BOOTSTRAP INSTALLATIE MET CDN

https://getbootstrap.com/

1. Basis HTML Setup met Bootstrap CDN

Je hoeft alleen de Bootstrap CSS- en JavaScript-bestanden in te laden via de CDN-links in de `<head>` en vóór de sluitende `</body>` tag in je HTML-bestand.

Hier is een voorbeeld van een basis HTML-pagina met Bootstrap 5.3.2 via CDN:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Bootstrap 5.3.3 Project</title>
    <!-- Bootstrap CSS (via CDN) -->
    <link

href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css
"
      rel="stylesheet"
      integrity="sha384-
zWj6hrx8xAYoOeYIbJJ8eqh3jSNv3xkKwUgZMycwBJFqS/ud7NYi85zvcu/HWcuG"
      crossorigin="anonymous"
    />
  </head>
  <body>
    <div class="container">
      <h1 class="text-center">Hello, Bootstrap 5.3.3!</h1>
      <p class="text-center">Dit is een eenvoudige Bootstrap setup via
CDN.</p>
    </div>
      <!-- Bootstrap JavaScript (via CDN) -->
    <script

src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.mi
n.js"
      integrity="sha384-
qg4wKY7LX5pJkueRlQlvR64K/Y9txzjzG1DT+0isGrjZlhtxv3XhZLkpDxlHvD5g"
      crossorigin="anonymous"
    ></script>
  </body>
</html>
```

2. Toelichting

- **CSS via CDN**: In de `<head>` voeg je de link toe naar de Bootstrap CSS via de jsDelivr CDN. Dit zorgt ervoor dat je toegang hebt tot alle Bootstrap 5.3.2 stijlen zonder lokaal bestanden te hoeven installeren.

- **Popper.js via CDN**: Popper.js is nodig voor sommige van Bootstrap's JavaScript-componenten (zoals dropdowns en tooltips). Deze moet je vóór de Bootstrap JavaScript-bestanden laden.

- **JavaScript via CDN**: Laad de Bootstrap JavaScript-bestanden via CDN net vóór de sluitende `</body>`-tag. Dit zorgt ervoor dat de interactieve onderdelen van Bootstrap (zoals modals, tooltips, dropdowns, enz.) werken.

3. Voor- en nadelen van CDN-gebruik

Voordelen:

- **Snellere setup**: Je hebt geen npm of build-processen nodig. Gewoon de CDN-links inladen in je HTML.
- **Snellere laadtijden**: Browsers cachen vaak bestanden van CDNs. Dit kan de laadtijd verbeteren, vooral als de gebruiker al andere websites heeft bezocht die dezelfde Bootstrap CDN gebruiken.
- **Eenvoud**: Geen build-tools zoals Webpack of Vite nodig; je kunt direct beginnen.

Nadelen:

- **Minder controle**: Je kunt de Bootstrap-stijlen niet zo gemakkelijk aanpassen, omdat je geen SCSS-bestanden importeert of aanpast.
- **Afhankelijk van een externe server**: Als de CDN om welke reden dan ook niet beschikbaar is, werkt je site niet zoals verwacht.
- **Geen versiebeheer**: Je zit vast aan de versie die via de CDN wordt geladen, tenzij je handmatig de versie-upgrades bijhoudt door de URL te veranderen.

4. Extra functionaliteiten toevoegen

Je kunt nog steeds aangepaste stijlen of JavaScript toevoegen aan je project. Plaats je aangepaste CSS-bestanden **ná** de Bootstrap CDN in je HTML, zodat jouw stijlen de Bootstrap-stijlen overschrijven indien nodig.

```
<link href="custom.css" rel="stylesheet">
```

Je kunt ook extra JavaScript-bestanden toevoegen **ná** het Bootstrap CDN-script in je HTML.

```
<script src="custom.js"></script>
```

Samenvatting:

- Voeg de Bootstrap CSS en JavaScript links via een CDN toe aan je project.
- Deze methode is eenvoudig en snel, ideaal voor kleinere projecten of wanneer je snel een prototype wilt bouwen.
- Je kunt nog steeds je eigen CSS en JavaScript toevoegen om het gedrag en de stijl aan te passen.

Met deze eenvoudige setup via de CDN kun je meteen beginnen met het gebruiken van **Bootstrap 5.3.2**!

BOOTSTRAP INSTALLATIE MET NPM

Wat is NPM?

NPM = Node Package Manager

- NPM wordt **standaard meegeleverd met Node.js**. Node.js is een javascript framework.
- NPM gebruik je om **packages (libraries/modules)** te installeren die je in je project kan gebruiken, zoals:
  - `npm install bootstrap`
  - `npm install tailwindcss`
  - ...
- **BELANGRIJK:** NPM wordt *niet apart geïnstalleerd*, maar **komt automatisch mee met Node.js**. Dus wanneer je via **NVM Node installeert**, krijg je **NPM er gratis bij**.

Wat is NVM?

NVM = Node Version Manager

- Het is een tool waarmee je **meerdere versies van Node.js op één computer kan beheren**.
- Je kan met één commando wisselen tussen versies, bv. een project draait op **Node 18**, een ander op **Node 22** → NVM maakt dat makkelijk.
- NVM installeert **Node én NPM automatisch** per versie — dus **elke Node-versie heeft zijn eigen NPM**.

➡ Waarom is dat nuttig?

- Sommige frameworks of projecten werken enkel op **bepaalde Node-versies**.
- Zonder NVM moet je Node telkens **verwijderen en opnieuw installeren** — dat is onhandig.
- Met NVM kan je gewoon zeggen: `nvm use 20` → en klaar.

Samengevat:

Tool	Betekenis	Wat doet het?	Installeer je het apart?
NVM	Node Version Manager	Beheert meerdere Node-versies	✓ Ja, apart installeren
Node.js	JavaScript runtime	Draait JS buiten de browser	⚙ Geïnstalleerd via NVM
NPM	Node Package Manager	Installeert packages/modules	🎁 Wordt meegeleverd met Node.js

Conclusie voor installatie op je systeem:

1. **Je installeert NVM** → dit is de manager.
2. **Met NVM installeer je Node.js** → dat zet Node op je systeem.
3. **NPM zit automatisch inbegrepen bij Node.js** → geen aparte installatie nodig.

Stap 1 — Controleer of je al NVM hebt

Begin een nieuw webproject in je www folder, Bijvoorbeeld bootstrapsetup en open deze in je editor.

Open **PowerShell** of je terminal in je editor en typ:

```
nvm version
```

Stap 2 — Download NVM voor Windows

1. Ga naar de officiële GitHub release van **nvm-windows**: 👉 https://github.com/coreybutler/nvm-windows/releases/latest
2. Download **nvm-setup.exe**
3. Voer de installer uit 👉 **installeer in een pad zonder spaties**, voorbeeld:

```
C:\nvm
```

Hij vraagt ook waar Node.js zelf moet staan → gebruik bijvoorbeeld:

```
C:\nodejs
```

PowerShell moet als Administrator geopend zijn tijdens installatie.

Stap 3 — Controleer of NVM werkt in PowerShell

Sluit je editor en open je terminal/ **PowerShell opnieuw** en test:

```
nvm version
```

Als je een versie ziet → klaar om Node te installeren.

Stap 4 — Installeer Node.js met NVM (voorbeeld: versie 22)

```
nvm install 25.0.0
nvm use 25.0.0
```

Je kunt zien welke versies beschikbaar zijn:

```
nvm list available
```

Je geïnstalleerde versies bekijken:

