|  |
| --- |
| Provrex B.V. |
| Stageverslag |
| Portfolio programmeren met remix en prisma. |

|  |
| --- |
| Ambergen, Rick  12-2-2024 |

Contents

[Inleiding 2](#_Toc171460039)

[Beschrijving van het stagebedrijf. 3](#_Toc171460040)

[Producten en diensten van het stagebedrijf. 4](#_Toc171460041)

[Praktijkcase: De opdracht. 7](#_Toc171460042)

[Waarom Remix en Prisma? 7](#_Toc171460043)

[Interviews en Aanpassingen 8](#_Toc171460044)

[Indexpagina 8](#_Toc171460045)

[‘Saves’-pagina 9](#_Toc171460046)

[Overige Pagina's 10](#_Toc171460047)

[Responsive Design 11](#_Toc171460048)

[Praktijkcase: Het ontwikkelen. 11](#_Toc171460049)

[Beschrijving van je werkzaamheden en ervaringen op de werkplek. 14](#_Toc171460050)

[Slot 14](#_Toc171460051)

# Inleiding

Tijdens de eerste helft van mijn stage bij Provrex heb ik gewerkt aan twee belangrijke webapplicaties: Virtutec en Salcon.

Deze webapplicaties gebruiken frameworks zoals React, Next.js, Three.js, Drei en Tailwind.

De kennis die ik beheerde over deze frameworks en subframeworks was minimaal en er was nog veel kennis voor mij op te doen.

Toch na het besluit te nemen van tickets van Jira op te pakken en hiermee bezig te gaan.

Echter waren deze tickets te complex voor mij en heb ik meerdere realisaties gemaakt;

1. Het verder werken aan al bestaande webapplicaties is niet het werk wat ik later zou willen doen. Niet om het feit dat ik het niet interessant vindt maar meer om het feit dat het maken van nieuwe webapplicaties meer mijn passie is.
2. Dankzij mijn ADHD ben ik enorm creatief, echter beheers ik de skills niet om mijn creativiteit te uiten, op een skill na en dat is het maken van creatieve websites.
3. De beheersde kennis over de gebruikte frameworks en subframeworks gebruikt door Provrex bij mij erg minimaal is en dat er veel onbekende informatie is.

Na het maken van deze laatste realisatie heb ik het besluit genomen om documentatie, tutorials en collegas te benaderen om zoveel mogelijk kennis op te doen over de frameworks en subframeworks waar gebruik gemaakt van wordt zodat misschien niet hier maar in mijn toekomst dit me makkelijker af zal gaan.

Dit verslag zal beschrijven waar Provrex voor staat, wat Provrex hun producten zijn, wat ik tijdens mijn stage bij Provrex gedaan heb en de opdracht die ik gedaan heb tijdens mijn stage hier.

# Beschrijving van het stagebedrijf.

[Provrex](https://www.provrex.nl) is een innovatief bedrijf gevestigd in Emmen, Nederland, dat zich richt op het leveren van geavanceerde oplossingen met Virtual Reality (VR), Augmented Reality (AR) en Mixed Reality (MR).

Ze maken webapplicaties voor diverse sectoren zoals engineering, onderwijs en gezondheidszorg. Hun diensten omvatten onder andere virtuele rondleidingen, digitale tweelingen en voorspellend onderhoud.

Daarnaast biedt Provrex hardware en andere VR-gerelateerde services aan om bedrijven te ondersteunen bij het realiseren van de beste gebruikerservaring.

Ze werken samen met verschillende klanten om op maat gemaakte oplossingen te bieden die aansluiten bij de specifieke behoeften van elk project.

Provrex is trots op hun vermogen om technologieën te integreren die niet alleen efficiëntie en productiviteit verhogen, maar ook de manier veranderen waarop bedrijven en instellingen functioneren.

Door middel van hun expertise in VR, AR en MR, helpen ze klanten om innovatieve toepassingen te implementeren die leerprocessen verbeteren, complexe systemen visualiseren en onderhoudsprocedures optimaliseren.

Het bedrijf beschikt ook over een scala aan VR-apparatuur en biedt consultancy services om ervoor te zorgen dat klanten de beste en meest kosteneffectieve oplossingen krijgen. Met een team van experts staan ze klaar om klanten te begeleiden en te ondersteunen bij het realiseren van hun technologische doelen.

