



Hilmar van der Veen

Zandvoort

25-05-1987

+31 6 8014 9947

hilmar.van.der.veen@the-future-group.com

Frontend Angular | React | C# .NET



SAMENVATTING



Ik ben een hands-on front-end developer met meer dan zeven jaar ervaring in het realiseren van schaalbare, toegankelijke en onderhoudsvriendelijke applicaties. Met een sterke focus op React en TypeScript heb ik de afgelopen jaren uiteenlopende toepassingen opgeleverd die zowel technisch robuust als intuïtief in gebruik zijn.


Ik werk doelgericht en gestructureerd, met oog voor kwaliteit én resultaat. Van componentarchitectuur tot integratie met backend-API's: ik neem verantwoordelijkheid voor technische keuzes en vertaal functionele wensen naar werkende oplossingen. Daarbij houd ik rekening met security- en privacyprincipes vanaf het ontwerp.


In mijn rol als senior of lead ontwikkelaar coach ik collega's, voer ik code reviews uit en bewaak ik consistentie binnen het team. Ik draag actief bij aan het verbeteren van CI/CD-processen, UX-afstemming en WCAG-conforme implementaties. Dankzij mijn fullstack achtergrond in .NET Core heb ik goed zicht op backendlogica, waardoor ik frontends bouw die naadloos aansluiten op de onderliggende systemen.

RELEVANTE TECHNIKEN


	Medior	Senior	Lead		Medior	Senior	Lead
Angular 2 - 18	●	●	●	Microsoft Identity	●	●	○
React 16 +	●	●	●	Entity Framework	●	●	○
Vue 3+	●	○	○	CQRS	●	●	○
Typescript	●	●	●	MassTransit	●	●	○
JavaScript (ES5/ES6/ESNext)	●	●	●	RabbitMQ	●	●	○
HTML 5	●	●	○	REST / OpenAPI / SOAP	●	●	○
CSS3, LESS, SASS	●	●	○	OAUTH 2	●	●	○
Tailwind CSS	●	●	○	OpenID Connect	●	●	○
Styled-components	●	●	○	ASP.NET Web API	●	●	●
Bootstrap / Material UI	●	●	○	.NET MVC	●	●	●
Stencil / Web Components	●	●	○	FIGMA Sketch Wireframe	●	●	○
RxJS	●	●	●	WCAG 2.0 (Toegankelijkheid)	●	●	○
Redux / Thunk	●	●	●	Design Systems	●	●	●
Zustand	●	●	●	Component-based design	●	●	●
React Query	●	●	●	Storybook	●	●	●
NextJS Vite	●	●	●	Cypress	●	●	○
BEM (CSS architectuur)	●	●	○	Playwright	●	●	○
SVG-animaties	●	●	○	Jest	●	●	○
C# / .NET Core (3 5 6)	●	●	●	Karma	●	●	○
ASP.NET Core SignalR	●	●	○	xUnit junit	●	●	○
Node.js	●	●	●	SonarQube	●	●	○
Azure DevOps	●	●	○	ESLint	●	●	○
Azure Pipelines (CI / CD)	●	●	○	Prettier	●	●	○
Jenkins	●	○	○	StyleCop	●	●	○
Tekton	●	○	○	SQL Server (MSSQL)	●	●	○
ArgoCD	●	○	○	LINQ	●	●	○
Git GitHub GitLab BitBucket	●	●	○	Neo4J (GraphDB)	●	○	○
Docker	●	○	○	Azure Cosmos DB	●	●	○

Kubernetes	●	○	○	JSON	●	●	○
MAVEN	●	○	○	Azure (App Services, Cloud, DevOps, Pipelines)	●	●	○
JIRA	●	●	●	AWS (Lambda S3 Bucket)	●	○	○
Confluence	●	●	●	NX (Mono repo)	●	○	○
LDAP	●	●	○	NPM	●	○	○
JWT	●	●	○	JAVA 8	●	○	○
SPLUNK	●	○	○	Headless CMS	●	●	○
Lighthouse / aXe	●	●	○	GitHub Actions	●	○	○

OPLEIDINGEN 			
Master Physics (AMEP)	UVA	<input checked="" type="checkbox"/>	2012 -2014
BSc Physics and Astronomy	UVA	<input checked="" type="checkbox"/>	2012 afgerond

