

Kennick Games portfolio

Door Kenrick Halff

Doel

Dit document is bedoeld om een demonstratie te geven van verschillende frontend technieken en technologieën. Kennick Games is een zelf gebouwde web applicatie zonder gebruikt van een CMS of ander design/content management software. De applicatie staat op het moment offline, en is nog in ontwikkeling.

Kennick Games is een spelletje website die ik aan het ontwikkelen ben. Het doel van Kennick Games een platform bieden voor mensen die zelf een spel hebben gemaakt maar niet aan licentie kosten vast wilt zitten van programmas zoals Steam. Door reclames aan te bieden en een betaal model voor gebruikers kunnen de developers zelf ook wat bijverdienen.

Technologiën

Frontend

De frontend gebruikt als javascript framework React (ontwikkeld door Facebook).

SASS word gebruikt voor de CSS.

De design richtlijnen zijn Material Design door Google.

Webpack word gebruikt om alle javascript en css te compilen naar 1 bestand dat word geminimaliseerd en geoptimaliseerd zodat de gebruiker het snelst mogelijke performance de applicatie zelf zo min mogelijk data verbruikt voor mobiel gebruik

Backend

De backend gebruikt een NodeJs server.

De server is dus geschreven in Javascript, en maakt gebruik van het Express framework.

Hiervoor is gekozen om “universal rendering” mogelijk te maken. Dit betekend dat als Javascript in de browser niet beschikbaar is, dat de server het op zich neemt om het javascript uit te voeren op de server en het resulterende HTML te sturen naar de browser.

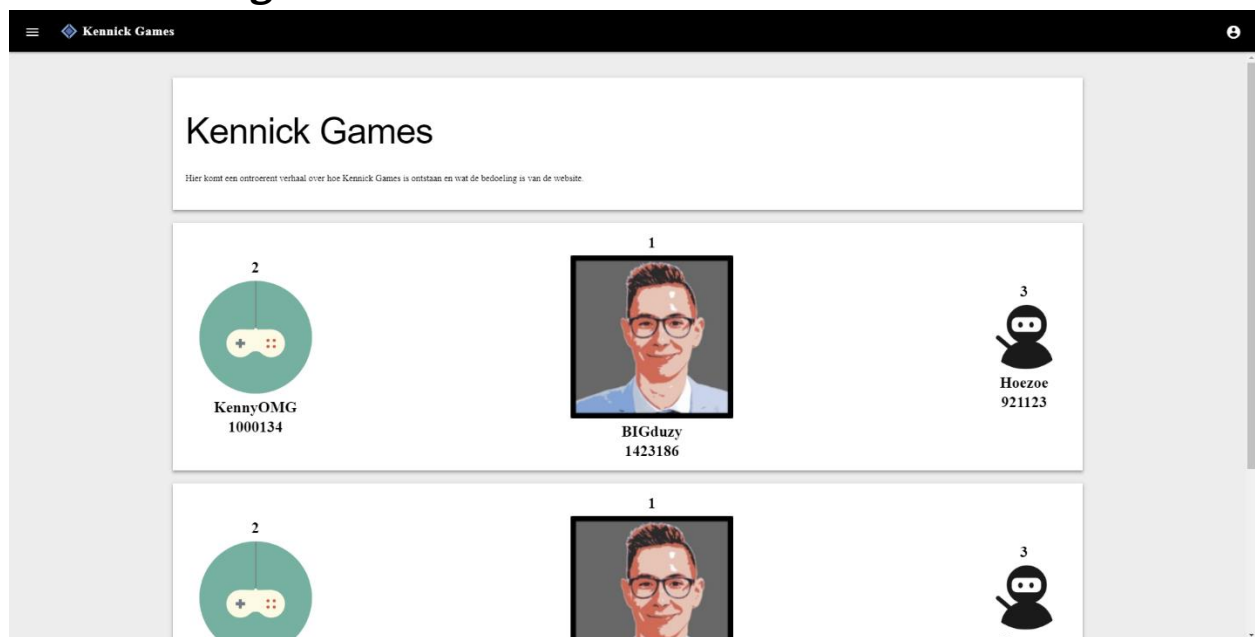
Voor de database word Sequelize gebruikt, dit is een ORM zodat de database kan worden gebouwd doormiddel van de geprogrammeerde modellen.

