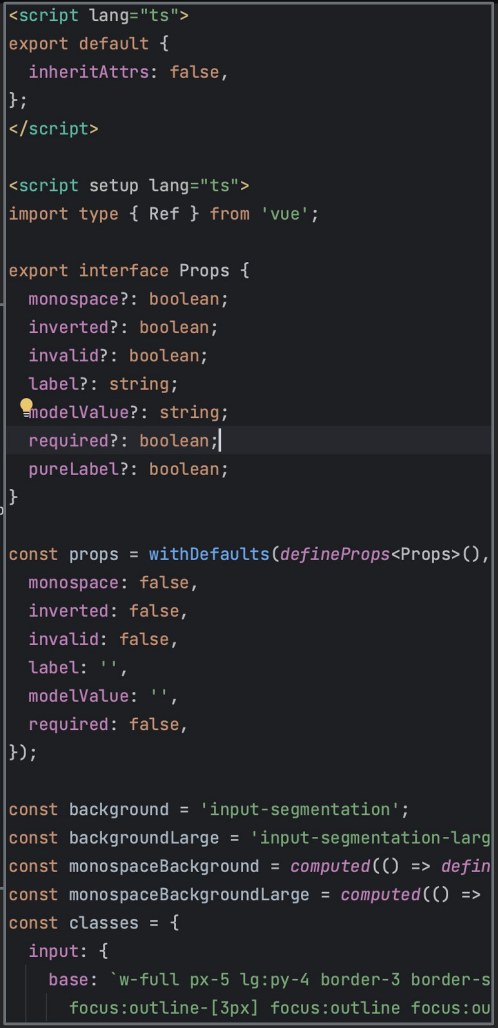
Development process

- Component: Input





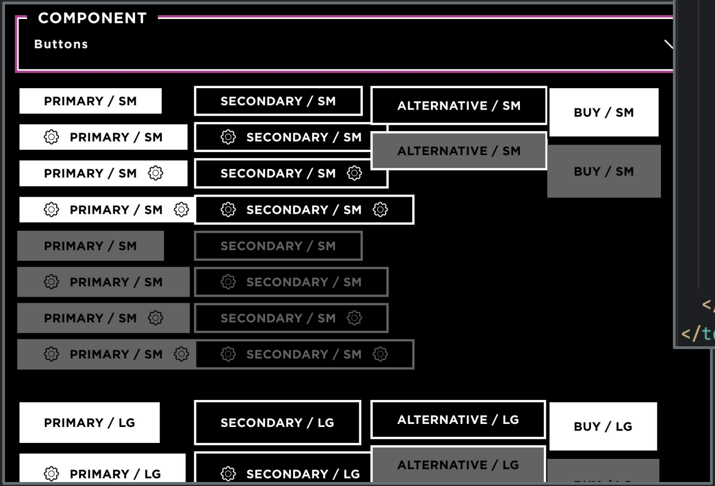
```
<template>
  <div class="relative w-full">
    <input
      type="text"
      v-bind="attrs"
      :required="required"
      :class="inputclassList"
      :value="modelValue"
      @input="emit( event: 'update:modelValue', ($event.target as HTMLInp
    />
    <label
      v-if="props.label"
      :for="attrs['id'] as string"
      :class="labelClasslist"
    >
      {{ props.label }}<span v-if="required">*</span>
      <span
        v-else-if="!pureLabel"
        class="ml-[7px] inline-block font-normal normal-case"
        >(optional)</span
      >
    </label>
    <div
      v-if="showIcon"
```