TRAININGEN 			
React Angular Typescript python C# SQL	Udemy	<input checked="" type="checkbox"/>	2020
Scrum master C# CSS HTML	Young Capital	<input checked="" type="checkbox"/>	2016
Talen:			
Nederlands (Vloeiend)			
Engels (Vloeiend)			

RELEVANTE WERKERVARING

Senior Frontend Developer / Technisch Lead	 Belastingdienst
Januari 2023 - December 2024	
Klant: Belastingdienst	
Werklocatie: thuis en 2 dagen op locatie in Apeldoorn	
Voertaal: Nederlands	

Project 1: BOLD Web Components – Design System Governance & Frameworkintegratie

De Belastingdienst werkte met het centrale BOLD design system (Stencil Web Components), maar teams ondervonden structurele problemen bij het gebruik ervan in Angular-, React- en Vue-omgevingen. De documentatie was ontoereikend en leidde tot inconsistent gedrag.

Ik nam het voortouw om deze problemen structureel op te lossen. Ik testte alle componenten per framework, schreef heldere documentatie in Confluence met platform-specifieke voorbeelden en edge cases, en ontwikkelde wrappers en polyfills voor compatibiliteit. Voor ontbrekende functionaliteit leverde ik tijdelijke bridge-components, inclusief gecontroleerde uitfasering.

Door mijn inzet konden front-end teams sneller en foutloos integreren met BOLD. De adoptie verbeterde, de toegankelijkheid (WCAG) werd geborgd en componentgedrag werd consistent over teams en frameworks heen.

Project 2: CANVAS – Visuele Formuliereneditor (Greenfield Low-code Project)

Redacteuren bij de Belastingdienst moesten zonder programmeerkennis complexe formulierflows kunnen opbouwen. Er was behoefte aan een intuïtieve low-code editor met drag-and-drop-functionaliteit, conditionele logica en versiebeheer.

Ik ontwierp en bouwde de frontend van de editor op basis van een componentgebaseerde architectuur. Gebruikers konden formulieronderdelen samenstellen via een canvas, instellen via configuratiepanelen en testen in een tryout-weergave. Validaties, afhankelijkheden, meertaligheid en toegankelijkheid waren volledig via de UI instelbaar.

Daarnaast breidde ik backend-API's uit (Java, MySQL) zodat wijzigingen correct werden opgeslagen en reproduceerbaar waren. Dit resulteerde in een krachtig redactietool waarmee redacteuren zelfstandig toegankelijke formulieren konden publiceren, zonder afhankelijkheid van ontwikkelaars.

Project 3: Ondernemersportaal – Data-driven Frontend Architectuur

Voor het Ondernemersportaal ontwierp ik een datagedreven frontendarchitectuur die het mogelijk maakte om pagina's sneller, consistent en zonder code duplicatie op te bouwen. De wens was om meerdere aangiftepagina's, zoals btw, loonheffing, dividendbelasting en vliegbelasting, modulair te genereren op basis van backendconfiguratie, met herbruikbare componenten en flexibele gedragsaanpassing.

Ik ontwikkelde een set generieke Angular-componenten waarvan de logica en weergave werden aangestuurd via injectable services en configuratiebestanden. Invoervelden, infoblokken, validaties en hulpteksten pasten hun gedrag dynamisch aan op basis van waarden uit andere componenten of data uit de backend. De architectuur maakte gebruik van conditionele zichtbaarheid, meertalige labels en validatieregels die via JSON-schema's werden ingeladen.


Daarnaast implementeerde ik frontendfilters die bestonden uit dropdowns, zoekvelden en chips, inclusief verwijderbare chips voor actieve filters. Voor tabellen met beperkte datasets schreef ik een wrappercomponent die client-side sortering mogelijk maakte, zonder dat de backend opnieuw aangesproken hoefde te worden. Deze wrapper hield rekening met kolomtypes, toegankelijkheidseisen en sorteerlogica volgens de specificaties van het BOLD design system.


Door deze aanpak konden nieuwe pagina's met minimale effort worden uitgerold, terwijl toegankelijkheid (WCAG), herbruikbaarheid en consistentie geborgd bleven. De architectuur bleek schaalbaar, onderhoudsvriendelijk en zeer geschikt voor hergebruik in soortgelijke portalen binnen de organisatie.