De server functioneert als een RESTapi, door het gebruik van een javascript framework in de frontend heeft de server niet de verantwoordelijkheid van het renderen van de html (behalve als javascript niet beschikbaar is). Dit zorgt dat de server minder lasten heeft dan bij een traditionele stack en valt dus te besparen op hosting kosten.

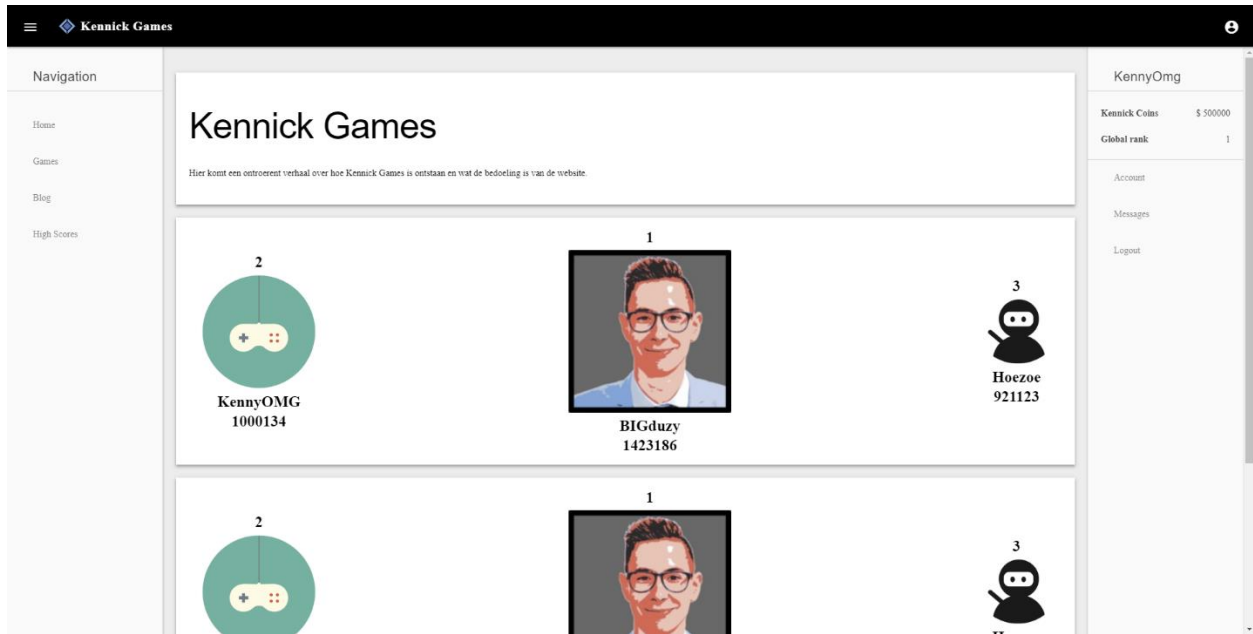
Ik heb gebruik gemaakt van een boilerplate die ik zelf ontwikkeld heb voor het opzetten van de API: <https://github.com/Kennyomg/restapi-boilerplate>