# Producten en diensten van het stagebedrijf.

Om er achter te komen welke producten en diensten Provrex aanbied is er een interview gehouden. Hierbij zijn de volgende vragen gesteld om in beeld te krijgen welke producten en diensten Provrex als bedrijf aanbied aan hun klanten;

* Welke (web)applicatie(s) ga je beschrijven?
* Virtutec en Salcon aangezien dit de 2 applicaties zijn waar ik zelf ook tijdelijk aangewerkt heb.

Vanaf hier begint het interview;

* Wat is de doelgroep/gebruikersgroep van de (web)applicatie?
* De doelgroep is de industrie en dan voornamelijk de maak industrie, dus denk aan grote machines, van verpakking machines tot centrifuges van bloedplasma.
  + Salcon is dan ook echt een verkoop tool, sales config en ook die is geend op industrie als dan op verkoop kanalen van onze klanten.
* Wat is de functionaliteit van de (web)applicatie?
* Virtutec is heel breed want dit is het boven liggende platform van Salcon, Procon en Sercon. Dus Virtutec zorgt ervoor dat klanten bij deze producten kunnen komen en dient gelijk als platform waar de virtuele modelen van de machines worden geupload.
  + Salcon is sales config dus hiermee kunnen de bedrijven hun eigen producten configgen en verkopen aan hun klanten dus dit dient als een verkoop tool voor onze klanten.
* Wat doet het, wat biedt het de gebruiker aan functies en dergelijke
* Een platform wat hun producten in 3D visualiseren. En daarmee configureerbaar zijn voor hun klanten als voor hun verkoop kanalen.
  + Dit zorgt ervoor dat de klant gevisualiseerde ofertes kan uitbrengen naar hun klanten.
* In welke taal/talen is de (web)applicatie ontwikkelt en waarom is hiervoor gekozen?
* De backend van de platformen is gecshreven in laravel, hiervoor is gekozen omdat intern hier van de kennis en omdat het een populair en stabiel php framework is.
  + De frontend is geschreven in react, ook hierbij geld dat de interne kennis hiervoor aanwezig is en daarom de keuze bij dit framework is gemaakt.
  + Ook is er nog gebruik gemaakt van Three.js en Drei, Three.js is gebruikt voor het visualiseren van de 3d modellen en Drei is gebruikt voor de controlle van de camera.
* Hoeveel mensen hebben aan het ontwikkelen van deze (web)applicatie gewerkt en welke afspraken zijn er gemaakt m.b.t. bijvoorbeeld het coderen?
* Goeie vraag, Er hebben zo’n zeven man aan gewerkt waarbij er aan de standaard conderings regels van laravel en react zijn gehouden.
  + Voor laravel is dit obeject gerelateerd php en voor react was het allemaal heel basic coderings regeling. In beide beide wordt gebruik gemaakt van de [MVC](https://en.wikipedia.org/wiki/Model–view–controller) structuur.
* Hoe zag de ontwikkelomgeving er uit? Denk aan IDE’s, servers, tools en meer.
* Lokaal moest het in docker werkend zien te krijgen dus ook gewoon met MPN
* Hoe en wanneer is er getest?
* Eerst wordt het lokaal getest, hierna volgen testen op de acceptatie omgeving en daarna wordt naar de productie gepushed.
* Ging het hier om TDD (Test Driven Development)? (Denk ook aan Unittesting, debugging, ..... )
* Nee niet altijd omdat we met 3d bestanden werken is dat heel lastig.
* Hoe ziet de structuur van de (web)applicatie er uit?
* De webapplicatie bestaat uit een back-end, een front-end, een database en S3 file storage (strato S3 storage).
* Wat is de structuur van de database?
* Zeer complex. Dit is de beste manier hoe het beschreven kan worden aangezien we er 4 volle dagen mee bezig zijn geweest om de structuur van de database uit te zetten en uit te ontwikkelen.
* Is er nagedacht over de UI? Zo ja, hoe? En hoe is dit dan geïmplementeerd in de (web)applicatie?
* Daar hebben we een UI/UX designer voor ingehuurd, echter is deze niet meer bij ons in diesnt inmiddels.
* Hoe wordt er onderhoud gepleegd op de (web)applicatie? Is dit op basis van ontwikkelingen van de software of vragen van de klanten?
* Vragen van de klanten, vanaf daar worden er nieuwe functies of updates voor gevraagd, die zetten wij dan weer eerst via lokaal naar acceptatie naar productie.
* Als er gesproken wordt over een deployment (nieuwe versie), hoe wordt dit dan aangepakt? Wordt de nieuwe versie intern of extern ontwikkeld? En welke partijen zijn er betrokken?
* Altijd intern, we hebben ons eigen ontwikkel team dus dit pakken wij intern op en de opdracht/vraag wordt dan ontwikkeld terwijl er contact wordt gehouden met de klant en dan uiteindelijk na de test periode wordt deze op de acceptatie branch gezet.