Technieken & Tools	Werkwijze
HTML5, CSS3, JavaScript (ES6+), TypeScript, Angular, React, Web Components, CSS, Stencil, WCAG 2.0 (toegankelijkheid), Playwright, Cypress, REST, OpenAPI 3.0, JWT, Java 8, Maven, Gitlab, Git, NPM, Jenkins, Tekton, Argo CD, SPLUNK, Docker, SonarQube, ESLint, Storybook, Confluence, Jira, MySQL, OWASP, OATH 2	Scrum, Kanban, Nauwe samenwerking met UX, PO, backend-ontwikkelaars en DevOps. Actieve rol in refinement, teamcoaching, code reviews en CI/CD-verbeteringen.

Frontend Developer / Design System Specialist	
Juli 2022 - Januari 2023	
Klant: Nationale Postcode Loterij	
Werklocatie: thuis en 2 dagen op locatie in Amsterdam	
Voertaal: Engels	
Project: Internationaal Design System voor Marketingcampagnes	
<p>In een greenfield-omgeving werkte ik aan de opbouw van een schaalbaar design system dat wereldwijd herbruikbaar moest zijn voor alle marketingcampagnes van de Nationale Postcode Loterij. Het doel: een uniforme frontendsbasis neerzetten met focus op toegankelijkheid, snelheid en eenvoudige aanpasbaarheid.</p> <p>Ik ontwierp en ontwikkelde modulaire UI-componenten in React, TypeScript en Next.js, gestyled met TailwindCSS en SCSS voor maximale flexibiliteit. Componenten waren eenvoudig te configureren op kleur, typografie, gedrag en lay-out, en ik borg volledige WCAG 2.0-compliance. Ik implementeerde SVG-animaties en een headless CMS-integratie voor dynamische content. Voor performanceoptimalisatie gebruikte ik server-side rendering (SSR) met Next.js, gericht op het minimaliseren van laadtijden.</p> <p>Aan de DevOps-kant zette ik GitLab CI/CD-pijplijnen op, en bouwde ik microservices via AWS Lambda en REST API's (OpenAPI 3.0). Dit maakte het mogelijk om snel te deployen per campagne, met korte doorlooptijden van concept tot livegang. Het design system werd uitgerold over meerdere markten en campagnes en leverde een consistente gebruikerservaring op met hoge performance en volledige toegankelijkheid.</p>	
Technieken & Tools	Werkwijze
React, TypeScript, Next.js, TailwindCSS, SCSS, SVG Animations, Headless CMS, GitLab CI, AWS Lambda, REST API's (OpenAPI 3.0), SSR, HTML5, JavaScript (ES6+), NPM, Git, Docker,	Scrum (Agile), Intensieve samenwerking met UX-designers, DevOps en marketingteams.


Cypress, Playwright, ESLint, Prettier, Jira, Confluence, S3-bucket, SPLUNK, SonarQube, Zustand, React-query, OWASP, JWT, OATH 2	
---	--


Lead Frontend Developer	
Oktober 2020 - Juli 2022	
Klant: Athlon	
Werklocatie: thuis	
Voertaal: Engels	
Project: Migratie & Herbouw van Legacy-applicatie naar Moderne Angular Stack	
De bestaande Angular 1.6-applicatie voldeed niet meer aan de hedendaagse eisen op het gebied van performance, onderhoudbaarheid en toegankelijkheid. Mijn opdracht was om zowel de technische architectuur als de gebruikersinterface volledig te vernieuwen, met behoud van bestaande functionaliteit, maar binnen een moderne Angular-stack.	
Ik leidde de migratie naar Angular 12 en ontwierp een nieuwe frontendarchitectuur gebaseerd op RxJS en reactive programming. Hierbij stond schaalbaarheid, WCAG-compliance en performance-optimalisatie centraal. Ik herstructureerde de UI op basis van moderne stylingprincipes en begeleidde junior- en medior-developers bij het schrijven van kwalitatieve, goed geteste frontendcode. Daarnaast integreerde ik de frontend met meerdere backend REST API's via OpenAPI 3.0.	
Ik ontwikkelde een flexibele componentbibliotheek en bouwde een generiek selfservice-platform met conditionele logica en validatie, gericht op low-code principes. Voor dashboards realiseerde ik dynamische datavisualisaties, met responsive grafieken en sorteerbare tabellen. In samenwerking met UX-designers vertaalde ik wireframes en gebruikersfeedback naar een consistente, toegankelijke frontend.	
Het eindresultaat was een toekomstbestendige applicatie, gebouwd op moderne standaarden, volledig WCAG-conform en eenvoudig uitbreidbaar. De herbouwde applicatie maakte snellere interactie mogelijk, verlaagde de foutkans en resulteerde in minder druk op support dankzij duidelijke, goed ontworpen selfservice-functionaliteit.	
Technieken & Tools	Werkwijze
Angular 12, TypeScript, RxJS, TailwindCSS, SCSS, JavaScript (ES6+), OpenAPI 3.0, REST, WCAG 2.0, HTML5, Cypress, Jest, Azure, Jenkins, Docker, GitHub, Git, SonarQube, Java 8, ESLint, Prettier, Jira, Confluence, KeyCloak, Bootstrap UI, OWASP, JWT, OATH 2	Scrum (Agile) – Leidend in technische beslissingen, coaching en refinement. Nauwe samenwerking met UX en backend.