Design

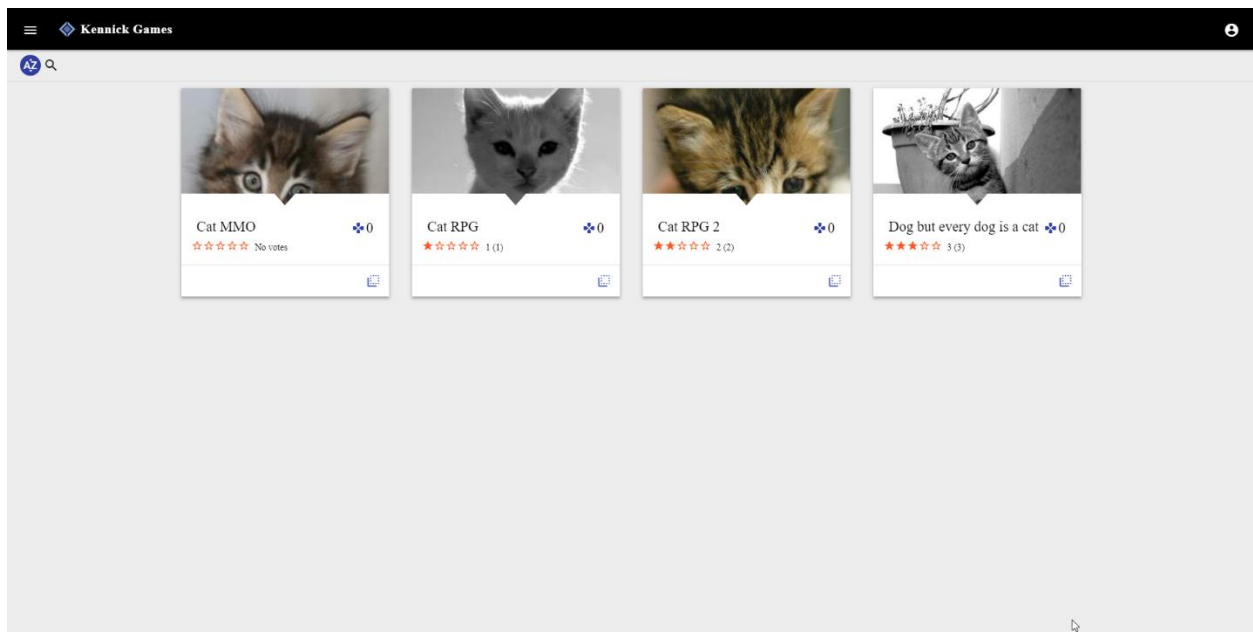
Material Design



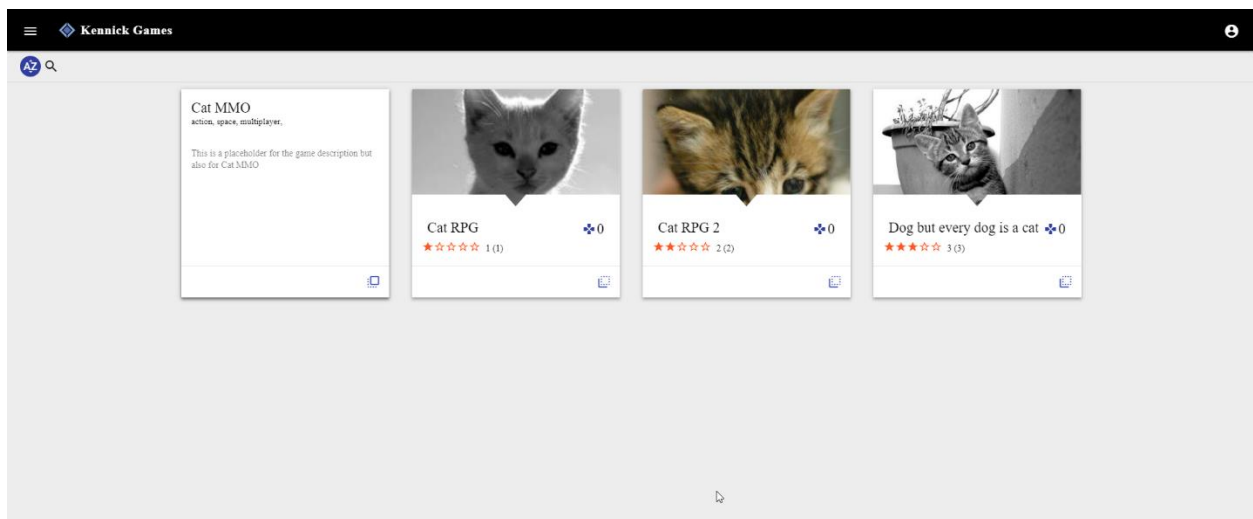
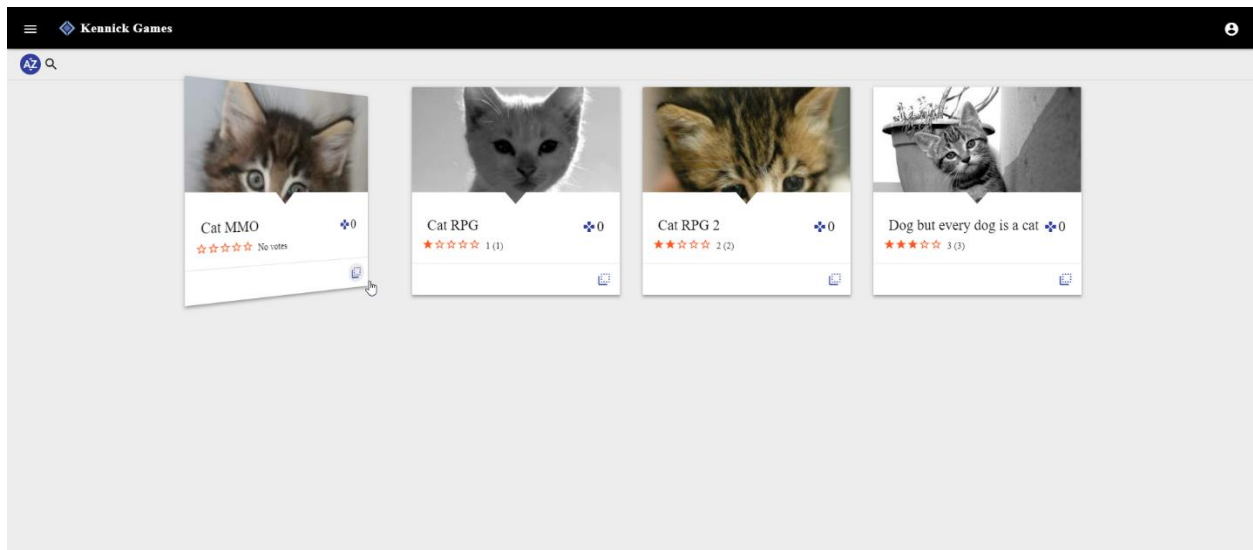
Dit is de home pagina en weergeeft de gebruikers met de hoogste score. Dit is een totale score wat je hebt verdient verspreid over elk spel. Hier kan je zien dat er gebruik word gemaakt van Material cards, dit zorgt voor een rustig ontwerp waarbij de belangerijke elementen toch worden benadrukt