# Praktijkcase: De opdracht.

De eerste opdrachten waren in de form van tickets op Jira, dit zijn de opdracthen waar ik mee bezig ging;

* Het dashboard van Virtutec maken/uitbreiden.
* Het ‘rekbaar’ maken van een rollerbaan 3d model.

Na een goeie poos alleen al het proberen op te zetten van de development envoirement was ik begonnen met de ticket van het rekbaar maken van het 3d model van de rollerbaan.

De eerste stappen die ik hierbij heb ondernomen was het proberen met een slider het model langer te maken. Dit ging in het begin al vrij moeizaam aangezien er een overduidelijk gebrek aan kennis van mijn kant was over de codetaal die gebruikt werdt.

Op advies ben ik onderzoek gaan doen naar meshes en hoe ik deze met code zou kunnen manipuleren. Maar helaas hier kwam ik ook niet veel verder mee.

Na overleg was er het besluit genomen dat ik een poging ging doen om het dashboard uittebreiden met nieuwe functies.

Maar helaas wederom kwam hier hetzelfde resultaat uit.

Na wederom in overleg te gaan is er voor mij een nieuwe opdracht bedacht.

Ik heb de opdracht gekregen om [mijn bestaande portfolio](https://rickmuda.nl), dat momenteel is opgebouwd met HTML, CSS en JS, te herschrijven met behulp van nieuw geleerde code frameworks.

De frameworks die ik ga gebruiken zijn Remix en Prisma.

## Waarom Remix en Prisma?

[Remix](https://remix.run) is een full-stack webframework dat de focus legt op de gebruikersinterface (UI) en gebruik maakt van webstandaarden om een snelle, soepele en robuuste gebruikerservaring te leveren. Hierdoor wordt het gebruik van de website veel plezieriger.

[Prisma](https://www.prisma.io) is een moderne ORM (Object-Relational Mapping) die het werken met databases eenvoudig en efficiënt maakt. Het biedt een type-safe API, waardoor gegevens snel en betrouwbaar kunnen worden opgevraagd en gemanipuleerd. Met Prisma kan ik mij volledig richten op de ontwikkeling van mijn applicatie, terwijl de complexe database-interacties voor mij worden afgehandeld.

De reden dat ik voor een combinatie van Remix en Prisma heb gekozen, is omdat beide frameworks naadloos op elkaar aansluiten. Remix stelt mij in staat een gebruiksvriendelijke en aantrekkelijke UI te creëren, terwijl Prisma zorgt voor veilige en efficiënte datamanipulatie.

## Interviews en Aanpassingen

De interviews waren vrij kort omdat het hoofdontwerp en idee al aanwezig waren in mijn bestaande portfolio. Desalniettemin wil ik enkele aanpassingen doorvoeren om de UI en UX te verbeteren.

## Indexpagina

A screenshot of a video game

Description automatically generatedVoor de indexpagina komt er een laadschema dat doet denken aan een oude computer die opstart, om het thema van mijn portfolio te versterken.

Hoewel de basis van de indexpagina behouden blijft, wordt deze vertaald van HTML, CSS en JS naar TypeScript met Remix.

## ‘Saves’-pagina

Als inspiratie voor de ‘saves’-pagina ga ik het menu scherm van Crash Bandicoot 4: It’s About Time gebruiken.

Ik zal even een korte uitleg geven wat ik ga vervangen of weghalen en wat ik ga gebruiken als inspiratie

* Tekst: De tekst "select a save slot" bovenaan het scherm wordt verwijderd om een minimalistisch gevoel te behouden en omdat deze tekst niet perse nodig is om te laten zien welke pagina het is.
* Illustratie: Het masker wordt vervangen door een tekening van mezelf, met subtiele animaties gemaakt in Adobe After Effects voor een levendigere uitstraling van de webpagina.
* Beschrijving: De tekst onder de illustratie geeft een korte beschrijving van de pagina achter de knop waarover de muis hoverd. Als er nergens overheen wordt gehooverd, blijft de tekst van de laatst gehoverde knop staan.
* Knoppen: De twee knoppen die zich links onderin het scherm bevinden zullen vervangen worden door een kleinere klikbaar stukje tekst met een login en logout knop (op basis van ofde gebruiker ingelogged is of uitgelogged is).