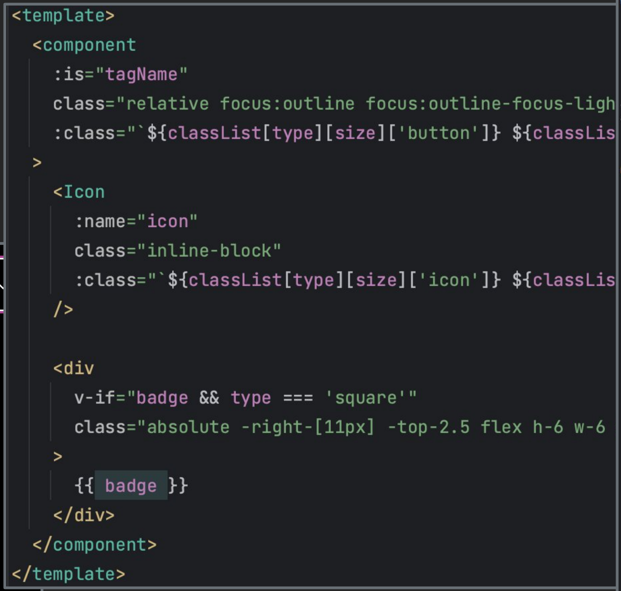


```
<script lang="ts">
export default {
  inheritAttrs: false,
};
</script>

<script setup lang="ts">
import type { Ref } from 'vue';

export interface Props {
  monospace?: boolean;
  inverted?: boolean;
  invalid?: boolean;
  label?: string;
  modelValue?: string;
  required?: boolean;
  pureLabel?: boolean;
}

const props = withDefaults(defineProps<Props>(),
  monospace: false,
  inverted: false,
  invalid: false,
  label: '',
  modelValue: '',
  required: false,
});

const background = 'input-segmentation';
const backgroundLarge = 'input-segmentation-larg
const monospaceBackground = computed(() => defin
const monospaceBackgroundLarge = computed(() =>
const classes = {
  input: {
    base: `w-full px-5 lg:py-4 border-3 border-s
      focus:outline-[3px] focus:outline focus:ou
```

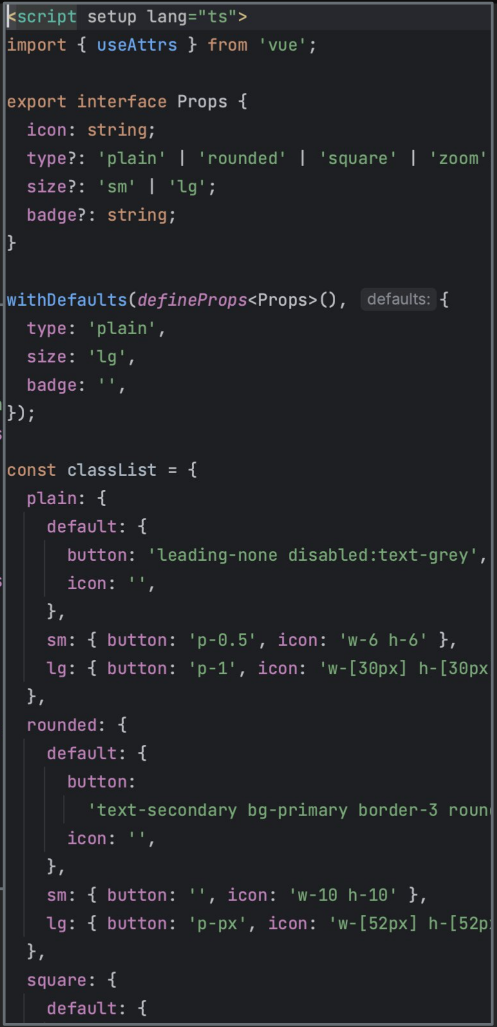
Development process

- Component: Button





```
<template>
  <component
    :is="tagName"
    class="relative focus:outline focus:outline-focus-ligh
    :class="`${classList[type][size]['button']} ${classLis
  >
    <Icon
      :name="icon"
      class="inline-block"
      :class="`${classList[type][size]['icon']} ${classLis
    />

    <div
      v-if="badge && type === 'square'"
      class="absolute -right-[11px] -top-2.5 flex h-6 w-6
    >
      {{ badge }}
    </div>
  </component>
</template>
```