```
nvm list
```

Stap 5 — Controleer je Node & npm versie

```
node -v
npm -v
```

1. Initieer je project met npm

Op de node package manager te gebruiken van node.js dien je eerst node.js te installeren. Dit hebben we in voorgaande dus gedaan

Als je project nog geen npm-project is, initialiseer je het met het volgende commando:

`npm init -y` . Dit installeert een package.json bestand.

2. Installeer de nieuwste versie van Bootstrap en Popper.js

Gebruik npm om de laatste versie van Bootstrap en Popper.js te installeren:

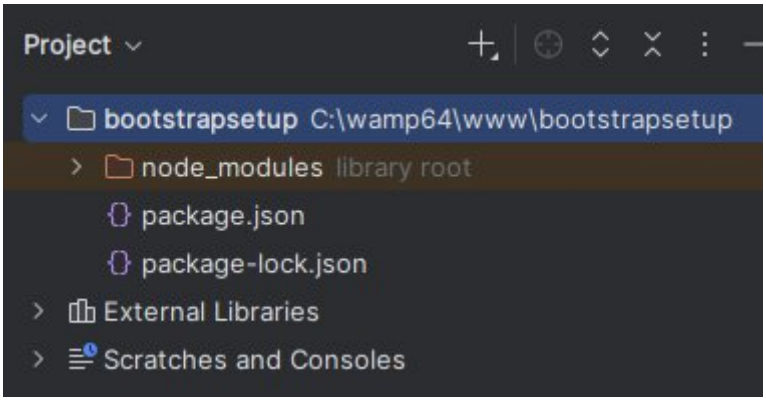
```
npm install –save bootstrap @popperjs/core (dash dash save!)
```

3. Bundling en build tools (bijv. Webpack, Vite)

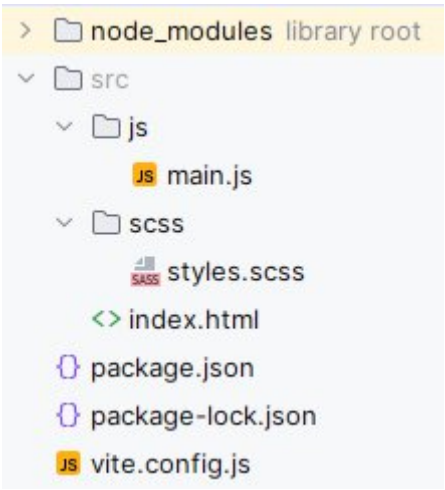
Je hebt een bundler nodig om SCSS te compileren naar CSS en je JavaScript-bestanden te bundelen. Hier geef ik een voorbeeld met **Vite** (een moderne en snelle build tool):

1. Installeer Vite:

```
npm install –save-dev vite (dash dash save-dev vite!)
```



2. Update nu je projectstructuur zodat het er als volgt zal uitzien. Maak bestanden en mappen aan waar nodig:



3. Open main.js:

```
// Import our custom CSS
import '../scss/styles.scss'
// Import all of Bootstrap's JS
import * as bootstrap from 'bootstrap'
```

4. Open index.html en de module main.js toe

```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, user-scalable=no,
initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <script type="module" src="./js/main.js"></script>
```

5. Open styles.scss en voeg de scss van bootstrap toe

```
// Import all of Bootstrap's CSS
@import "bootstrap/scss/bootstrap";
```

6. Open je vite.config.js (configuratiebestand):

```
import { resolve } from 'path'
export default {
    root: resolve(__dirname, 'src'),
    build: {
        outDir: '../dist'
    },
    base:'./',
    server: {
        port: 8080
    }
}
```

7. Open package.json en voeg de scripts toe om je project te developen of te builden.

```
"scripts": {
  "start": "vite",
  "build": "vite build",
  "test": "echo \"Error: no test specified\" && exit 1"
},
```

4. Installeer sass dependency

npm install -D sass-embedded

sass-embedded is de officiële Sass-compiler nieuwe stijl, ontwikkeld door het Sass-team zelf, bedoeld om SCSS-bestanden (.scss) te vertalen naar gewone CSS.

5. Gebruik SCSS voor aanpasbare thema's

Als je de SCSS-bestanden van Bootstrap wilt gebruiken om variabele aan te passen, dan dien je de onderstaande volgorde aan te houden. Variabelen bovenaan, dan bootstrap inladen en vervolgens je eigen css:

```
// Import our custom variables
$primary:purple;
// Import all of Bootstrap's CSS
@import "bootstrap/scss/bootstrap";

h1{color:red;}
```

6. Compileer en start de development server

Nu je configuratie klaar is, kun je Vite gebruiken om je development server te starten en je bestanden te bundelen.

In je terminal voer je het volgende uit om de development server te starten:

```
npm run start
```

een snelle lokale server en bundelt je SCSS en JavaScript. Je kunt nu in je browser naar `http://localhost:8080` gaan om je project te zien.

7. Productiebouw

Als je klaar bent om je project te bundelen voor productie, kun je Vite laten bouwen door het volgende commando uit te voeren:

```
npm run build
```

Dit zal de productieklare bestanden plaatsen in de map `dist/`, klaar voor deployment naar je server.

REGELS OM MET BOOTSTRAP TE WERKEN

Om met bootstrap aan de slag te gaan geven we hier enkele vuistregels mee. Het is normaal dat je deze regels nu nog niet snapt omdat we in de volgende hoofdstukken deze eerst één voor één zullen bekijken. Nadat je deze hoofdstukken hebt doornomen kom je naar hier terug en zal je zien dat dit een zeer goede leidraad is om bootstrap pagina's te maken.

Regels met `row` en `col`	**Regels met `row-cols`**
1. Begin met een `container`: Plaats altijd een Bootstrap-container (`.container`, `.container-fluid`, of `.container-{breakpoint}`) als basis om inhoud centraal te structureren en responsief te maken.	**1. Begin met een `container`**: Gebruik een `.container`, `.container-fluid`, of `.container-{breakpoint}` om de basis te leggen voor een centraal uitgelijnde en responsieve layout.
2. Voeg een `row` toe binnen de container: Elke `container` bevat één of meer `rows` om de kolommen netjes in lijn te houden.	**2. Gebruik een `row` met `row-cols`-klassen**: In plaats van individuele kolomdefinities, maak je een enkele `row` aan met de `row-cols`-klassen (`.row-cols-2`, `.row-cols-md-3`, etc.). Hiermee bepaal je automatisch het aantal kolommen per rij, afhankelijk van het schermformaat, zonder specifieke `col`-klassen toe te voegen.
3. Gebruik `col`-klassen binnen elke `row`: In een `row` voeg je telkens een `col` toe. Gebruik de responsieve kolomklassen (`col`, `col-md`, `col-lg`, etc.) om te bepalen hoeveel ruimte elke kolom inneemt op verschillende schermformaten.	**3. Automatische Responsiviteit**: `row-cols` past automatisch het aantal kolommen aan op basis van de schermgrootte. Gebruik `row-cols-{breakpoint}-{number}` om verschillende aantallen kolommen per schermformaat te specificeren. Bijvoorbeeld, `.row-cols-1 .row-cols-md-2 .row-cols-lg-4` creëert één kolom op kleine schermen, twee kolommen op middelgrote schermen, en vier op grote schermen.
4. Nesting: `rows` en `cols` kunnen genest worden voor complexere indelingen. Elke geneste kolom kan een nieuwe `row` bevatten, die op haar beurt weer `col`-elementen kan bevatten.	-
5. Flexbox Positionering: Binnen de kolommen kunnen alle mogelijke tags gepositioneerd worden met flexbox. Maak gebruik van flexbox-utility-klassen, zoals `d-flex`, `justify-content-*`, `align-items-*` op de `col`-parent, en `align-self-*` op individuele elementen voor nauwkeurige uitlijning.	**4. Direct Positioneren binnen Kolommen**: Binnen de kolommen kun je flexbox-utility-klassen toepassen zoals `d-flex`, `justify-content-*`, `align-items-*`, en `align-self-*` om elementen flexibel te positioneren zonder individuele `col`-klassen.
6. Gebruik van Utility Classes voor Afstand: Creëer ruimte tussen en rond elementen door middel van `padding` en	**5. Afstand tussen Kolommen**: Gebruik `gutter`-klassen (`g-0`, `g-2`, `g-3`, etc.) om de ruimte tussen kolommen te beheren. Deze

Regels met `row` en `col`	Regels met `row-cols`
`margin` utility-classes zoals `.mt-3`, `.px-4`, etc., en maak gebruik van `gutter`-klassen (`g-0`, `g-sm-3`, etc.) om de ruimte tussen kolommen te beheren. Dit zorgt voor een consistent en responsief ontwerp.	klassen zorgen ervoor dat de kolommen op een consistente manier worden gescheiden.
7. Responsive Layouts: Houd rekening met verschillende schermformaten door breakpoints te gebruiken in `col`-klassen en `gutter`-klassen, zodat de lay-out goed blijft werken op zowel mobiele als desktop-apparaten.	**6. Gebruik Utility Classes voor Afstanden**: Voor aanvullende marges en padding kun je Bootstrap's standaard utility-classes toepassen (`.mt-3`, `.px-4`, etc.) om elementen extra ruimte te geven binnen de `row-cols` layout.
-	**7. Nested Flexbox**: Wanneer nodig, kun je elementen binnen de kolommen verder structureren met flexbox door `d-flex` en andere flexbox-utility-klassen toe te voegen voor specifieke uitlijning en positionering.

Voorbeeld: 3 cards naast elkaar volgens beide bovenstaande regels.

Voorbeeld 1: Gebruik van `row` en `col`