De vakken met ‘new game’ worden klikbare pagina's voor:

* Over mij
* Projecten
* W.I.P.
* Contact

## Overige Pagina's

Over mij: Tekst wat voorgeschreven is komt met een typing effect op de pagina, terwijl de tekst uitgetypt wordt speelt er een .wav bestand af wat klinkt als of een character uit een game aan het praten is zoals bijvoorbeeld de characters in de game [animal crossing](https://www.youtube.com/watch?v=2VOalnTTYjM).

Ook komt hier een foto van mezelf bij te staan waarvan het zal lijken als [een canadees persoon uit southpark](https://www.youtube.com/watch?v=zIUtG5BdFqQ) aan het praten is terwijl de tekst uitgetypt wordt.

Projecten: Deze pagina krijgt aanzienlijke aanpassingen. Met een dankzij prisma gekoppelde database kunnen de projecten door een admin gebruiker (ik) toegevoegd worden. Elke projectpagina krijgt dezelfde lay-out door gebruik te maken van een uniek ID die dan gepakt wordt zodra er op de cover image gedrukt wordt van een project.

In de project detail pagina komt dan ook een knop waarmee het debetrevende project bezocht, bekeken en gebruikt kan worden.

Ook komt er op deze pagina een kleine image gallery met fotos en een gif van het project waar door heen gekeken kan worden.

W.I.P. (Work in Progress): De WIP pagina zal projecten bevatten die nog niet afzijn of waar nog aan begonnen moet worden. De rest van deze paginas zullen dezelfde layout en functies hebben van de projecten paginas.

Contact: De contact pagina gaat opgedeeld worden in 2 delen (links en rechts) met een email form waarmee makkelijk emails naar mij verstuurd zouden kunnen worden en een heleboel icons van sociale media platformen waar ik gebruik van maak.

Wanneer er over 1 van de 2 delen heen gehoverd wordt onthoud de pagina dit en vervaagd de andere kant zodat de aandacht duidelijk op de op dat moment gebruikte kant kan blijven.

## Responsive Design

Tot slot zal er een responsive design gemaakt worden voor alle paginas waaronder de paginas die alleen door admins gebruikt kunnen worden, zodat de hele website op ieder platform dezelfde experience kan brengen.

# Praktijkcase: Het ontwikkelen.

Voor het ontwikkelen van het project is er gekozen voor een stapgeweisde methode. Deze keuze is gemaakt omdat dit voor mij het beste zal werken en er geen overlaps kwamen en dat er verschillende paginas half afwaren voordat er aan een andere begonnen was.

Voordat we bij de eerste pagina uitleg uitkomen was er een klein probleempje met een functie voor de website. Een back knop op paginas waar dit nodig was werkte niet op de 4 paginas maar dit was een probleem met het koppelen van de css stylesheets en was makkelijk opgelost na een korte debugging fase.

De eerste pagina van het project is een boot-up met sterke inspiratie van oude computers, de inspiratie hiervan komt van een [codepen](https://codepen.io/FailedSleep/pen/xdWoOe) gemaakt door [Failed Sleep](https://codepen.io/FailedSleep).

Met zijn code als referentie is hetgelukt om een bootscherm te maken met custom tekst (zie foto).

Deze ‘animatie’ heb ik vrij kort gemaakt zodat het niet ‘te lang’ duurt voordat je bij de eerste echte pagina komt.

Hierna was de index pagina aan de beurt, deze was vrij simpel aangezien het orginele design mij al goed beviel.

A screenshot of a video game

Description automatically generatedDeze was ook vrij makkelijk te herschrijven aangezien het veel css was waar geen drastische aanpassingen aangemaakt hoefde te worden.

Een kleine bug waar ik tegen aanliep was dat na de boot animatie de css werdt niet goed geladen waardoor de tekst op de mainscreen pagina niet de corecte font kreeg.  
Dit was vrij simpel opgelost door een aparte css class aantemaken die specifiek doorgeeft dat ze de goeie font moeten hebben.

De volgende pagina was de saves pagina, zoals [eerder beschreven](#_‘Saves’-pagina) heb ik hier inspiratie voor gepakt van een spel.