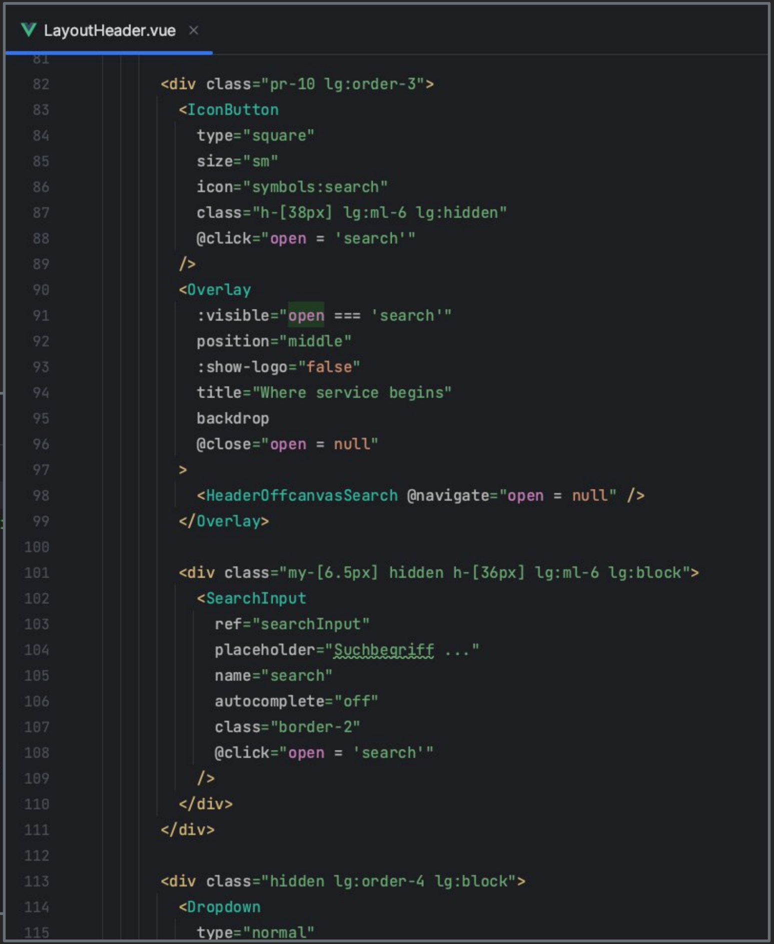
```
<script setup lang="ts">
import { useAttrs } from 'vue';

export interface Props {
  icon: string;
  type?: 'plain' | 'rounded' | 'square' | 'zoom'
  size?: 'sm' | 'lg';
  badge?: string;
}

withDefaults(defineProps<Props>(), defaults: {
  type: 'plain',
  size: 'lg',
  badge: '',
});

const classList = {
  plain: {
    default: {
      button: 'leading-none disabled:text-grey',
      icon: '',
    },
    sm: { button: 'p-0.5', icon: 'w-6 h-6' },
    lg: { button: 'p-1', icon: 'w-[30px] h-[30px
  },
  rounded: {
    default: {
      button:
        'text-secondary bg-primary border-3 roun
      icon: '',
    },
    sm: { button: '', icon: 'w-10 h-10' },
    lg: { button: 'p-px', icon: 'w-[52px] h-[52p
  },
  square: {
    default: {
```

Development process

- We were able to develop components with several developers at the same time, because we were independent of each other
- it was possible to scale the team and speed up
- The design tokens were the basis for everyone

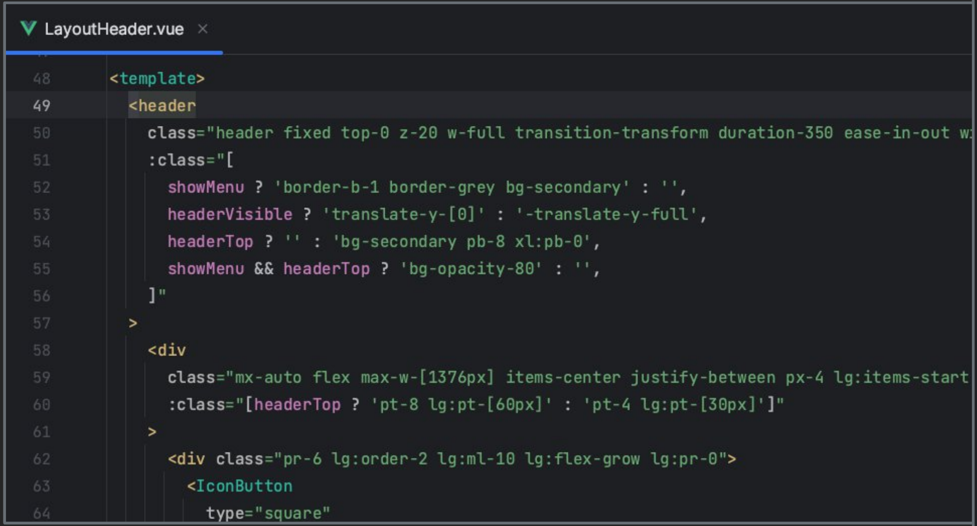
Development process

- In the next step, we integrated the individual small components into the demo store template