```
<div class="container">
  <div class="row">
    <div class="col-md-4 mb-4">
      <div class="card">
        <img src="https://via.placeholder.com/150" class="card-img-top"
alt="Placeholder image">
        <div class="card-body">
          <h5 class="card-title">Card Title 1</h5>
          <p class="card-text">Some quick example text to build on the card
title and make up the bulk of the card's content.</p>
          <a href="#" class="btn btn-primary">Go somewhere</a>
        </div>
      </div>
    </div>
    <div class="col-md-4 mb-4">
      <div class="card">
        <img src="https://via.placeholder.com/150" class="card-img-top"
alt="Placeholder image">
        <div class="card-body">
          <h5 class="card-title">Card Title 2</h5>
          <p class="card-text">Some quick example text to build on the card
title and make up the bulk of the card's content.</p>
          <a href="#" class="btn btn-primary">Go somewhere</a>
        </div>
      </div>
    </div>
    <div class="col-md-4 mb-4">
      <div class="card">
        <img src="https://via.placeholder.com/150" class="card-img-top"
alt="Placeholder image">
        <div class="card-body">
```

```
          <h5 class="card-title">Card Title 3</h5>
          <p class="card-text">Some quick example text to build on the card
title and make up the bulk of the card's content.</p>
          <a href="#" class="btn btn-primary">Go somewhere</a>
        </div>
      </div>
    </div>
  </div>
</div>
```

Voorbeeld 2: Gebruik van row-cols

```
<div class="container">
  <div class="row row-cols-1 row-cols-md-3 g-4">
    <div class="col">
      <div class="card">
        <img src="https://via.placeholder.com/150" class="card-img-top"
alt="Placeholder image">
        <div class="card-body">
          <h5 class="card-title">Card Title 1</h5>
          <p class="card-text">Some quick example text to build on the card
title and make up the bulk of the card's content.</p>
          <a href="#" class="btn btn-primary">Go somewhere</a>
        </div>
      </div>
    </div>
    <div class="col">
      <div class="card">
        <img src="https://via.placeholder.com/150" class="card-img-top"
alt="Placeholder image">
        <div class="card-body">
          <h5 class="card-title">Card Title 2</h5>
          <p class="card-text">Some quick example text to build on the card
title and make up the bulk of the card's content.</p>
          <a href="#" class="btn btn-primary">Go somewhere</a>
        </div>
      </div>
    </div>
    <div class="col">
      <div class="card">
        <img src="https://via.placeholder.com/150" class="card-img-top"
alt="Placeholder image">
        <div class="card-body">
          <h5 class="card-title">Card Title 3</h5>
          <p class="card-text">Some quick example text to build on the card
title and make up the bulk of the card's content.</p>
          <a href="#" class="btn btn-primary">Go somewhere</a>
        </div>
      </div>
    </div>
  </div>
</div>
```

Uitleg

- **Voorbeeld 1 (row & col)**: Hier gebruik ik een `row` met drie `col-md-4` kolommen, zodat de kaarten op grotere schermen in drie kolommen worden weergegeven. Op kleinere schermen worden ze automatisch één per rij.
- **Voorbeeld 2 (row-cols)**: Hier gebruik ik `row row-cols-1 row-cols-md-3 g-4`, waarmee het aantal kolommen per schermformaat automatisch wordt bepaald. `row-cols-1` zorgt voor één kaart per rij op kleine schermen, en `row-cols-md-3` toont drie kaarten naast elkaar op middelgrote schermen en groter.

VERSCHIL TUSSEN CONTAINER EN CONTAINER_FLUID

Container

De `.container`-class wordt gebruikt om een vaste breedte aan je content te geven, afhankelijk van het schermformaat (breakpoints). Het zorgt ervoor dat je content netjes wordt uitgelijnd in het midden van de pagina en dat het ontwerp niet te breed wordt op grote schermen. Dit is handig voor traditionele layouts waarbij je niet wilt dat de content helemaal uitrekt tot aan de randen van het scherm.

Hier zijn de vaste breedtes die Bootstrap hanteert voor `.container` op verschillende schermformaten: - **Extra small (XS)** (tot 576px): 100% breedte (geen vaste breedte) - **Small (SM)** (≥ 576px): 540px - **Medium (MD)** (≥ 768px): 720px - **Large (LG)** (≥ 992px): 960px - **Extra Large (XL)** (≥ 1200px): 1140px - **XXL** (≥ 1400px): 1320px

Container-fluid

De `.container-fluid`-class zorgt ervoor dat je container altijd 100% breed is, ongeacht het schermformaat. Dit betekent dat je content van de ene rand van het scherm tot de andere reikt, wat handig is als je volledige schermruimte wilt benutten voor bepaalde designs.

Voorbeeld:

```
<div class="container">
  <p>Dit is een vaste container. Het zal afhankelijk van het schermformaat
een bepaalde vaste breedte hebben.</p>
</div>

<div class="container-fluid">
  <p>Dit is een fluid container. Het rekt altijd uit over de volledige
breedte van het scherm.</p>
</div>
```

Bootstrap Breakpoints

Bootstrap maakt gebruik van breakpoints om responsieve layouts te creëren. De breakpoints bepalen wanneer de layout verandert op basis van het schermformaat. Hier zijn de belangrijkste breakpoints in Bootstrap 5:

- **Extra small (XS)**: Voor schermen kleiner dan 576px.
- **Small (SM)**: Voor schermen van 576px en groter.
- **Medium (MD)**: Voor schermen van 768px en groter.
- **Large (LG)**: Voor schermen van 992px en groter.
- **Extra Large (XL)**: Voor schermen van 1200px en groter.

- **XXL**: Voor schermen van 1400px en groter.

Je kunt deze breakpoints gebruiken om je layout aan te passen aan verschillende schermgroottes. Zo kun je bijvoorbeeld kolommen maken die op grote schermen naast elkaar staan, maar op kleine schermen onder elkaar worden weergegeven.

Voorbeeld van het grid-systeem met breakpoints:

```
<div class="container">
  <div class="row">
    <div class="col-sm-12 col-md-6 col-lg-4">Kolom 1</div>
    <div class="col-sm-12 col-md-6 col-lg-4">Kolom 2</div>
    <div class="col-sm-12 col-lg-4">Kolom 3</div>
  </div>
</div>
```

- Op **extra small** schermen (XS) zullen de kolommen volledige breedte innemen.
- Op **small** schermen (SM) en groter worden de kolommen verdeeld in twee per rij.
- Op **large** schermen (LG) en groter staan er drie kolommen naast elkaar.

Met deze toevoegingen krijgt je cursus een duidelijkere uitleg over het gebruik van containers en de responsieve breakpoints in Bootstrap.

Stap 1: Layout maken met Bootstrap's Grid Systeem

Bootstrap heeft een grid-systeem waarmee je makkelijk kolommen en rijen kunt maken. Hier is een voorbeeld:

