|  |
| --- |
| Provrex B.V. |
| Stageverslag |
| Programmeren met remix en prisma voor portfolio. |

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

Contents

[Inleiding 5](#_Toc170829602)

[Beschrijving van het stagebedrijf 6](#_Toc170829603)

[Product(en) / dienst(en) van het stagebedrijf 7](#_Toc170829604)

[Praktijkcase: de opdracht 8](#_Toc170829605)

[Praktijkcase: Ontwikkelen 9](#_Toc170829606)

[Beschrijving van je werkzaamheden en ervaringen op de werkplek 10](#_Toc170829607)

[Slot 11](#_Toc170829608)

# Inleiding

HET STAGEVERSLAG

Tijdens je 2e-jaars stage moet je een stageverslag schrijven.

Dit is best veel werk, dus begin er op tijd mee.

De eisen die aan het verslag gesteld worden zijn:

 Het stageverslag moet ongeveer 20 pagina’s tekst bevatten. Illustraties tellen niet mee.

 Hoofdstuk 2 en 3 lever je halverwege je stage bij je stagebegeleider in.

 Het gehele verslag lever je uiterlijk 2 weken voor het einde van de stage in.

 Je stagebegeleider geeft aan of hij/zij de tussenproducten en eindproducten op papier of digitaal wil ontvangen.

 Voor de vormgeving en indeling volg je de aanwijzingen hieronder nauwkeurig op. Kun je bepaalde hoofdstukken of onderdelen niet maken? Overleg dan met je stagebegeleider.

VERPLICHTE HOOFDSTUKKEN

Het stageverslag van de opleiding Software Development bouw je op de volgende manier op:

** Inleiding**

** Beschrijving van het stagebedrijf**

** Product(en) / dienst(en) van het stagebedrijf**

** Praktijkcase: de opdracht**

** Praktijkcase: Ontwikkelen**

** Beschrijving van je werkzaamheden en ervaringen op de werkplek**

** Slot.**

De hoofdstukken 3, 4 en 5 lichten we hieronder in het kort toe.

# Beschrijving van het stagebedrijf

# Product(en) / dienst(en) van het stagebedrijf

# Praktijkcase: de opdracht

HOOFDSTUK 3 – DE PRODUCT(EN)/DIENST(EN) VAN HET STAGE BEDRIJF

Je stagebedrijf zal met een grote waarschijnlijkheid zelf (web)applicaties hebben gemaakt. Voor deze opdracht ga je minstens één van deze (web)applicaties een gedetailleerd verslag schrijven, waarbij je ten minste de onderstaande vragen in dit hoofdstuk beantwoord:

 Welke (web)applicatie(s) ga je beschrijven?

 Wat is de doelgroep/gebruikersgroep van de (web)applicatie?

 Wat is de functionaliteit van de (web)applicatie?

Wat doet het, wat biedt het de gebruiker aan functies en dergelijke

 In welke taal/talen is de (web)applicatie ontwikkelt en waarom is hiervoor gekozen?

 Hoeveel mensen hebben aan het ontwikkelen van deze (web)applicatie gewerkt en welke afspraken zijn er gemaakt m.b.t. bijvoorbeeld het coderen?

Denk b.v. aan naamgevingsconventies in code  Hoe zag de ontwikkelomgeving er uit? Denk aan IDE’s, servers, tools en meer  Hoe en wanneer is er getest?

Ging het hier om TDD (Test Driven Development)? (Denk ook aan Unittesting, debugging, ..... )  Hoe ziet de structuur van de (web)applicatie er uit?

 Wat is de structuur van de database?

 Is er nagedacht over de UI? Zo ja, hoe? En hoe is dit dan geïmplementeerd in de (web)applicatie?

 Hoe wordt er onderhoud gepleegd op de (web)applicatie? Is dit op basis van ontwikkelingen van de software of vragen van de klanten?

 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?

# Praktijkcase: Ontwikkelen

HOOFDSTUK 4 – PRAKTIJKCASE: DE OPDRACHT

Voor je praktijkcase ga je in overleg met je praktijkbegeleider. Het gaat erom dat je een opdracht krijgt om:

 een (deel van een) (web)applicatie zelf te ontwikkelen of aan een bestaande (web)applicatie een nieuwe functionaliteit te ontwikkelen

In beide gevallen mag het betrekking hebben op een interne (web)applicatie (voor gebruik binnen de eigen organisatie) of om een externe (web)applicatie (gebruik door andere organisaties, oftewel klanten). Dit hoofdstuk deel je ten minste in de onderstaande paragrafen in:

1. De opdracht

Hierin beschrijf je gedetailleerd in eigen woorden wat de opdracht is.