LayoutHeader.vue

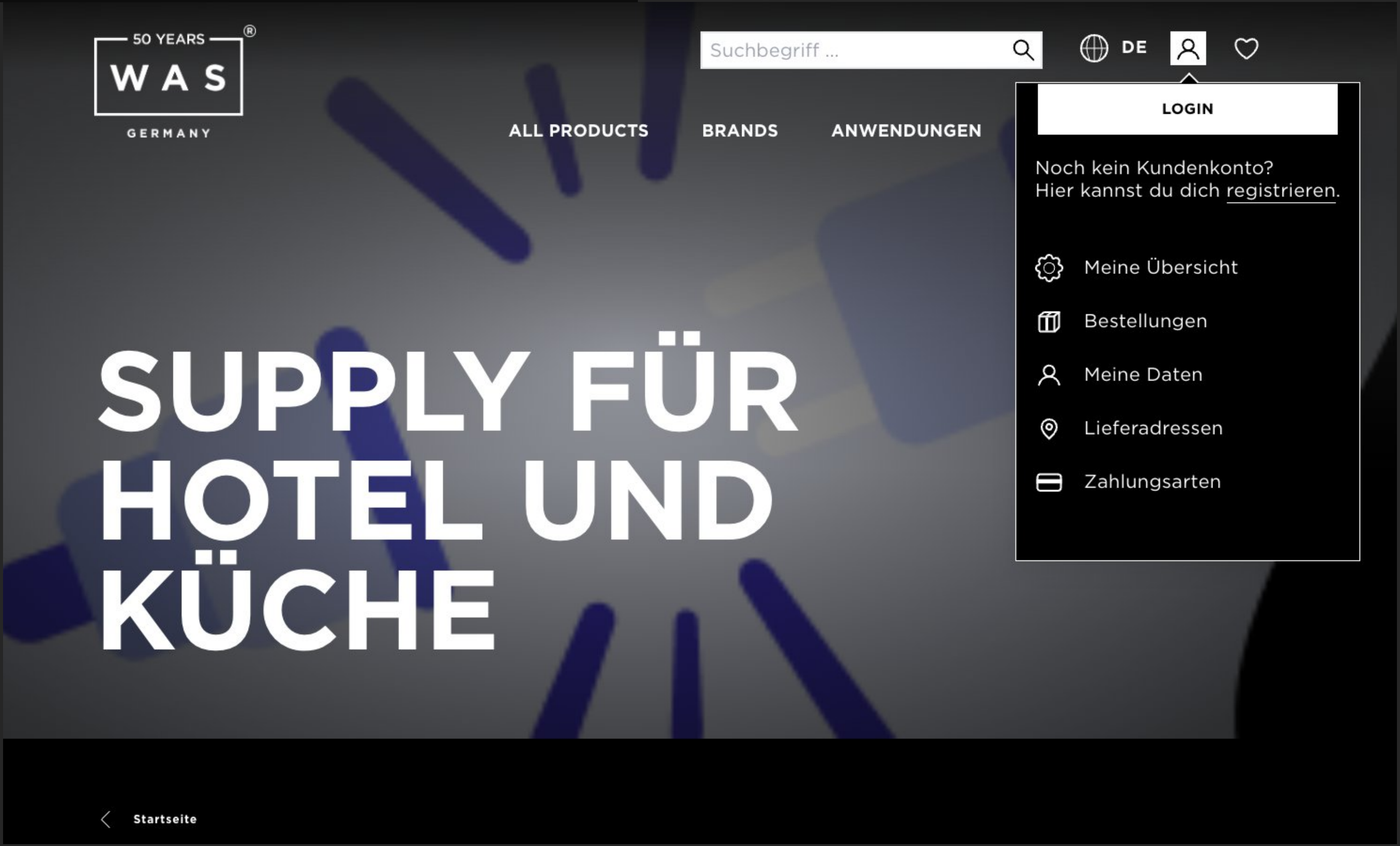
```
<div class="pr-10 lg:order-3">
  <IconButton
    type="square"
    size="sm"
    icon="symbols:search"
    class="h-[38px] lg:ml-6 lg:hidden"
    @click="open = 'search'"
  />
  <Overlay
    :visible="open === 'search'"
    position="middle"
    :show-logo="false"
    title="Where service begins"
    backdrop
    @close="open = null"
  >
    <HeaderOffcanvasSearch @navigate="open = null" />
  </Overlay>

  <div class="my-[6.5px] hidden h-[36px] lg:ml-6 lg:block">
    <SearchInput
      ref="searchInput"
      placeholder="Suchbegriff ..."
      name="search"
      autocomplete="off"
      class="border-2"
      @click="open = 'search'"
    />
  </div>
</div>

<div class="hidden lg:order-4 lg:block">
  <Dropdown
    type="normal"
```