```
<div class="container">
    <div class="row">
        <div class="col-md-4">
            <p>Kolom 1</p>
        </div>
        <div class="col-md-4">
            <p>Kolom 2</p>
        </div>
        <div class="col-md-4">
            <p>Kolom 3</p>
        </div>
    </div>
</div>
```

Uitleg:

- **`row`**: Hiermee creëer je een rij in het grid.
- **`col-md-4`**: Dit betekent dat elke kolom 4 van de 12 kolommen op een medium scherm gebruikt, dus ze nemen samen de volledige breedte in.

Met deze opzet kun je eenvoudig layouts maken die zich aanpassen aan verschillende schermgroottes.

Stap 2: Buttons en Typografie

Bootstrap biedt een aantal handige stijlen voor knoppen en tekst. Hier is een voorbeeld:

```
<div class="container">
    <button class="btn btn-primary">Primaire Knop</button>
    <button class="btn btn-secondary">Secundaire Knop</button>
</div>
```

Uitleg:

- **`btn`**: Dit is de basisclass voor knoppen.
- **`btn-primary`**: Dit geeft de knop een primaire stijl (meestal blauw).
- **`btn-secondary`**: Dit is een secundaire stijl (meestal grijs).

Stap 3: Responsieve afbeeldingen

Bootstrap zorgt er ook voor dat afbeeldingen automatisch schalen en responsief zijn. Je kunt eenvoudig een afbeelding toevoegen en responsief maken met de `img-fluid` class:

```
<div class="container">
    <img src="https://via.placeholder.com/400x300" class="img-fluid"
alt="Voorbeeldafbeelding">
</div>
```

Uitleg:

- **`img-fluid`**: Dit zorgt ervoor dat de afbeelding automatisch schaalt binnen de container waarin deze zich bevindt en zich aanpast aan de schermgrootte.

Stap 4: Navigatiebalk

Bootstrap maakt het eenvoudig om een navigatiebalk (navbar) te maken. Hier is een basisvoorbeeld:

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
    <div class="container-fluid">
        <a class="navbar-brand" href="#">Mijn Website</a>
        <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-
expanded="false" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNav">
            <ul class="navbar-nav">
                <li class="nav-item">
                    <a class="nav-link active" href="#">Home</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="#">Features</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="#">Contact</a>
                </li>
            </ul>
        </div>
    </div>
</nav>
```

Uitleg:

- **`navbar`**: Dit is de basisclass voor een navigatiebalk.
- **`navbar-expand-lg`**: Dit zorgt ervoor dat de navigatiebalk uitbreidt op grotere schermen en inklapt op kleinere schermen.
- **`navbar-light` en `bg-light`**: Dit zorgt voor een lichte stijl voor de navigatiebalk.
- **`navbar-toggler`**: Hiermee voeg je een knop toe om de navigatie op kleine schermen in te klappen.
- **`navbar-nav`**: Dit is de lijst met navigatie-items.
- **`nav-link`**: Deze class wordt gebruikt voor individuele links binnen de navigatie.

Stap 5: Cards

Een andere handige component in Bootstrap is de “card”. Dit is een flexibele manier om content te presenteren in een nette opmaak.

```
<div class="container">
    <div class="card" style="width: 18rem;">
        <img src="https://via.placeholder.com/150" class="card-img-top"
alt="Afbeelding">
        <div class="card-body">
            <h5 class="card-title">Kaart Titel</h5>
            <p class="card-text">Dit is een voorbeeld van een eenvoudige
Bootstrap-kaart. Je kunt hiermee content in een overzichtelijke lay-out
presenteren.</p>
            <a href="#" class="btn btn-primary">Ga ergens heen</a>
        </div>
    </div>
</div>
```

Uitleg:

- **`card`**: De basisclass voor een kaart.
- **`card-img-top`**: Hiermee voeg je een afbeelding toe aan de bovenkant van de kaart.
- **`card-body`**: De inhoud van de kaart, waar de titel, tekst en knop zich bevinden.

Stap 6: Modals

Modals zijn pop-ups die veel worden gebruikt om extra informatie weer te geven zonder dat de gebruiker de huidige pagina verlaat.

```
<div class="container">
    <button type="button" class="btn btn-primary" data-bs-toggle="modal"
data-bs-target="#mijnModal">
        Open Modal
    </button>

    <div class="modal fade" id="mijnModal" tabindex="-1" aria-
labelledby="exampleModalLabel" aria-hidden="true">
        <div class="modal-dialog">
            <div class="modal-content">
                <div class="modal-header">
                    <h5 class="modal-title" id="exampleModalLabel">Modal
Titel</h5>
                    <button type="button" class="btn-close" data-bs-
dismiss="modal" aria-label="Close"></button>
                </div>
                <div class="modal-body">
                    Dit is de inhoud van de modal.
                </div>
                <div class="modal-footer">
                    <button type="button" class="btn btn-secondary" data-bs-
dismiss="modal">Sluiten</button>
                    <button type="button" class="btn btn-
primary">Opslaan</button>
                </div>
            </div>
        </div>
    </div>
</div>
```

Uitleg:

- **`modal`**: De basisclass voor de modal.
- **`fade`**: Zorgt ervoor dat de modal met een fade-effect wordt geopend.
- **`modal-dialog`** en **`modal-content`**: Structureren de inhoud van de modal.
- **`btn-close`**: Sluitknop voor de modal.

Stap 7: Flexbox Utility Classes

Bootstrap maakt gebruik van Flexbox om content eenvoudig uit te lijnen en te positioneren. Hier zijn enkele handige Flexbox utilities die je kunt gebruiken:

```
<div class="container">
    <div class="d-flex justify-content-between">
        <div class="p-2">Element 1</div>
        <div class="p-2">Element 2</div>
        <div class="p-2">Element 3</div>
    </div>
</div>
```

Uitleg:

- **`d-flex`**: Dit verandert het element in een Flexbox-container.
- **`justify-content-between`**: Verspreidt de items gelijkmatig over de container met ruimte tussen de elementen.
- **`p-2`**: Geeft padding aan de elementen.

Andere handige Flexbox utilities:

- **`align-items-center`**: Hiermee centreer je de items verticaal.

Stap 8: Responsief Grid met Breakpoints

Bootstrap's grid-systeem ondersteunt verschillende schermformaten met "breakpoints". Deze breakpoints helpen om verschillende layout aanpassingen te maken voor verschillende schermgroottes. Je kunt bijvoorbeeld bepalen dat een kolom een bepaalde breedte heeft op desktop, maar volledig breed moet zijn op mobiele apparaten.

```
<div class="container">
    <div class="row">
        <div class="col-lg-6 col-md-8 col-sm-12">
            <p>Dit blok neemt 6 kolommen op grote schermen (lg), 8 kolommen
op medium schermen (md), en 12 kolommen op kleine schermen (sm).</p>
        </div>
    </div>
</div>
```

Uitleg:

- **`col-lg-6`**: Op grote schermen neemt deze kolom 6 van de 12 kolommen in beslag (de helft van de breedte).
- **`col-md-8`**: Op medium schermen neemt het 8 van de 12 kolommen in (twee derde van de breedte).
- **`col-sm-12`**: Op kleine schermen neemt het de volledige breedte in (12 van de 12 kolommen).

Dit maakt je layout volledig responsief en aanpasbaar aan elk schermformaat.

