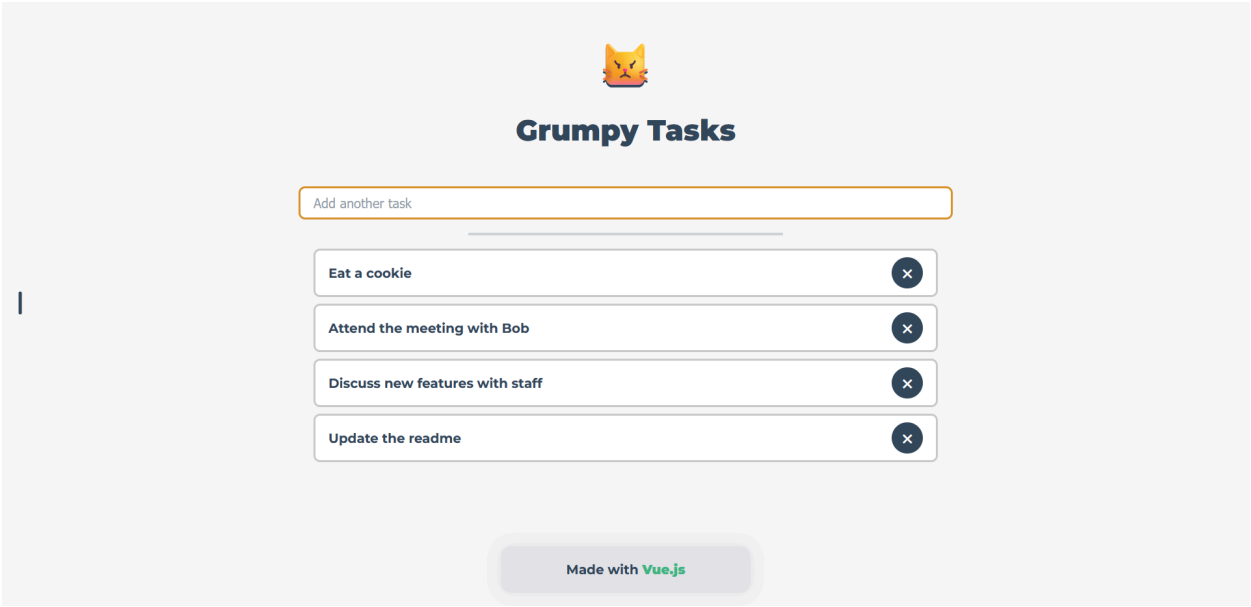
FELIX NAGY
22SDB_A
SOFTWARE DEVELOPER
Aan Arjen de Haan & Niels Vermeulen

VERDIEPING VUE

Een vergelijking tussen Vue, React en Angular



Inhoudsopgave

Inleiding

Vue.js, React en Angular zijn de meest gevraagde frameworks om te kennen en te beheersen. Deze frameworks hebben de meeste vacatures en kansen tegenwoordig. Mijn doel met dit verslag is om deze veelgevraagde JavaScript-frameworks te onderzoeken, en uiteindelijk een framework te kiezen waarmee ik iets ga ontwikkelen; ik zal een vergelijking maken tussen Vue, React en Angular, zodat ik beter voorbereid ben op mijn toekomst.

Ik zal elk framework kort introduceren en vervolgens de voor- en nadelen ervan analyseren.

Vue

Vue, uitgebracht in 2014, is een JavaScript-framework dat de laatste jaren steeds populairder wordt. Het staat bekend om de eenvoudige learning curve, omdat het erg makkelijk is om te leren voor beginners. Je kan heel gemakkelijk een nieuw project aanmaken met Vite en meteen aan de slag gaan met het maken van een website. In vergelijking met React, is Vue makkelijker om te leren, omdat React geavanceerdere JavaScript code en concepten gebruikt. De officiële documentatie van Vue is duidelijk en overzichtelijk, er zijn daarnaast ook officiële video's van de developers, maar ook op YouTube vind je erg veel uitleg over het framework.