LayoutHeader.vue

```
<template>
  <header
    class="header fixed top-0 z-20 w-full transition-transform duration-350 ease-in-out w
    :class="[
      showMenu ? 'border-b-1 border-grey bg-secondary' : '',
      headerVisible ? 'translate-y-[0]' : '-translate-y-full',
      headerTop ? '' : 'bg-secondary pb-8 xl:pb-0',
      showMenu && headerTop ? 'bg-opacity-80' : '',
    ]"
  >
    <div
      class="mx-auto flex max-w-[1376px] items-center justify-between px-4 lg:items-start
      :class="[headerTop ? 'pt-8 lg:pt-[60px]' : 'pt-4 lg:pt-[30px]']"
    >
      <div class="pr-6 lg:order-2 lg:ml-10 lg:flex-grow lg:pr-0">
        <IconButton
          type="square"
```

Composable frontends in action: Development process



Suchbegriff ...

DE

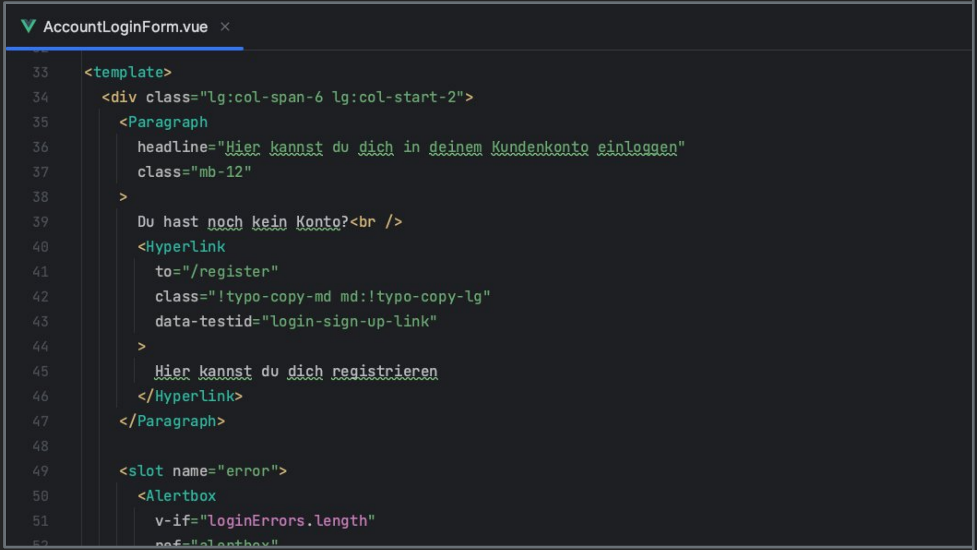
LOGIN

Noch kein Kundenkonto?
Hier kannst du dich registrieren.