Stap 9: Tabellen

Bootstrap biedt eenvoudige manieren om tabellen stijlvol weer te geven. Hier is een voorbeeld van een gestileerde tabel:

```
<div class="container">
    <table class="table">
        <thead>
            <tr>
                <th scope="col">#</th>
                <th scope="col">Naam</th>
                <th scope="col">Leeftijd</th>
                <th scope="col">Functie</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <th scope="row">1</th>
                <td>Jan</td>
                <td>30</td>
                <td>Ontwikkelaar</td>
            </tr>
            <tr>
                <th scope="row">2</th>
                <td>Els</td>
                <td>25</td>
                <td>Designer</td>
            </tr>
        </tbody>
    </table>
</div>
```

Uitleg:

- **`table`**: De basisclass voor een tabel.
- **`thead` en `tbody`**: Scheiden de tabelkop van de tabelinhoud.
- **`scope="col"`**: Helpt bij het correct interpreteren van kolommen door schermlezers.

Je kunt ook stijlen zoals **`table-striped`** of **`table-bordered`** toevoegen om de tabellen er nog fraaier uit te laten zien.

Stap 10: Forms

Bootstrap biedt eenvoudige manieren om nette en consistente formulieren te maken. Hier is een voorbeeld van een formulier met een paar invoervelden en een knop:

```
<div class="container">
    <form>
        <div class="mb-3">
            <label for="email" class="form-label">E-mailadres</label>
            <input type="email" class="form-control" id="email"
placeholder="naam@voorbeeld.com">
        </div>
        <div class="mb-3">
            <label for="wachtwoord" class="form-label">Wachtwoord</label>
            <input type="password" class="form-control" id="wachtwoord"
placeholder="Wachtwoord">
        </div>
        <button type="submit" class="btn btn-primary">Verzenden</button>
    </form>
</div>
```

Uitleg:

- **`form-control`**: Deze class zorgt ervoor dat het invoerveld netjes opgemaakt is en de volledige breedte van zijn container inneemt.
- **`form-label`**: Voegt een label toe voor elk invoerveld.
- **`mb-3`**: Dit geeft een marge onder de invoervelden, zodat ze niet te dicht op elkaar staan.

Stap 11: Customizen met SASS

Bootstrap is volledig gebouwd op SASS, wat betekent dat je de stijlen kunt aanpassen en nieuwe variabelen kunt maken. Als je wilt customizen, kun je Bootstrap downloaden en zelf je eigen stijlen opbouwen met behulp van SASS. Zie voorgaande installatie met npm.

Stap 13: Jumbotron (Hero Secties)

In plaats van de ouderwetse **jumbotron** class (die verwijderd is in de nieuwere versies van Bootstrap), gebruiken we nu een sectie genaamd **hero** om prominente content te tonen.

```
<div class="container py-5">
    <div class="p-5 mb-4 bg-light rounded-3">
        <div class="container-fluid py-5">
            <h1 class="display-5 fw-bold">Welkom bij onze website</h1>
            <p class="col-md-8 fs-4">Met Bootstrap kun je snel een mooie,
responsieve site maken.</p>
            <button class="btn btn-primary btn-lg" type="button">Meer
lezen</button>
        </div>
    </div>
</div>
```

Uitleg:

- **`display-5`**: Een class voor extra grote kopteksten.
- **`fw-bold`**: Verhoogt de tekstdikte (bold).
- **`py-5`**: Voegt verticale padding (ruimte) toe aan de sectie.
- **`bg-light` en `rounded-3`**: Achtergrondkleur en afgeronde hoeken voor een visueel aantrekkelijk effect.

Stap 12: Toasts

Toasts zijn meldingen die je op een elegante manier kunt laten verschijnen zonder dat de pagina volledig opnieuw geladen hoeft te worden. Ze zijn vaak te zien bij acties zoals het opslaan van gegevens of het tonen van statusmeldingen.

```
<div class="toast-container position-fixed bottom-0 end-0 p-3">
    <div class="toast" role="alert" aria-live="assertive" aria-atomic="true">
        <div class="toast-header">
            <strong class="me-auto">Melding</strong>
            <small>11 mins ago</small>
            <button type="button" class="btn-close" data-bs-dismiss="toast"
aria-label="Close"></button>
        </div>
        <div class="toast-body">
            Dit is een voorbeeld van een toastmelding.
        </div>
    </div>
</div>
```

Uitleg:

- **`toast-container`**: Zorgt ervoor dat de toast op een vaste positie wordt weergegeven (zoals rechts onderaan in dit voorbeeld).
- **`toast`**: De basisclass voor de toastcomponent.
- **`btn-close`**: Een knop om de toast te sluiten.

Je kunt de toast dynamisch activeren met JavaScript om hem op het juiste moment te tonen.

Stap 13: Carrousel (Afbeeldingslider)

De carrousel is een veelgebruikte component voor het tonen van een diavoorstelling van afbeeldingen of content. Hier is een voorbeeld:

```
<div id="carouselExample" class="carousel slide" data-bs-ride="carousel">
    <div class="carousel-inner">
        <div class="carousel-item active">
            <img src="https://via.placeholder.com/800x300" class="d-block w-
100" alt="...">
        </div>
        <div class="carousel-item">
            <img src="https://via.placeholder.com/800x300" class="d-block w-
100" alt="...">
        </div>
        <div class="carousel-item">
            <img src="https://via.placeholder.com/800x300" class="d-block w-
100" alt="...">
        </div>
    </div>
    <button class="carousel-control-prev" type="button" data-bs-
target="#carouselExample" data-bs-slide="prev">
        <span class="carousel-control-prev-icon" aria-hidden="true"></span>
        <span class="visually-hidden">Previous</span>
    </button>
    <button class="carousel-control-next" type="button" data-bs-
target="#carouselExample" data-bs-slide="next">
        <span class="carousel-control-next-icon" aria-hidden="true"></span>
        <span class="visually-hidden">Next</span>
    </button>
</div>
```

Uitleg:

- **`carousel slide`**: De basisclass voor de carrousel.
- **`carousel-inner`**: Bevat de individuele dia's (afbeeldingen of content).
- **`carousel-item active`**: Markeert de eerste dia als actief (zichtbaar).
- **`carousel-control-prev` en `carousel-control-next`**: Knoppen om door de dia's heen te navigeren.

Stap 14: Collapse (Inklappen van content)

Bootstrap heeft een handige **collapse** functionaliteit waarmee je delen van de content kunt inklappen en uitbreiden. Dit is handig voor FAQ-secties of content waar je de gebruiker niet meteen alles wilt laten zien.

```
<p>
    <a class="btn btn-primary" data-bs-toggle="collapse"
href="#voorbeeldCollapse" role="button" aria-expanded="false" aria-
controls="voorbeeldCollapse">
        Toon/verberg inhoud
    </a>
</p>
<div class="collapse" id="voorbeeldCollapse">
    <div class="card card-body">
        Dit is ingeklapte inhoud. Klik op de knop om deze te tonen of te
verbergen.
    </div>
</div>
```

Uitleg:

- **`collapse`**: Hiermee wordt het element ingeklapt totdat de gebruiker de knop indrukt.
- **`btn btn-primary`**: De knop waarmee de content wordt ingeklapt of uitgevouwen.

Stap 157: Offcanvas (Zijdelings menu)

Een **offcanvas** is een menu dat vanaf de zijkant van het scherm inschuift. Dit is populair voor mobiele navigatiemenu's.

```
<button class="btn btn-primary" type="button" data-bs-toggle="offcanvas"
data-bs-target="#offcanvasExample" aria-controls="offcanvasExample">
    Open menu
</button>

<div class="offcanvas offcanvas-start" tabindex="-1" id="offcanvasExample"
aria-labelledby="offcanvasExampleLabel">
    <div class="offcanvas-header">
        <h5 class="offcanvas-title" id="offcanvasExampleLabel">Offcanvas
menu</h5>
        <button type="button" class="btn-close" data-bs-dismiss="offcanvas"
aria-label="Close"></button>
    </div>
    <div class="offcanvas-body">
        <p>Dit is een voorbeeld van een offcanvas-menu.</p>
    </div>
</div>
```

Uitleg:

- **`offcanvas`**: De basisclass voor een zijdelings menu.
- **`offcanvas-start`**: Geeft aan dat het menu vanaf de linkerzijde (start) verschijnt.
- **`btn-close`**: Een knop om het menu te sluiten.