Frontend Developer / UI Coach	
Augustus 2020 - Oktober 2020	
Klant: Randstad	
Werklocatie: Amsterdam	
Voertaal: Engels	
Project: Rebranding en Componentontwikkeling voor Randstad- en Yacht-websites	
Randstad en Yacht ondergingen een visuele en technische herpositionering waarbij bestaande websites vernieuwd en uitgebreid moesten worden met toegankelijke, consistente UI-componenten conform de nieuwe merkidentiteit.	
Ik werkte aan de herontwikkeling van front-endcomponenten in vanilla JavaScript, Angular en React. Hiervoor gebruikte ik SCSS en de BEM-methodiek om structuur en herbruikbaarheid te waarborgen. Ik ontwikkelde modulaire componenten met focus op toegankelijkheid (WCAG 2.0) en begeleidde collega's in het toepassen van Angular best practices en componentgedreven ontwikkelprincipes. Mijn rol als UI Coach hielp teams consistentie en kwaliteit verhogen, met behoud van flexibiliteit tussen platforms.	
Daarnaast ontwikkelde ik stylingcomponenten voor het Hippo Bloomreach headless CMS en testte ik alle visuele en functionele elementen op toegankelijkheid en responsive gedrag. Ik werkte nauw samen met UX-designers en business stakeholders om pagina's en flows op elkaar af te stemmen en ervoor te zorgen dat de nieuwe componenten naadloos aansloten op de vernieuwde merkidentiteit.	

Het resultaat was een uniforme, schaalbare UI-bibliotheek die de visuele kwaliteit, toegankelijkheid en performance van de Randstad- en Yacht-platformen merkbaar verbeterde. Door mijn coaching en technische begeleiding nam de ontwikkelsnelheid toe en werd de kwaliteit van releases stabiel.

Technieken & Tools	Werkwijze
React, Angular, SCSS, JavaScript (vanilla), BEM, WCAG 2.0, component-based design, Hippo Bloomreach (headless CMS), FTL, Maven, GitLab, Jenkins, Git, SonarQube, ESLint, Prettier, Bitbucket, Confluence, Github, JAVA 12, Spring Boot	Scrum (Agile), Coachen van teams, documenteren van componentrichtlijnen en kwaliteitsborging in CI/CD-omgeving.

Fullstack Developer / Frontend-Lead (Angular & .NET Core)	
Augustus 2019 - Maart 2020	
Klant: Omniplan	
Werklocatie: Amsterdam	
Voertaal: Nederlands	
	