Er is een uitklapbaar menu links en recht. Links is voor navigatie door de website en rechts is voor het beheer van je account.



Hier is een demonstratie als je navigeert naar “Games”. Dit is een lijst van de beschikbare spellen in vorm van “tegels”. Rechts onderin de tegel is een icoon aangeduid wat interactief is. Hiermee kan je de tegel omdraaien, dit is dan ook geanimeerd in pure CSS.



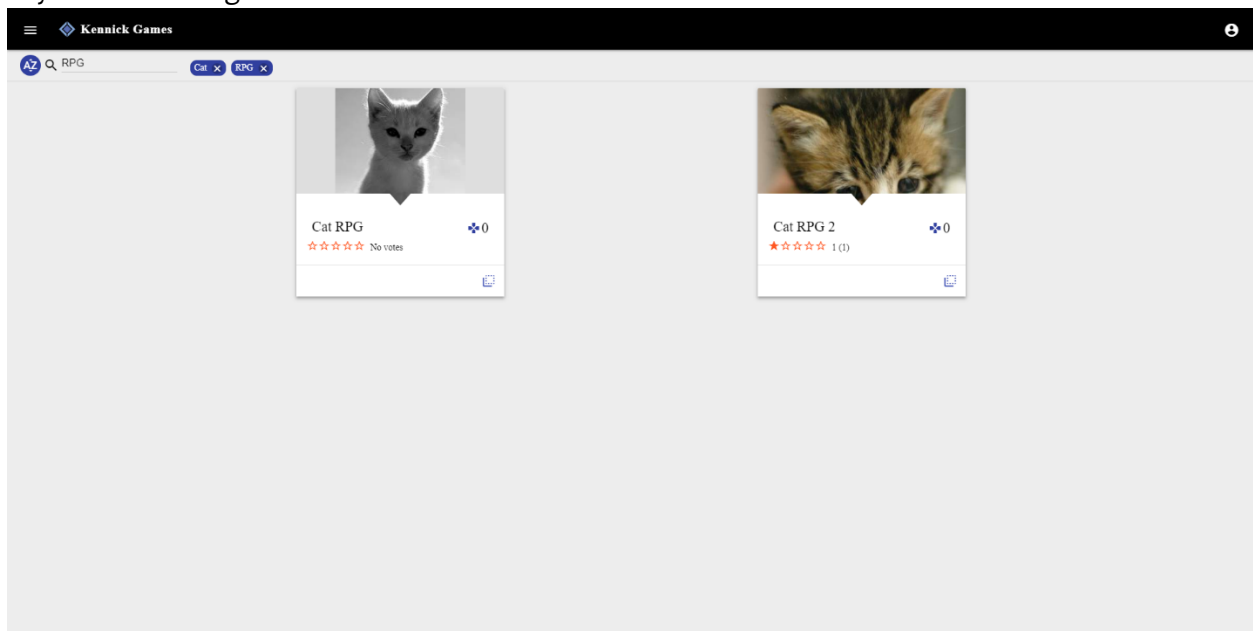
Het draaien van de tegel is geanimeerd. Op de achterkant van de tegel is een beschrijving beschikbaar.