- Meine Übersicht
- Bestellungen
- Meine Daten
- Lieferadressen
- Zahlungsarten

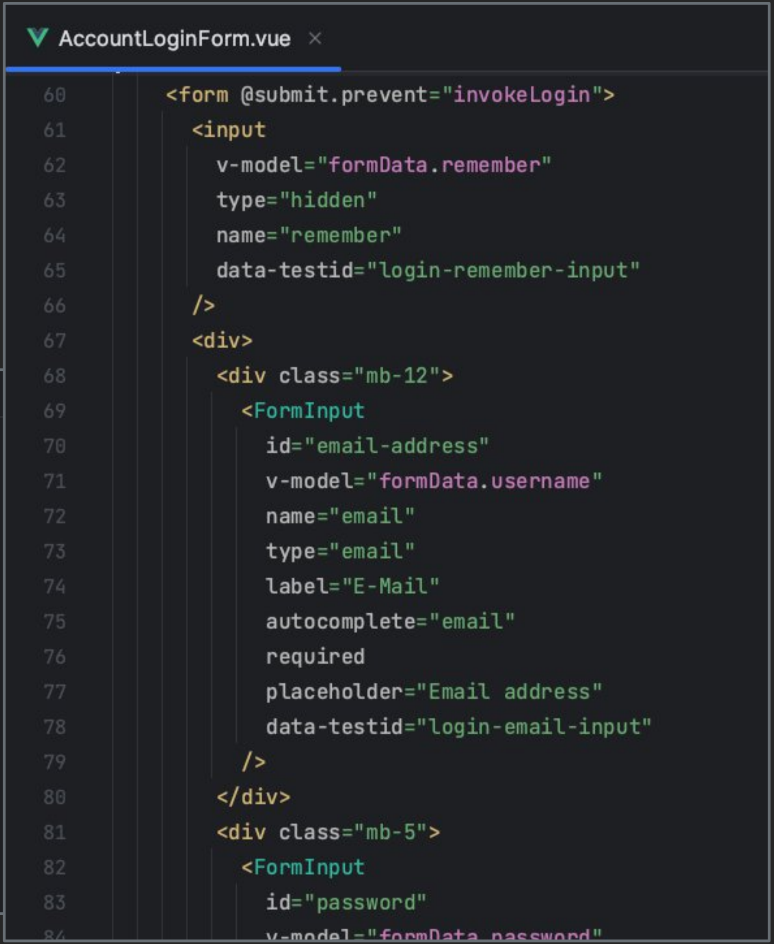
Development process

- In the next step, we integrated the individual small components into the demo store template



AccountLoginForm.vue

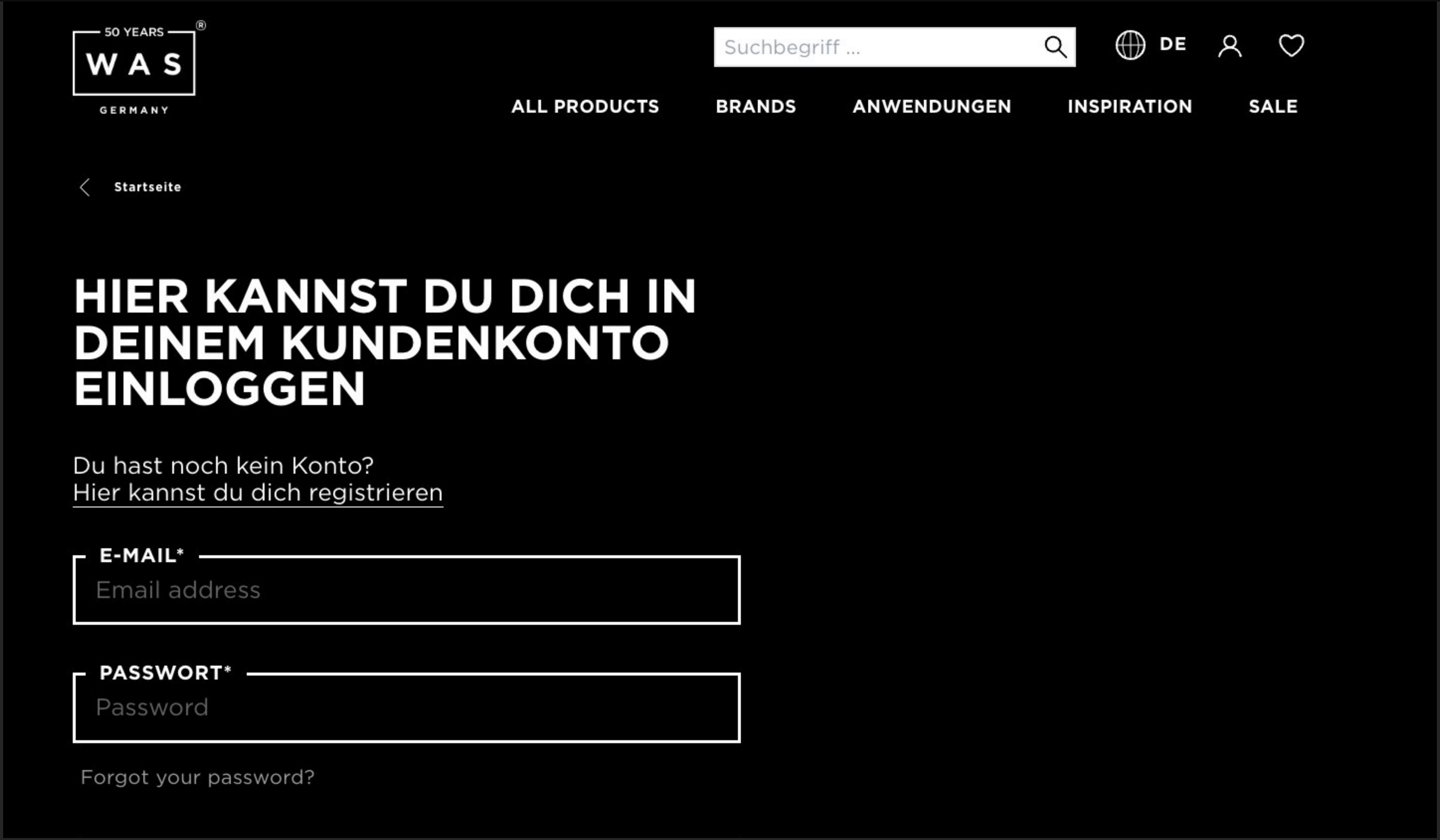
```
<template>
  <div class="lg:col-span-6 lg:col-start-2">
    <Paragraph
      headline="Hier kannst du dich in deinem Kundenkonto einloggen"
      class="mb-12"
    >
      Du hast noch kein Konto?<br />
      <Hyperlink
        to="/register"
        class="!typo-copy-md md:!typo-copy-lg"
        data-testid="login-sign-up-link"
      >
        Hier kannst du dich registrieren
      </Hyperlink>
    </Paragraph>

    <slot name="error">
      <Alertbox
        v-if="loginErrors.length"
        ref="alertbox"
```