Voordelen Vue

- Eenvoudige leercurve, ideaal voor beginners.
- Je kan het toevoegen aan bestaande projecten.
- Actieve community en goede documentatie.
- Het is heel snel door de virtuele DOM.

Nadelen Vue

- Minder geschikt voor grote, complexe projecten.
- Vue is een kleiner systeem met minder plug-ins en tools voor de ontwikkelaar, in vergelijking met React of AngularJS.

React

React is in 2014 ontwikkeld door Facebook en is een van de populairste JavaScript libraries voor het programmeren van user interfaces. Het is gebaseerd op componenten, en maakt het makkelijk voor developers om snel websites te bouwen.

Voordelen React

- Grote community-ondersteuning.
- Groot systeem met veel libraries, plug-ins en tools.
- Geschikt voor zowel kleine als grote projecten.

Nadelen React

- Moeilijk voor beginners.
- React is een library, geen volledig framework, wat dus meer configuratie vereist.

Angular

Angular, ontwikkeld door Google in 2010, is een uitgebreid framework voor het bouwen van web- en mobiele applicaties. Het is wel een van de oudere frameworks voor JavaScript, dit is te zien aan de populariteit; React en Vue worden tegenwoordig veel meer gebruikt. Aan de website is al meteen te zien dat het niet modern meer is.

Voordelen Angular

- Het heeft ingebouwde functies voor routing, formulieren en validatie.
- Goed voor grote en complexe projecten.
- Gebruikt TypeScript.

Nadelen Angular

- Moeilijk om te leren, vanwege de vele ingebouwde functies.
- Outdated en niet modern genoeg voor de meeste developers.

Onderzoek

De 3 frameworks heb ik op de vorige pagina's uitgelegd, en alle voor- en nadelen beschreven. Het is nu dus tijd om echt te onderzoeken met welk framework ik aan de slag ga. Het is belangrijk om te weten waar ik mijn informatie vandaan moet halen. Ik heb daarom gekeken naar de documentatie van elk framework, en andere betrouwbare websites waaronder Medium.

Veel developers hebben natuurlijk al onderzoek gedaan over de frameworks; *React VS VueJS*, *Angular or React*, deze posts zijn overal te zien en ook erg leerzaam. Ik heb daarom ook veel van andere onderzoeken geleerd, in plaats van alles zelf uit te vogelen. Het leek mij namelijk logisch—vooral op het gebied van programmeren—om te leren van anderen.

Tijdens het onderzoeken kwam ik er langzaam achter dat Angular misschien niet de beste keuze is om mee te werken: React en Vue zijn moderner en meer gevraagd bij bedrijven. Dit komt vooral doordat Angular een ouder framework is. De nieuwe alternatieven zijn de standaard geworden omdat ze kleiner zijn en daardoor beter presteren. Ze gebruiken andere manieren voor componenten en kunnen alles veel sneller laden, Angular heeft hier niet op gelet en is daarom verouderd. Een voorbeeld hiervan is de virtual DOM die React en Vue gebruiken en Angular niet.

Later in het onderzoek kwam ik er ook achter dat Vue wat makkelijker is voor beginners dan React. De documentatie van Vue is heel duidelijk; het beantwoordt elke vraag die je maar hebt en bevat meerdere officiële video's van het bedrijf. Het wordt wel gezegd dat je met React wel meer kan bereiken, maar omdat ik nog nooit met een JavaScript framework/library gewerkt heb, leek VueJS mij een betere keuze.

Leerdoelen

Na grondig onderzoek en vergelijking van Vue, React en Angular, heb ik besloten om met Vue te gaan werken voor mijn software project. Overal is te lezen dat Vue heel eenvoudig is voor beginners. Hoewel React en Angular ook sterke opties zijn, was Vue de beste keuze voor mijn project. Als eindproduct ben ik van plan om een task manager/todo list te maken, met VueJS. Met behulp van componenten ga ik de website samenstellen.