Stap 16: Tooltips en Popovers

Bootstrap biedt ingebouwde tooltips en popovers die handig zijn om extra informatie te geven wanneer de gebruiker over een element zweeft.

Tooltips

Tooltips worden meestal gebruikt om korte beschrijvingen te tonen wanneer je over een element gaat.

```
<button type="button" class="btn btn-secondary" data-bs-toggle="tooltip"
data-bs-placement="top" title="Tooltip op de bovenkant">
    Beweeg over mij
</button>

<script>
    var tooltipTriggerList = [].slice.call(document.querySelectorAll('[data-
bs-toggle="tooltip"]'));
    var tooltipList = tooltipTriggerList.map(function (tooltipTriggerEl) {
        return new bootstrap.Tooltip(tooltipTriggerEl);
    });
</script>
```

Uitleg:

- **`data-bs-toggle="tooltip"`**: Deze data-attribuut activeert de tooltip functionaliteit.
- **`data-bs-placement="top"`**: Geeft de locatie van de tooltip aan (in dit geval boven het element).
- **JavaScript**: Je moet een beetje JavaScript toevoegen om de tooltips te initialiseren.

Popovers

Popovers zijn vergelijkbaar met tooltips, maar ze bieden meer ruimte voor content.

```
<button type="button" class="btn btn-lg btn-danger" data-bs-toggle="popover"
title="Popover titel" data-bs-content="Dit is de inhoud van de popover.">
    Klik voor popover
</button>

<script>
    var popoverTriggerList = [].slice.call(document.querySelectorAll('[data-
bs-toggle="popover"]'));
    var popoverList = popoverTriggerList.map(function (popoverTriggerEl) {
        return new bootstrap.Popover(popoverTriggerEl);
    });
</script>
```

Uitleg:

- **`data-bs-toggle="popover"`**: Dit activeert de popover.
- **`title` en `data-bs-content`**: Hiermee voeg je de titel en inhoud van de popover toe.

Stap 17: Bootstrap Iconen

Bootstrap biedt een eigen iconenbibliotheek die je kunt gebruiken zonder externe iconensets te integreren. Je kunt de iconen eenvoudig insluiten door de **Bootstrap Icons** te gebruiken.

Voeg deze link toe aan je HTML-bestand om toegang te krijgen tot de iconen:

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap-icons/font/bootstrap-icons.css" rel="stylesheet">
```

Vervolgens kun je de iconen gebruiken zoals hieronder:

```
<i class="bi bi-alarm"></i>
<i class="bi bi-battery-half"></i>
<i class="bi bi-calendar"></i>
```

Uitleg:

- **`bi bi-alarm`**: `bi` staat voor Bootstrap Icons en `bi-alarm` is de specifieke naam van het alarm-icoon.

Stap 18: Formulieren met Validatie

Bootstrap maakt het eenvoudig om formulieren te valideren met ingebouwde HTML5-validatieregels, gecombineerd met visuele feedback voor gebruikers. Hier is een voorbeeld:

```
<form class="needs-validation" novalidate>
    <div class="mb-3">
        <label for="validationCustom01" class="form-label">Voornaam</label>
        <input type="text" class="form-control" id="validationCustom01" value="" required>
        <div class="valid-feedback">
            Ziet er goed uit!
        </div>
        <div class="invalid-feedback">
            Voer alstublieft een voornaam in.
        </div>
    </div>
    <button class="btn btn-primary" type="submit">Verzenden</button>
</form>

<script>

    (function () {
        'use strict'
        var forms = document.querySelectorAll('.needs-validation')
        Array.prototype.slice.call(forms).forEach(function (form) {
            form.addEventListener('submit', function (event) {
                if (!form.checkValidity()) {
                    event.preventDefault()
                    event.stopPropagation()
                }
                form.classList.add('was-validated')
            }, false)
        })
    })();
</script>
```

Uitleg:

- **`novalidate`**: Hiermee schakel je de standaard HTML-validatie uit, zodat Bootstrap's eigen validatie kan worden gebruikt.
- **`valid-feedback` en `invalid-feedback`**: Geeft gebruikers visuele feedback op basis van de geldigheid van de invoer.
- **JavaScript**: De JavaScript-code zorgt ervoor dat het formulier de juiste validatie toepast wanneer de gebruiker het probeert te verzenden.

Stap 21: Aangepaste Thematisering (Bootstrap Customizer)

Hoewel Bootstrap out-of-the-box geweldig is, wil je soms je eigen stijlen en kleuren toepassen. Bootstrap biedt een eenvoudige manier om dit te doen met variabelen in de bronbestanden die in SASS worden gebruikt. Voor eenvoudige aanpassingen kun je echter ook de **Bootstrap Customizer** gebruiken, waarmee je zonder SASS aanpassingen kunt maken aan je project.

In plaats van zelf SASS te configureren, kun je ook gebruik maken van **CSS-variabelen** in je project. Bijvoorbeeld, je kunt de primaire kleur van Bootstrap aanpassen met het volgende stukje CSS:

```css
:root {
    --bs-primary: #6c757d; /* Grijze kleur in plaats van standaard blauw */
}
```

Uitleg:

- **`:root`**: Hiermee definieer je CSS-variabelen die in je hele project gebruikt kunnen worden.
- **`--bs-primary`**: Dit is de variabele die de primaire kleur van Bootstrap definieert.

Stap 22: Gebruik van Bootstrap met JavaScript Plugins

Bootstrap komt met veel ingebouwde JavaScript-functionaliteiten die je kunt activeren zonder veel handmatig werk. Hier zijn enkele van de belangrijkste plugins:

- **Dropdowns**: Voor geavanceerde dropdownmenu's.
- **Modals**: Voor pop-upvensters.
- **Tooltips**: Voor kleine uitleg of tekstballonnetjes.
- **Carousels**: Voor een dynamische afbeeldingslider.

Als je deze functies wil aanpassen of uitbreiden, kun je gebruik maken van de ingebouwde JavaScript-plugins of zelf extra logica toevoegen.

.

19: Bootstrap in combinatie met JavaScript Frameworks

Wanneer je Bootstrap in combinatie met populaire JavaScript-frameworks zoals **React**, **Vue**, of **Angular** wilt gebruiken, moet je een paar extra stappen nemen om alles soepel te laten werken. We zullen React als voorbeeld nemen, maar hetzelfde principe geldt voor andere frameworks.

Bootstrap met React

Om Bootstrap te gebruiken in een React-project, installeer je de Bootstrap bibliotheek via npm:

```
npm install bootstrap
```

Voeg vervolgens de Bootstrap CSS toe in je `src/index.js` of `src/App.js`:

```
import 'bootstrap/dist/css/bootstrap.min.css';
```

Nu kun je alle standaard Bootstrap-componenten gebruiken in je React-applicatie. Bijvoorbeeld:

```
import React from 'react';

function App() {
  return (
    <div className="container">
      <h1 className="text-center">Welkom bij React met Bootstrap!</h1>
      <button className="btn btn-primary">Klik hier</button>
    </div>
  );
}

export default App;
```

Bootstrap met React-Bootstrap

Je kunt ook **React-Bootstrap** gebruiken, wat specifiek is gebouwd om de Bootstrap-componenten naadloos te integreren in React zonder de noodzaak van jQuery. Installeer dit via npm:

```
npm install react-bootstrap bootstrap
```

In plaats van Bootstrap's CSS-classes te gebruiken, gebruik je nu React-componenten:

```
import React from 'react';
import { Button, Container } from 'react-bootstrap';

function App() {
  return (
    <Container>
      <h1 className="text-center">React-Bootstrap</h1>
      <Button variant="primary">Klik hier</Button>
    </Container>
  );
}

export default App;
```

Stap 20: Geavanceerde Grid Technieken

Hoewel je al bekend bent met het basis grid-systeem van Bootstrap, zijn er enkele geavanceerdere technieken waarmee je veel meer controle hebt over hoe je pagina zich gedraagt op verschillende schermformaten.

Nested Grids

Met **nested grids** kun je grids binnen grids plaatsen. Dit geeft je de mogelijkheid om complexe lay-outs te maken.