Voor de foto heb ik mijn partner gevraagd om een tekening met de color pallete van het project, deze heb ik toen in met After Effect subtiele animatie gegeven zodat de pagina iets levendiger lijkt.

A screenshot of a video game

Description automatically generated

Na de saves pagina kwam de about me pagina.

Deze pagina bracht de meeste problemen maar er waren 2 grote problemen die hier ook in meer detail behandeld zullen worden.

Het eerste probleem was dat de orginele type writer effect waar gebruik van was gemaakt in mijn momentele portfolio niet bruikbaar was in TypeScript.

Nou in eerste instantie was er een package die ik gevonden had die helaas niet de modules had waar ik gebruik van wou maken, maar na toch nog verder gezocht te hebben kwam ik bij [typewriter-effect](https://www.npmjs.com/package/typewriter-effect) uit.

Niet alleen was de documentatie hiervan duidelijker in mijn mening maar was het ook gelijk duidelijk hoe ik hiermee aan het werk kon.

Het tweede probleem was bij het feit dat er een foto vervangen moest worden terwijl de tekst aan het typen was.

Eerst was het dat de fotos bleven veranderen, toen dat de images veranderde wanneer de tekst aan het ‘verwijderen’ was.

Gelukkig was er nog 1 functie die toegevoegd moest worden en dat zijn de soundeffects voor wanneer de tekst uitgetypt wordt, dezelfde problemen als die aanwezig waren bij de fotos waren hier wederom aanwezig maar dezelfde oplossing werkte hier ook.

Na de functionaliteit was de styling aan de beurt.

Na het maken van de about me pagina was de projecten paginas aan de beurt, het gaat hier dan om 3 verschillende pagina’s;

* De projecten pagina zelf waar alle toegevoegde projecten komen te staan.
* De pagina voor het toevoegen van projecten.
* De pagina met de details van het geselecteerde project.

De pagina waar de projecten zelf op komen te staan blijft simpel, rijen van 3 projecten en zodra er 3 op 1 rij staan zal het volgende project op de rij hieronder komen te staan. Ook bevat deze pagina een koptekst met ‘Projects’ omdat er later nog een pagina bij komt die exact hetzelfde is maar met een ander doeleinde.

De pagina voor het toevoegen van projecten is vrij simpel aangezien dit alleen maar gebruikt zal worden door admin gebruikers, deze pagina bevat een form waar er een paar dingen aan een project toegevoegd moet worden;

* De naam van het project.
* De categorie waar het project binnen valt.
* De beschrijving van het project.
* Een cover image die op de projecten pagina zelf zou komen te staan.
* Een gif die als eerste getoont zal worden in de image gallery op de project detail pagina.
* 3 andere fotos die ook in de image gallery terecht zullen komen.
* De path naar het project zodat het project ook bezocht en gebruikt kan worden.

Dit wordt allemaal opgeslagen in de database zodat de code zelf niet een grote rommel wordt met enorm veel paths voor de fotos etc.

Als laatste de project details pagina.

Deze wordt in 2 delen opgedeeld (links en rechts) met aan de linker kant zit de image gallery zodat de bezoeker voor het bezoeken van het project kan zien hoe het project eruit ziet. En hieronder bevindt zich de knop om het project te bezoeken en indien een admin ingelogd is een knop om het project te verwijderen (inclusief vanuit de database)

De rechter kant van zal de titel en beschrijvng bevatten met hieronder een kleine bar die aan de onderkant komt waar makkelijk te zien is welke code talen er gebruikt worden. Deze informatie zal via de Github API komen zodat dit ook niet in de code zelf voor ieder project zelf geschreven moet worden.

Tijdens het ontwikkelen van deze specefieke pagina was er een probleem aanwezig dat wanneer er een project verwijderd werdt dat de pagina naar een 404 werdt gestuurd en dit kwam omdat de pagina verversed werdt maar de pagina niet meer bestond.

Na het vragen van wat collegas op stage is het gelukt om dit probleem op te lossen door een extra redirect toe tevoegen naar de projects pagina wanneer er een 404 is.

Ook was het probleem aanwezig dat de delete knop altijd aanwezig was terwijl alleen admin gebruikers deze hoorde te kunnen zien, dit was echter makkelijk op te lossen door dezelfde code te gebruiken die ook voor het toevoegen van een project knop gebruikt was.

# Beschrijving van je werkzaamheden en ervaringen op de werkplek.

# Slot