De leerdoelen zijn:

- Vue beheersen en kunnen gebruiken om simpele websites te maken.
- Weten hoe je het beste met componenten werkt.
- Weten hoe je variabelen en andere data tussen componenten moet delen.

Stappenplan

Om mijn leerdoelen te behalen en een goede task manager te maken, moet ik bepaalde stappen gaan nemen. Hieronder valt het maken van opdrachten, documentaties lezen, tutorials kijken en in het algemeen veel bezig zijn met Vue.

De stappen die ik ga doornemen zijn:

1. **Leer de basis van Vue**, lees de documentatie goed door en zorg dat je weet hoe Vue werkt. Weet dus hoe je een project aanmaakt, hoe componenten werken en hoe de mappenstructuur eruitziet.

2. **Kort brainstormen over de task** manager, wat wordt het project? Hoe gaat het eruit zien, wat zijn de functionaliteiten? Meestal kan je een task toevoegen, afchecken en verwijderen. Een ding is zeker, het moet *mobile first* gemaakt worden.

3. **Maak de basis met een tutorial** op YouTube. Hierdoor zal ik begrijpen hoe Vue werkt en hoe de componenten gebruikt worden, je moet toch ergens beginnen.

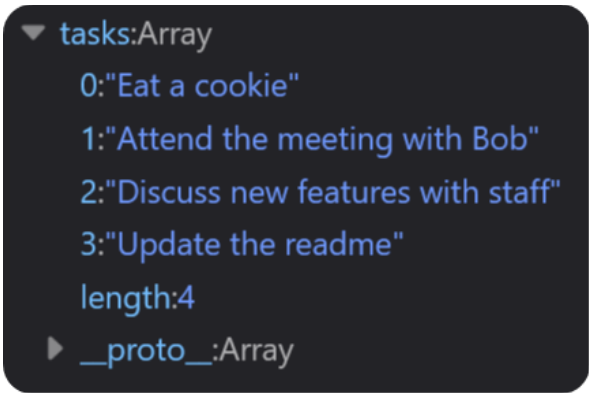
4. **Unieke eigenschappen** toevoegen aan de task manager. Geef het een leuke naam of andere unieke eigenschap. Het moet ook niet meer lijken op het project van de tutorial.

5. **Breid het project uit** door andere functies toe te voegen aan de website. Als het goed is, bevat het nu de basisfunctionaliteit. Voeg eventueel andere functies zoals een navbar en responsive design.

Met deze stappen zal ik mijn leerdoelen behalen en een goede task manager maken. Het is belangrijk om een stappenplan te maken, zodat je weet wat je moet doen en hoe je het moet doen. Het is ook belangrijk om een stappenplan te maken, zodat je niet te veel tijd kwijt bent aan het project. Als je geen stappenplan maakt, kan je te veel tijd kwijt zijn aan het project en dat is niet de bedoeling.

Grumpy Tasks

Het product is af, een simpele task manager die je kan gebruiken om je taken te beheren. Je kan taken toevoegen, verwijderen en afvinken.



De taken worden opgeslagen in de local storage van de browser in de vorm van een array. Dit zorgt ervoor dat de taken niet verloren gaan als je de pagina ververst. Wanneer je een item verwijdert, wordt de index dynamisch bijgewerkt.

Deze waarden worden weergegeven op de pagina in een overzichtelijke lijst. De input heeft ook standaard validatie, waaronder de lengte en spaties. Je kunt een taak verwijderen of afchecken, het wordt dan ook verwijderd uit de local storage en de ID wordt dynamisch aangepast.

Reflectie

Hoe is dit project gelopen en wat heb ik ervan geleerd?

Samengevat, heb ik onwijs veel geleerd over Vue. Ik weet nu hoe ik beter en sneller websites kan maken. De componenten waar Vue gebruik van maakt zijn heel overzichtelijk en handig om te gebruiken. In de documentatie staat alles ook perfect beschreven.