Project: Modulaire Authenticatieomgeving & Low-code Frontend	
<p>Omniplan vernieuwde haar platformarchitectuur om veilige integratie met externe partijen mogelijk te maken. Zowel de frontend als backend werd volledig herbouwd. Ik was verantwoordelijk voor de implementatie van een nieuwe Angular 7 frontend in combinatie met een herontworpen .NET Core backend, inclusief architectuur en een flexibel autorisatie- en configuratiesysteem.</p> <p>Ik ontwikkelde een modulaire frontend met HTML, CSS, TypeScript en RxJS, gericht op herbruikbaarheid, toegankelijkheid en performance. Voor de backend ontwierp ik een op Microsoft Identity en JWT gebaseerde autorisatiestructuur, waarmee externe organisaties veilig toegang konden krijgen tot REST API's van Omniplan. Deze hybride architectuur maakte veilige authenticatie mogelijk tussen partijen zonder directe systeemkoppeling.</p> <p>Daarnaast ontwierp ik een nieuwe microservicesarchitectuur op basis van Azure App Services en Docker/Kubernetes, en herstructureerde ik het datamodel in Entity Framework op MSSQL voor betere schaalbaarheid. Aan de frontenzijde realiseerde ik een dynamisch configureerbare interface, waarin validaties, permissies en invoervelden direct gestuurd werden via backendconfiguraties – vergelijkbaar met een low-code omgeving.</p> <p>Het resultaat was een schaalbaar, veilig en onderhoudsvriendelijk platform dat eenvoudig kon worden aangepast aan nieuwe functionele wensen, met minimale ontwikkelinspanning en maximale flexibiliteit in externe integraties.</p>	
Technieken & Tools	Werkwijze
Angular 7, TypeScript, RxJS, HTML5, CSS3, C#, ASP.NET Core, Microsoft Identity, JWT, REST API, Entity Framework, MSSQL, configuration-based UI, Docker, Kubernetes, Azure App Services, Azure Cloud Services, DevOps, TFS, Git, Confluence, xUnit, SonarQube, ESLint, Prettier, OWASP, OATH 2	Scrum (Agile), Verantwoordelijk voor technische architectuur, integratieaanpak en fullstack implementatie.

Full stack software developer (Angular & .NET Core)	
April 2019 – Juli 2019	
Klant: Ortec	
Werklocatie: Zoetermeer	
Voertaal: Engels	
	
Project: Logistiek Portaal voor FedEx – Route-inzicht en Optimalisatie	
<p>Voor FedEx ontwikkelde Ortec een portaal waarmee logistieke medewerkers realtime inzicht kregen in routes, ladingen en bezorglocaties. Mijn rol was het realiseren van een toegankelijke, performante frontend in Angular, gericht op validatie, schaalbaarheid en foutbestendige datavisualisatie – en de integratie met een .NET Core backendarchitectuur.</p> <p>Ik bouwde Angular 7-componenten met Angular Material, TypeScript, HTML, SCSS en RxJS. De UI bevatte een interactieve datatabel met sortering, filters en ondersteuning voor route-optimalisatie. Ik implementeerde modals, ingebouwde validatie en foutafhandeling waarmee gebruikers direct visuele feedback kregen bij invoerfouten. Hiermee werd de foutgevoeligheid sterk verlaagd. Daarnaast integreerde ik RxJS-observables voor realtime updates, en verzorgde ik de volledige connectie met REST API's op basis van OpenAPI en .NET Core microservices.</p>	