2. Interview(s)

Interview de opdrachtgever(s) om voldoende informatie te verzamelen zodat je in staat bent je ontwerp te maken. In deze paragraaf doe je verslag van de interviews.

De opdracht die ik gekregen heb is om met de nieuw geleerde code frameworks mijn (HTML, CSS en JS) portfolio opnieuw te schrijven.

De 2 frameworks waar ik gebruik van ga maken zijn remix en prisma.

Remix is een full-stack webframework dat je laat focussen op de U.I (User Interface) en terugwerkt via web standaarden om een snelle, soepele en veerkrachtige gebruikerservaring te leveren. Hierdoor wordt het gebruiken van jouw website veel plezieriger.

Prisma is een moderne ORM (Object-Relational Mapping) die je helpt om een makkelijk en efficiënt met je database te werken.

Het biedt een type-safe API waardoor je snel en vertrouwelijk gegevens kan opvragen en manipuleren. Met Prisma kan je je volledig richten op het ontwikkelen van je applicatie, terwijl het moeilijke werk van database interactie voor je gedaan wordt.

De reden dat ik voor een combinatie van Remix en Prisma heb gekozen is omdat beide erg mooi op elkaar inspelen, Remix zorgt ervoor dat ik voor een makkelijke en mooie U.X. kan maken terwijl Prisma mij een veilige manier aanbiedt om mijn data in de database te bewaren en aan te passen als dit nodig is.

De interviews waren vrij kort aangezien het hoofd ontwerp en idee er al waren, mijn al bestaande portfolio, echter wil ik er wel dingen aan aanpassen zodat de U.I. en U.X. beter en makkelijker zijn.

Voor de index pagina aan komt een soort laad scherm wat lijkt als een oude computer die opstart zodat het thema van m’n portfolio nog duidelijker wordt.

De index pagina krijgt niet heel veel aanpassingen aangezien ik zelf enorm tevreden ben met hoe deze origineel is maar deze gaat ook ‘vertaald’ worden van HTML, CSS en JS naar typescript met Remix.

Als inspiratie voor de ‘inhoudsopgave’ ( wat bij mij dan de saves.tsx is ) van het menu scherm van Crash Bandicoot 4: It’s About Time genomen ( zie foto hieronder voor het menu scherm )

De ‘select a save slot’ boven in het scherm heb ik besloten weg te halen zodat ik alsnog een vrij minimalistisch gevoel kan geven aan deze pagina en al helemaal omdat het de 2e pagina is die iemand zou bezoeken op mijn portfolio.

Aku Aku ( het masker ) ga ik vervangen met een tekening van mezelf die ik met Adobe After Effect kleine subtiele animaties ga geven zodat het levendiger lijkt.

De tekst onder Aku Aku wordt een kleine beschrijving van de pagina waar op dat moment overheen gehoverd wordt, als de muis nergens overheen hoverd dan onthoudt tie de meest recentelijke knop waarover heen gehoverd wordt en blijft die tekst er staan.

De knop met de tekst new game ga ik vervangen met een kleine subtiele login en logout knop voor het toevoegen en bewerken van projecten als de gebruiker

( wat waarschijnlijk alleen ik ben maar voor gemakzucht heb ik dit gekozen omdat het makkelijker is dan iedere keer de code te veranderen om een nieuw project toe te voegen of aan te passen )

De vakken met ‘new game’ worden de bezoekbare pagina’s waar dus de volgende onder zullen vallen;

About me

Projects

W.I.P.

Contact

About me spreekt voor zich dit wordt bijna een kopie van hoe dit nu is ( met in de toekomst mogelijk een 3d model van mezelf in een ‘low poly’ stijl (denk ps1/2 en n64) die ik er dan met three.js erin kan zetten ).

Projects spreek ook wel voor zich echter deze zal best wel wat aanpassingen hebben in vergelijking met de huidige ‘projects’ pagina. Met behulp van Prisma ga ik een database toevoegen en als de gebruiker dan ingelogd is ( ik ) kan de gebruiker projecten toevoegen en aanpassen.