```
<div class="container">
    <div class="row">
        <div class="col-md-6">
            <div class="row">
                <div class="col-6">Nested 1</div>
                <div class="col-6">Nested 2</div>
            </div>
        </div>
        <div class="col-md-6">Rechterkolom</div>
    </div>
</div>
```

Hier zien we een rij die is opgesplitst in twee kolommen, waarbij de eerste kolom nogmaals in tweeën wordt gesplitst. Dit maakt nested grids een krachtig hulpmiddel voor complexe layouts.

Order en Offset

Met **order** en **offset** kun je de volgorde van kolommen aanpassen en kolommen op een bepaalde afstand van elkaar plaatsen.

```
<div class="container">
    <div class="row">
        <div class="col-md-4 order-md-2">Kolom 1 (2e positie)</div>
        <div class="col-md-4 order-md-1">Kolom 2 (1e positie)</div>
        <div class="col-md-4 order-md-3">Kolom 3 (3e positie)</div>
    </div>
</div>
```

In dit voorbeeld is de volgorde van de kolommen aangepast voor middelgrote schermen. Je kunt **offset** gebruiken om ruimte toe te voegen aan de linkerkant van een kolom:

```
<div class="col-md-4 offset-md-4">Gecentreerde kolom</div>
```

Hier wordt de kolom vier posities naar rechts verschoven, zodat deze gecentreerd is.

Stap 21: Dark Mode met Bootstrap

Een andere populaire trend is **Dark Mode**, en Bootstrap maakt het eenvoudig om een donkere versie van je site te implementeren.

Bootstrap heeft ingebouwde dark-modus klassen, zoals **`bg-dark`** en **`text-light`**, die je kunt gebruiken om de donkere thema's toe te passen. Hier is een eenvoudig voorbeeld:

```
<div class="container bg-dark text-light p-5">
    <h1>Dark Mode</h1>
    <p>Dit is een donkere versie van je pagina met lichte tekst.</p>
</div>
```

Voor een meer geavanceerde dark mode, kun je gebruik maken van JavaScript om dynamisch tussen light en dark mode te schakelen op basis van gebruikersvoorkeuren:

```
if (window.matchMedia && window.matchMedia('(prefers-color-scheme:
dark)').matches) {
    document.body.classList.add('bg-dark', 'text-light');
}
```

Stap 22: Aangepaste Stijlen en Branding

Bootstrap is een geweldig startpunt, maar vaak wil je dat een website opvalt en past bij de huisstijl van je bedrijf of project. Dit kun je eenvoudig bereiken door Bootstrap aan te passen met je eigen branding.

Logo's toevoegen aan de Navbar

Als je een logo wilt toevoegen aan je navigatiebalk, gebruik dan een afbeelding in plaats van tekst:

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
    <a class="navbar-brand" href="#">
        <img src="logo.png" alt="Logo" width="30" height="30">
    </a>
    ...
</nav>
```

Kleurenschema aanpassen met CSS

In plaats van SASS kun je eenvoudig Bootstrap's CSS-stijlen overschrijven in een aangepast bestand. Bijvoorbeeld:

```
.btn-primary {
    background-color: #ff5733; /* Je eigen kleur */
    border-color: #ff5733;
}
```

Stap 23: Animaties met Bootstrap

Bootstrap biedt enkele eenvoudige animaties, maar je kunt dit uitbreiden met bibliotheken zoals **Animate.css** of met behulp van CSS-transities.

Om een fade-in effect toe te voegen aan een element, kun je gebruik maken van Bootstrap's `fade` class:

```
<div class="alert alert-success fade show" role="alert">
    Dit bericht verschijnt met een fade-effect!
</div>
```

Voor complexere animaties kun je gebruik maken van **Animate.css** of eigen CSS-animaties:

```
@keyframes slideIn {
  from {
    transform: translateX(-100%);
  }
  to {
    transform: translateX(0);
  }
}

.element {
  animation: slideIn 1s ease-in-out;
}
```

Stap 24: Responsive Web Design met Media Queries

Bootstrap maakt het eenvoudig om responsieve lay-outs te creëren dankzij het ingebouwde grid-systeem en breakpoints, maar soms moet je maatwerk toepassen met **media queries**. Dit helpt om specifieke stijlen toe te passen afhankelijk van het schermformaat.

Een voorbeeld van een aangepaste media query:

```
/* Voor apparaten breder dan 1200px */
@media (min-width: 1200px) {
    .custom-large {
        background-color: #333;
        color: white;
    }
}

/* Voor apparaten kleiner dan 768px */
@media (max-width: 767.98px) {
    .custom-small {
        background-color: #f00;
        color: white;
    }
}
```

In dit voorbeeld worden verschillende stijlen toegepast op basis van de schermgrootte, buiten Bootstrap's standaard breakpoints om.

Stap 25: Performance Optimalisatie met Bootstrap

Hoewel Bootstrap zeer efficiënt is, kunnen extra onnodige stijlen en scripts de prestaties van je website beïnvloeden. Hier zijn enkele tips om je website sneller te maken:

1. Alleen de benodigde Bootstrap-componenten gebruiken

Bootstrap biedt een enorme bibliotheek, maar je hebt niet altijd alles nodig. Je kunt zelf een aangepaste versie van Bootstrap samenstellen en alleen de componenten selecteren die je nodig hebt. Dit doe je door je eigen SCSS-bestanden te maken en alleen de modules te importeren die je daadwerkelijk gebruikt.

Voorbeeld van een aangepaste SCSS-configuratie:

```
@import "bootstrap/scss/functions";
@import "bootstrap/scss/variables";
@import "bootstrap/scss/mixins";
@import "bootstrap/scss/grid";
@import "bootstrap/scss/buttons";
@import "bootstrap/scss/forms";
```

Door alleen deze onderdelen te importeren, bespaar je veel laadtijd en bandbreedte.

2. Optimaliseren van afbeeldingen en media

Gebruik geoptimaliseerde afbeeldingen, bijvoorbeeld met tools zoals **TinyPNG** of **ImageOptim**. Zorg ook dat je afbeeldingen laadt in de juiste resolutie voor de gebruikte schermgrootte, bijvoorbeeld met behulp van het HTML5-attribuut **`srcset`**:

```
<img src="small.jpg" srcset="medium.jpg 768w, large.jpg 1200w"
alt="Responsive afbeelding">
```

3. Lazy Loading toepassen

Voeg **lazy loading** toe aan je afbeeldingen zodat ze pas worden geladen wanneer ze in het zicht komen. Dit verbetert de laadtijd en prestaties van je website:

```
<img src="image.jpg" loading="lazy" alt="Afbeelding met lazy loading">
```

Stap 26: SEO Optimalisatie in combinatie met Bootstrap

Een van de nadelen van frameworks zoals Bootstrap is dat websites vaak generiek aanvoelen, wat hun SEO-score kan beïnvloeden. Hier zijn enkele manieren om ervoor te zorgen dat je website met Bootstrap goed scoort in zoekmachines:

1. Semantische HTML gebruiken

Het is belangrijk om semantische HTML-tags te gebruiken, zoals `<header>`, `<main>`, `<footer>`, `<section>`, etc., om zoekmachines beter te laten begrijpen waar je content over gaat:

```
<header>
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
        <!-- Navigatie-inhoud -->
    </nav>
</header>
<main>
    <section>
```

```
        <h1>Belangrijke Inhoud</h1>
        <p>Hier staat de belangrijkste informatie van de pagina.</p>
    </section>
</main>
<footer>
    <p>Copyright 2024 - Mijn Bedrijf</p>
</footer>
```

2. Gebruik van correcte heading hiërarchie

Zorg ervoor dat je headings logisch geordend zijn, van `<h1>` tot `<h6>`. Dit helpt zoekmachines om de structuur en hiërarchie van je pagina te begrijpen.