AccountLoginForm.vue

```
<form @submit.prevent="invokeLogin">
  <input
    v-model="formData.remember"
    type="hidden"
    name="remember"
    data-testid="login-remember-input"
  />
  <div>
    <div class="mb-12">
      <FormInput
        id="email-address"
        v-model="formData.username"
        name="email"
        type="email"
        label="E-Mail"
        autocomplete="email"
        required
        placeholder="Email address"
        data-testid="login-email-input"
      />
    </div>
    <div class="mb-5">
      <FormInput
        id="password"
        v-model="formData.password"
```

Composable frontends in action: Development process



50 YEARS WAS GERMANY

Suchbegriff ...

DE

ALL PRODUCTS BRANDS ANWENDUNGEN INSPIRATION SALE

Startseite

HIER KANNST DU DICH IN DEINEM KUNDENKONTO EINLOGGEN

Du hast noch kein Konto?
Hier kannst du dich registrieren

E-MAIL*
Email address

PASSWORT*
Password

Forgot your password?

DevOps / Hosting

- Separate projects for frontend and Shopware core
  - 2 Git Repositories
  - 2 Gitlab CI/CD Pipelines
  - 2 vHost

- Timme Hosting

- Supervisor as process manager for node application

Learnings

- It is important to take time for the basic configuration of design tokens and the small components
- Close communication between frontend and UX/UI team for the basic configuration
- We need a project blueprint for a component library in figma and for shopware composable frontends to speed up the basics

Learnings

- You can use composable frontends quite well, but it's not perfect yet
  - In some places you have to find individual temporary solutions and overwrite some composables
- Many plugins from the store are not yet implemented api first. So, we need more plugins with api support :)
- There are currently many changes to composable frontends
- Happy developers and UX/UI designer

Conclusion and next steps

- Based on our experiences from our first shopware composable frontends project:
  - Define component library and project template for figma as blueprint
  - Define project vue.js / tailwind template with shopware composable frontends (maybe based on blank template) as blueprint

- Setup development and testing process with storybook

- Next project starts in the next days with new challenges: b2bsuite and payment

Download slides

Thank you!