Hoe ik op het huidige moment voor ieder project een aparte pagina heb aan moeten maken voor wat extra uitleg en een manier om dit project te bezoeken wil ik ervoor zorgen dat dit niet meer nodig is.

Ik wil voor alle projecten uitleg pagina’s dezelfde lay-out hebben door middel van het meegeven van een ID aan ieder project bij het aanmaken zodat deze weer opgeroepen kan worden en ik dit allemaal binnen 1 bestand zou kunnen houden.

Deze pagina zal dan een kleine galerij bevatten met een paar foto’s van het project ( mogelijk met of zonder code ), een beschrijving van het project met aan de zijkant een ‘bar’ met een op percentage gebaseerd systeem laten zien welke code hiervoor gebruikt is ( zie afbeelding voor voorbeeld ).

W.I.P. wordt een pagina voor eventuele projecten en/of evenementen waar ik mee bezig ben op dat moment.

Deze gaat qua lay-out veel lijken op de projecten pagina om de illusie te creëren dat de 2 pagina’s met elkaar verbonden zijn.

Dit zal daarom waarschijnlijk ook hetzelfde idee bevatten met hoe ik dit doe met de projecten pagina ( mogelijk met een knop dat wanneer die ‘aan’ gezet wordt dat deze van W.I.P. naar de projects pagina gaat )

Niet alleen is dit voor een bezoeker om te zien waar ik mee bezig ben, ook is dit handig voor mezelf om mijn prioriteiten te kunnen stellen in de projecten die daar al een poos staan zodat ik deze ook ga afronden.

De contact pagina, deze spreekt wederom ook weer voor zichzelf echter i.p.v. hoe die er nu uitziet met allemaal tekst en de 3 ‘categorieën’ wil ik dit een mooiere U.I. maken zodat het overzichtelijker is en makkelijker wordt om contact met me op te nemen.

En dan als laatste een responsive design ontwerpen en maken, een goed portfolio moet natuurlijk op elk mogelijk apparaat er goed uitzien, hiervoor moet ik dus nog een ontwerp maken en deze dan ook toepassen op het project.

# Beschrijving van je werkzaamheden en ervaringen op de werkplek

HOOFDSTUK 5 – PRAKTIJKCASE: ONTWIKKELEN

Dit hoofdstuk deel je in de onderstaande paragrafen in:

1. Analyse van de interviews

Je analyseert de interviews en filtert hier de onderstaande zaken uit en beschrijft deze in eigen woorden:

a. De processen die betrekking hebben op de (web)applicatie

Wanneer en in welke volgorde vinden bepaalde werkzaamheden plaats?

b. De gegevensstroom

De (web)applicatie is op dit moment nog je zogenaamde black box. Welke gegevens komen van buiten de black box je (web)applicatie in en welke informatie moet vanuit de black box er weer uit komen. c. Eisen en wensen

Beschrijf de wensen en de eisen die de opdrachtgever(s) heeft qua functionaliteit, gegevens- en informatiestroom, gebruik van de (web)applicatie etc...

2. Beschrijving van de ontwikkelomgeving

3. Database ontwerp: Beschrijf in detail de tabellen, kolommen en de relaties tussen tabellen. Maak daarbij ook gebruik van schematechnieken.

4. Ontwerp(en) van de UI: Denk van te voren na over hoe een gebruiker met je (web)applicatie gaat werken. Maak prototypes van je schermen, beschrijf van de verschillende componenten op ieder scherm de eigenschappen.

5. Structuur van de (web)applicatie. Een aantal hulpvragen als voorbeeld: Hoe kom je van het ene deel van de (web)applicatie bij een ander deel van de (web)applicatie? En wanneer?

6. Conventies: naamgeving van methods, properties, classes, mappen, bestanden en meer. Beschrijf duidelijk de afspraken die hierover gemaakt worden.

7. Testen: Hoe ga je testen tijdens de ontwikkeling, denk aan TDD, Unittests, Debugging e.d.? Maak een plan voor een acceptatietest voor a) de opdrachtgever en b) één of meer gebruikers.

8. Version control: Hoe ga je version control regelen?

Toevoegen van andere paragrafen is uiteraard toegestaan. Des te beter je een dergelijk project beschrijft des te beter de overdracht is.

# Slot