Ik herstructureerde het componentmodel op basis van low-code en configuration-based principes, waarmee visuele en functionele elementen herbruikbaar en flexibel aanpasbaar werden. De deployment verliep via Docker naar Azure Cloud Services, volledig CI/CD-gedreven in een Scrum DevOps-team.	
Het eindresultaat was een schaalbaar portaal waarmee FedEx-medewerkers directe controle kregen over routes, met een intuïtieve interface die invoerfouten minimaliseerde en de interactie vereenvoudigde. De oplossing bleek robuust genoeg om toekomstige uitbreiding en externe integraties eenvoudig te ondersteunen.	
Technieken & Tools	Werkwijze
Angular 7, Angular Material, TypeScript, RxJS, SCSS, HTML5, REST API, OpenAPI, .NET Core, C#, Azure Cloud, Azure Cosmos DB, Docker, Microservices, data-driven UI, xUnit, StyleCop, SonarQube, ESLint, Prettier, Git, Gitlab, Jira, Confluence, OWASP, JWT, OATH 2	Werken in DevOps-team met CI/CD, focus op foutbestendigheid, schaalbaarheid en UX-consistentie.
Fullstack Developer (ReactJS, .NET Core, Neo4J)	
Maart 2018 – April 2019	
Bluefield smart access	
Werklocatie: Maarsen Utrecht	
Voertaal: Nederlands	
Project 1: Data-driven Webapplicatie met Cyclusvisualisatie Binnen Bluefield werkte ik aan een webapplicatie die complexe cyclische processen inzichtelijk moest maken voor eindgebruikers. Deze processen werden aangeleverd via CSV-bestanden en opgeslagen in een Neo4J graph-database. De klant wilde de status van deze cycli visueel kunnen volgen, inclusief foutdetectie en herstelopties. Mijn taak was het ontwerpen en ontwikkelen van een visueel dashboard waarin cycli overzichtelijk weergegeven werden, met interactieve elementen en een heldere statusweergave per cyclus. Ik bouwde de frontend in ReactJS, met gebruik van modern JavaScript (ES6+), HTML en CSS. De applicatie was gekoppeld aan een .NET Web API die de data uit Neo4J ontsloot. Ik ontwierp een tegelgebaseerde gebruikersinterface waarin elke cyclus een kleurgecodeerde status kreeg, zoals "succesvol", "mislukt" of "onbekend". Via een interactieve detailweergave konden gebruikers doorklikken naar uitgebreide informatie per cyclus, waaronder data en aantallen. Bij foutieve verwerking bood de interface de mogelijkheid om de betreffende cyclus opnieuw op te starten. Alle logica werd ondersteund door een schaalbare Redux/Redux-Thunk architectuur, gericht op herbruikbaarheid en onderhoudsgemak. De applicatie gaf gebruikers real-time inzicht in duizenden processen en maakte snelle bijsturing mogelijk bij fouten. Dit zorgde voor meer grip op dataverwerking, minder handmatige tussenkomst en een betrouwbaardere rapportagestroom.	
Project 2: Tommy Hilfiger – Toegangscontrole & Parkeerbeheer Tommy Hilfiger had behoefte aan een intern systeem om parkeerplaatsen te beheren en bezoekers gecontroleerd toegang te verlenen tot het terrein. Het systeem moest gebruikers in staat stellen reserveringen te maken en toegangsbewijzen te printen via een eenvoudige interface. Mijn opdracht was om een gebruiksvriendelijke webapplicatie te ontwikkelen waarin medewerkers parkeerplaatsen konden reserveren en bijbehorende tickets konden genereren en afdrucken. Ik ontwikkelde een dynamische kalenderinterface in React, waarmee gebruikers per dag parkeerplaatsen konden inplannen. De applicatie was ontworpen met het oog op gebruiksgemak en minimale leercurve. Ik zorgde voor een consistente visuele stijl en bouwde herbruikbare UI-componenten die eenvoudig geïntegreerd konden worden in andere delen van het platform. Het systeem werd met succes in gebruik genomen binnen de organisatie. Medewerkers konden snel en zelfstandig parkeerplaatsen reserveren en tickets printen, wat de administratieve last verminderde en de toegangsstroom aanzienlijk verbeterde.	
Project 3: Ziggo – Netwerkbeheer & Storingsvisualisatie Voor het technisch team van Ziggo ontwikkelde ik ondersteunende tools waarmee netwerkstoringen sneller gelokaliseerd en opgelost konden worden. Het doel was om monteurs en beheerders real-time inzicht te geven in de status van het netwerk. Ik bouwde een WPF Windows-applicatie waarmee beheerders netwerkcomponenten zoals fibernodes, CMTS'en en taps konden beheren. Daarnaast realiseerde ik een ReactJS-webapplicatie met een interactieve kaartweergave, waarin storingen direct zichtbaar werden gemaakt. Foutieve nodes werden visueel gemarkeerd op de kaart, zodat gebruikers snel konden ingrijpen. Deze frontendcommuniceerde met een Angular 6-laag en werd aangestuurd via microservices in C#, waarbij data	

werd verwerkt via RESTful APIs en berichten werden afgehandeld via RabbitMQ en MassTransit. De combinatie van visuele feedback, messaging en integratie zorgde voor een robuust platform met snelle doorlooptijden.

De ontwikkelde tools verbeterden de efficiëntie van het storingsbeheer aanzienlijk. Monteurs konden sneller reageren op problemen, wat leidde tot kortere hersteltijden en hogere klanttevredenheid. De technische opzet was schaalbaar en goed onderhoudbaar dankzij de toepassing van een component-based architectuur en DevOps deployment via Docker.