Al mijn vorige projecten die veel JS gebruikten, waren best onduidelijk. De bestanden waren veel te lang en niet duidelijk. Bijvoorbeeld mijn portfolio, waarvan het *app.js* bestand bijna 300 pagina's lang is. Alle dynamische code zoals de pagina's zelf, afbeeldingen en effecten worden daar geschreven. Ik probeerde het door middel van comments duidelijk te maken, maar op een gegeven moment was er simpelweg teveel code. Het is heel groot en onhandig op die manier, daar is Vue dus een goede oplossing voor.

Ik ben van plan om mijn portfolio te herschrijven met VueJS. Met componenten ga ik alle onderdelen van de website indelen, zoals de navbar, main secties, tekstvakken en de footer. Door het maken van deze opdracht ben ik daar nu bereid voor, ik ben geen expert met Vue, maar beheers het wel.

In principe waren er weinig problemen met het maken van de website. Tijdens het ontwikkelen krijg je veel foutmeldingen, maar die waren deze keer gelukkig makkelijk op te lossen. Het was erg wennen om steeds componenten aan te maken en bepaalde data te delen en te exporteren, maar na het een paar keer te doen was het erg gemakkelijk.

Het is begrijpelijk waarom veel bedrijven—die gericht zijn op front-end ontwikkeling—veel om Vue vragen. Websites ontwikkelen hiermee is heel simpel en overzichtelijk en daar houd ik van. Ik wil altijd op een overzichtelijke manier programmeren, comments en functies goed gebruiken etc, dit is nu nog makkelijker door VueJS!

Logboek

Waar heb ik specifiek aan gewerkt elke week?

Op 4 oktober 2023 heb ik het eerste verslag ingeleverd, waarbij ik onderzocht heb welke software veel gevraagd wordt (via Indeed). Sindsdien is er veel gebeurd, laten we even een terugblik nemen:

Week	**Beschrijving**
40	Inleveren van het eerste verslag.
41-42	Het project is aangemaakt op GitHub, ik heb met behulp van een video het gebruik van componenten getest. Er waren hier veel problemen.
43-45	Ik heb de website een naam gegeven: *Grumpy Tasks*. Het heeft een kleine easteregg, er komt namelijk een dynamische notificatie als je een bepaald aantal taken hebt toegevoegd. Dit is natuurlijk in de vorm van een component en heel duidelijk geschreven.
46-49	Deze periode was ik een tijd ziek, ik was helaas afwezig maar heb toen thuis gewerkt. Er zat hier zelfs een vakantie van een week tussen. Er was niet veel gebeurd rond deze weken, ik weet niet precies, maar kort na deze periode ging ik weer aan de slag.
50-52	Na het ziek zijn ging ik meteen weer verder, ik was het project aan het afronden en begon met het schrijven van het eindverslag.
01-05	Het schrijven van het verslag heeft de grootste prioriteit. De website heeft wat optimalisaties gekregen, zoals de sidebar. Je kan nu elke taak verwijderen, maar in plaats van een simpele knop ergens in een hoekje verstopt, heb ik een sidebar component gemaakt die mooi geopend kan worden als je er met de muis overheen hovert. Afronden van het project dus.

Vacatures

De vacatures die ik op Indeed gevonden heb, vereisten allemaal één of meerdere frameworks.

Basic-Fit

Aeroscan B.V.

MedApp

Bronnenlijst

Deze websites hebben mij geholpen met het verslag.

[Browserstack]

[Radixweb]

Naast algemene websites voor informatie die ik kon gebruiken voor de vergelijking, heb ik natuurlijk ook gekeken naar de documentatie van de frameworks. Hierdoor kreeg ik een beter beeld over het framework/library, en hoe je er mee te werk gaat.

[Vue Docs]

[React Docs]

[Angular Docs]