3. Verbeterde laadtijd met CDN

Het gebruiken van een CDN voor het laden van Bootstrap zorgt voor snellere laadtijden omdat veel browsers de bestanden van populaire CDNs al in de cache hebben. Dit vermindert de tijd die nodig is om de CSS en JavaScript-bestanden van Bootstrap te downloaden.

4. Gebruik van Metatags

Zorg ervoor dat je relevante metatags toevoegt voor zoekmachines en sociale media:

Hier zijn extra essentiële `meta`-tags die vandaag de dag relevant zijn in moderne HTML en met name belangrijk voor gebruik met Bootstrap en algemene webontwikkeling.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8"> <!-- Specificeert de tekenset als UTF-8 voor internationale tekens -->

  <meta name="viewport" content="width=device-width, initial-scale=1.0"> <!-- Zorgt voor responsief ontwerp door de breedte van de viewport overeen te laten komen met het schermformaat -->

  <meta http-equiv="X-UA-Compatible" content="IE=edge"> <!-- Zorgt ervoor dat IE de laatste rendering-engine gebruikt -->

  <meta name="description" content="Beschrijvende tekst van de pagina-inhoud, belangrijk voor SEO en wanneer de pagina wordt gedeeld."> <!-- Beschrijving van de pagina-inhoud, cruciaal voor zoekmachines en bij delen op social media -->

  <meta name="keywords" content="HTML, CSS, JavaScript, Bootstrap"> <!-- Reeks relevante trefwoorden voor zoekmachines, hoewel minder essentieel geworden voor SEO -->

  <meta name="robots" content="index, follow"> <!-- Geeft zoekmachines aan of de pagina geïndexeerd moet worden en links gevolgd moeten worden -->

  <meta name="author" content="Naam van de auteur of organisatie"> <!-- Geeft aan wie de auteur van de pagina is -->

  <meta property="og:title" content="Titel voor social media"> <!-- Open
```

```
Graph-tag, specificeert de titel die wordt weergegeven bij delen op social
media -->

  <meta property="og:description" content="Korte beschrijving voor social
media"> <!-- Open Graph-tag, beschrijft de inhoud bij delen op social media -
->

  <meta property="og:image" content="https://example.com/image.jpg"> <!--
Open Graph-tag, specificeert de afbeelding voor sociale netwerken bij het
delen van de pagina -->

  <meta property="og:url" content="https://example.com/page-url"> <!-- Open
Graph-tag, specificeert de URL van de pagina bij delen op social media -->

  <meta name="twitter:card" content="summary_large_image"> <!-- Twitter Card-
tag, bepaalt het type weergave bij delen op Twitter, meestal
'summary_large_image' voor een grote afbeelding -->

  <meta name="twitter:title" content="Titel voor Twitter"> <!-- Titel
specifiek voor Twitter-weergave bij delen -->

  <meta name="twitter:description" content="Beschrijving voor Twitter"> <!--
Korte beschrijving specifiek voor Twitter-weergave bij delen -->

  <meta name="twitter:image" content="https://example.com/image.jpg"> <!--
Afbeeldings-URL specifiek voor Twitter-weergave bij delen -->

  <meta name="theme-color" content="#ffffff"> <!-- Kleur voor de adresbalk op
mobiele apparaten in bepaalde browsers zoals Chrome op Android -->

  <meta name="apple-mobile-web-app-capable" content="yes"> <!-- Specificeert
dat de web-app zich als een standalone app gedraagt op iOS wanneer deze wordt
toegevoegd aan het startscherm -->

  <meta name="apple-mobile-web-app-status-bar-style" content="black-
translucent"> <!-- Bepaalt de kleur van de statusbalk voor iOS wanneer de app
fullscreen is -->

  <meta name="format-detection" content="telephone=no"> <!-- Voorkomt dat
telefoonnummers automatisch worden omgezet in klikbare links op mobiele
apparaten -->

  <link rel="icon" href="/favicon.ico" type="image/x-icon"> <!-- Link naar de
favicon, de kleine afbeelding die wordt weergegeven in de tabbladweergave van
de browser -->

  <link rel="apple-touch-icon" href="/apple-touch-icon.png"> <!-- Specifieke
icoon voor iOS-apparaten wanneer deze aan het startscherm worden toegevoegd -
->

</head>
```

Toelichting

- **Responsieve en compatibiliteit**: `charset`, `viewport`, en `X-UA-Compatible` zijn essentieel voor goede weergave en responsiviteit.
- **SEO en sociale media**: Tags zoals `description`, `keywords`, `robots`, `og:title`, en `twitter:card` zijn nuttig voor vindbaarheid en deelbaarheid.
- **Icoon-instellingen**: `icon` en `apple-touch-icon` zorgen voor consistente branding in browser-tabbladen en mobiele apparaten.

Als je een **Single Page Application (SPA)** bouwt met Bootstrap, bijvoorbeeld met een framework zoals **React** of **Vue**, zijn er enkele extra stappen om te overwegen:

- **Dynamische Content Laden**: In een SPA wordt de content vaak dynamisch geladen, zonder dat de pagina opnieuw laadt. Zorg ervoor dat je Bootstrap JavaScript-componenten (zoals modals en tooltips) opnieuw initialiseert als de DOM verandert.
- **Route Gebaseerde Layouts**: Je kunt Bootstrap's grid en componenten gebruiken om dynamische lay-outs te maken, afhankelijk van de huidige route binnen je SPA.

Hier is een voorbeeld met React:

```
import React from 'react';
import { BrowserRouter as Router, Route, Link } from 'react-router-dom';
import { Container, Button } from 'react-bootstrap';

function App() {
  return (
    <Router>
      <Container>
        <nav>
          <Link to="/home"><Button variant="primary">Home</Button></Link>
          <Link to="/about"><Button variant="secondary">About</Button></Link>
        </nav>
        <Route path="/home" component={HomePage} />
        <Route path="/about" component={AboutPage} />
      </Container>
    </Router>
  );
}

function HomePage() {
  return <h1>Welkom op de Homepage!</h1>;
}

function AboutPage() {
  return <h1>Over Ons</h1>;
}

export default App;
```

Stap 26: Progress Bars en Spinners

Progress bars en spinners zijn handig om de gebruiker visuele feedback te geven terwijl content wordt geladen of bewerkingen worden uitgevoerd.

Progress Bar

```
<div class="progress">
    <div class="progress-bar" role="progressbar" style="width: 75%;" aria-
valuenow="75" aria-valuemin="0" aria-valuemax="100">75%</div>
</div>
```

Spinner

```
<div class="spinner-border text-primary" role="status">
    <span class="visually-hidden">Laden...</span>
</div>
```

Stap 27: Testen van je Bootstrap Layout

Het is belangrijk om je layout op verschillende schermformaten en apparaten te testen. Gebruik tools zoals **Chrome DevTools** om je website te testen op verschillende schermgroottes en resoluties.

Cross-browser Testing Tools

Zorg er ook voor dat je website goed werkt in verschillende browsers, zoals Chrome, Firefox, Safari, en Edge. Gebruik tools zoals **BrowserStack** of **LambdaTest** om je website te testen in verschillende omgevingen.

Stap 28: Best Practices voor Productie

Tot slot, als je klaar bent om je website live te zetten, zijn er enkele best practices die je moet volgen om ervoor te zorgen dat je site veilig en snel is:

- **CSS en JS Minificatie**: Zorg ervoor dat je CSS en JavaScript-bestanden geminimaliseerd zijn om de bestandsgrootte te verkleinen.
- **Caching**: Gebruik browser caching en zorg ervoor dat je server de juiste headers uitzendt om te voorkomen dat gebruikers onnodig bestanden opnieuw moeten downloaden.
- **HTTPS**: Zorg ervoor dat je website via HTTPS wordt geladen om de beveiliging van je gebruikersgegevens te waarborgen.