Technieken & Tools	Werkwijze
ReactJS, Redux, Redux-Thunk, Angular 6, TypeScript, HTML5, CSS3, JavaScript (ES6+), RxJS, .NET Core, C#, Neo4J, SQL Server, RESTful Web API, RabbitMQ, MassTransit, Docker, Azure Cloud, Azure Cosmos DB, xUnit, StyleCop, SonarQube, Prettier, ESLint, GitHub, Git, Jira, Confluence, Bootstrap CSS	Scrum (Agile), Fullstack ontwikkeling met CI/CD, component-based architectuur en DevOps deployment pipelines.

Frontend Developer (ReactJS & WPF)	
December 2017 – Maart 2018	
Nipede	
Werklocatie: Hoorn	
Voertaal: Engels	
Project: Website en Desktopinterface voor 'De Persoonlijke Gezondheidscheck'	
<p>Voor Nipede ontwikkelde ik zowel een webapplicatie als een desktoptool ter ondersteuning van hun gezondheidsplatform, waarmee gebruikers inzicht kregen in vitaliteit en gezondheidsrisico's. De bestaande frontend vroeg om uitbreiding, optimalisatie en meer stabiliteit. Tegelijkertijd was er behoefte aan een beheerinterface voor interne contentredactie.</p> <p>Ik werkte aan twee sporen: het verbeteren van de bestaande ReactJS-applicatie en het ontwikkelen van een nieuwe WPF-desktopinterface in C#. In de React-applicatie breidde ik functionele componenten uit en verbeterde ik de performance en foutafhandeling. Ik implementeerde toegankelijkheidsverbeteringen (HTML5/CSS) en zorgde voor consistentie in styling, interactie en herbruikbaarheid op basis van ES6+ design patterns. De frontend-architectuur werd opgebouwd rond Redux en React Router.</p> <p>Parallel ontwierp en ontwikkelde ik een desktopinterface in WPF (MVVM-patroon), waarin medewerkers eenvoudig content zoals teksten en afbeeldingen konden beheren. Deze interface was gekoppeld aan RESTful API's en ondersteunde beeldredactie, planning en contentbeheer. Ik bewaakte de codekwaliteit met SonarQube en werkte nauw samen met backend developers. Binnen het Scrumteam droeg ik actief bij aan refinement, demo's en planningen.</p> <p>De combinatie van web- en desktopverbeteringen resulteerde in een snellere, stabielere en visueel consistentere gebruikerservaring. De nieuwe desktoptool maakte interne workflows efficiënter en verbeterde de kwaliteit en actualiteit van gepubliceerde content.</p>	
Technieken & Tools	Werkwijze
ReactJS, Redux, React Router, HTML5, CSS3, JavaScript (ES6+), RESTful WebAPI, WPF (C#), MVVM, SonarQube, ESLint, Prettier, Scrum, Jira, Bitbucket, SourceTree, Bootstrap CSS, Confluence, FHIR – Bekend met medische datamodellen (Nipede), snel inzetbaar in zorgcontext	Scrum (Agile), Hybride front- en desktopontwikkeling met focus op UX, herbruikbaarheid en technische kwaliteit.

Fullstack software developer / Migratie-specialist (VueJS & .NET)	
Aug 2017 – Dec 2017	
Conclusion	
Amsterdam	
Project: Modernisering van CLASS Leeromgeving	
<p>De bestaande CLASS-leeromgeving voldeed als ASP.NET Web Forms-applicatie niet langer aan moderne eisen voor toegankelijkheid, UX en onderhoudbaarheid. Mijn opdracht was tweeledig: het beheren van de legacy-applicatie en tegelijkertijd het moderniseren van de frontend met VueJS, gericht op performance, herbruikbaarheid en gebruikersvriendelijkheid.</p> <p>Ik ontwikkelde een modulaire frontend met VueJS, gebaseerd op HTML, CSS en JavaScript (ECMA 5/6). Componenten werden opgebouwd volgens component-based designprincipes, met focus op toegankelijkheid, semantiek en schaalbaarheid. Ik implementeerde herbruikbare UI-elementen en voerde optimalisaties uit voor foutafhandeling, validatie en performance. De frontend communiceerde via RESTful WebAPI's met een backend in ASP.NET Web Forms, en maakte gebruik van CQRS en NHibernate.</p>	

Daarnaast fungeerde ik als aanspreekpunt richting klant bij vragen over SLA's, performance en technische issues. Ik voerde root cause-analyses uit, documenteerde bevindingen en initieerde verbeteracties in samenwerking met het team. Testkwaliteit werd geborgd via xUnit-tests en SonarQube. In het Scrumteam leverde ik actief input tijdens planningen, demo's en sprintreviews.

De succesvolle migratie naar VueJS resulteerde in een snellere en gebruiksvriendelijkere leeromgeving. De stabiliteit en performance namen merkbaar toe, terwijl de onderhoudslast afnam. Tegelijk bleef de beschikbaarheid voor klanten gegarandeerd. Mijn werk leverde een hogere klanttevredenheid en betere UX op.

Technieken & Tools	Methodieken
VueJS, HTML5, CSS3, JavaScript (ECMA 5/6), ASP.NET Web Forms, C#, RESTful WebAPI, CQRS, NHibernate, SQL, Sitecore ASP.NET, xUnit, TFS, Jenkins, Octopus Deploy, TSUnit, Prettier, ESLint, SourceTree, Bitbucket, Jira, Confluence, UI-Kit CSS	Scrum (Agile) – Migratie- en transformatieproject binnen multidisciplinair team met focus op testbaarheid, performance en SLA's.

Fullstack software developer (Angular & .NET Core)	
Augustus 2016 – Augustus 2017	
TranseDEV	
Werklocatie: Hilversum	
Voertaal: Engels	
Project: Generieke API-laag en Reizigersinterface voor Mobiliteitssystemen	
Project 1: API Garden – Gestandaardiseerde API-laag voor legacy-integratie <p>Transdev beschikte over een verouderde Delphi-backend waarvan de functionaliteit toegankelijk moest worden gemaakt voor moderne web- en mobiele applicaties. Ik ontwierp en bouwde een generieke API-laag ('API Garden') met RESTful WebAPI's in .NET Core, die fungeerde als veilige en herbruikbare koppellaag tussen nieuwe frontends en legacy-logica. Deze API werd voorzien van moderne securityprincipes, authenticatie en autorisatie op applicatieniveau.</p> <p>Dankzij de nieuwe architectuur konden externe systemen veilig en uniform communiceren met het bestaande backendsysteem. De API Garden was een belangrijke stap richting een modulair en toekomstbestendig applicatielandschap.</p>	
Project 2: Telefonie-integratie – Wachtrijmodule met real-time statusinformatie <p>Reizigers die Transdev telefonisch benaderden kregen geen inzicht in wachttijden, wat tot onvrede en herhaalverkeer leidde. Ik ontwikkelde een wachtrij-integratiemodule waarmee realtime statusinformatie van bellers inzichtelijk werd via het ICE-lib platform. Wanneer een beller in de wachtrij werd geplaatst, kregen gebruikers direct een visuele weergave van wachttijd en positie.</p> <p>Deze uitbreiding verhoogde de transparantie richting klanten en verminderde de druk op de telefonische service. Het leverde aantoonbaar meer klanttevredenheid op en werd structureel opgenomen in de bredere serviceketen.</p>	
Project 3: Reizigersportaal – Angular-app voor real-time ritinformatie <p>Voor het reizigersportaal bouwde ik een Angular 2-applicatie waarmee reizigers real-time informatie over bus- en taxiritten konden opvragen. Ik ontwikkelde componenten met TypeScript en RxJS en bouwde een dynamische frontend waarmee gebruikersinformatie uit de API op een toegankelijke en gebruiksvriendelijke manier werd gepresenteerd.</p> <p>De UI was opgebouwd met HTML en CSS en geoptimaliseerd voor mobiel gebruik. De applicatie verbeterde de selfserviceopties voor reizigers en verminderde de druk op klantcontactkanalen.</p>	
Technieken & Tools	Methodieken
Angular 2, TypeScript, RxJS, HTML5, CSS3, RESTful WebAPI, .NET Core, ASP.NET MVC, C#, SQL Server, ICE-lib, xUnit, API Gateway, Jenkins, SonarQube, TSLint, Bitbucket, Jira, Confluence, SourceTree, OWASP, JWT, OATH 2	Scrum (Agile), Multidisciplinair team met focus op schaalbare integraties, realtime data en selfservice-oplossingen.