Hier is ook een “Player” beschikbaar om het spel te spelen. Deze player neemt een groot deel van het scherm in beslag, maar is responsive.

Zoekfunctie

In het speloverzicht is een zoek functie beschikbaar. Deze is handgemaakt en maakt gebruik van keywords en categoriën.

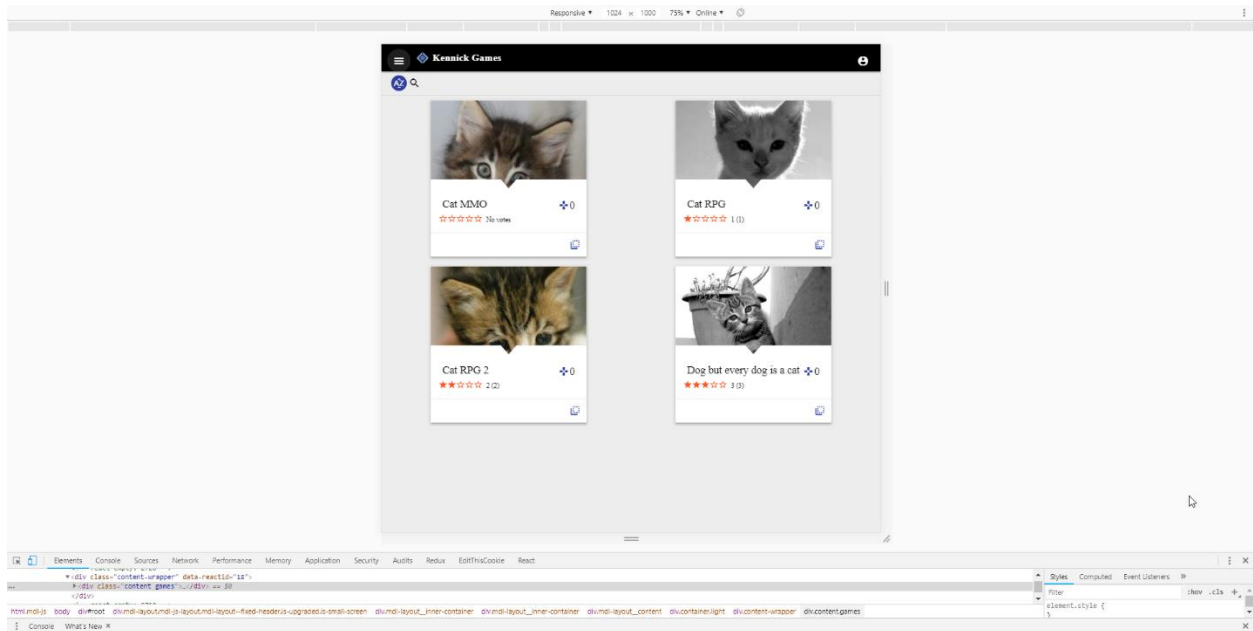


Dit is gedemonstreerd links boven in van dit scherm.

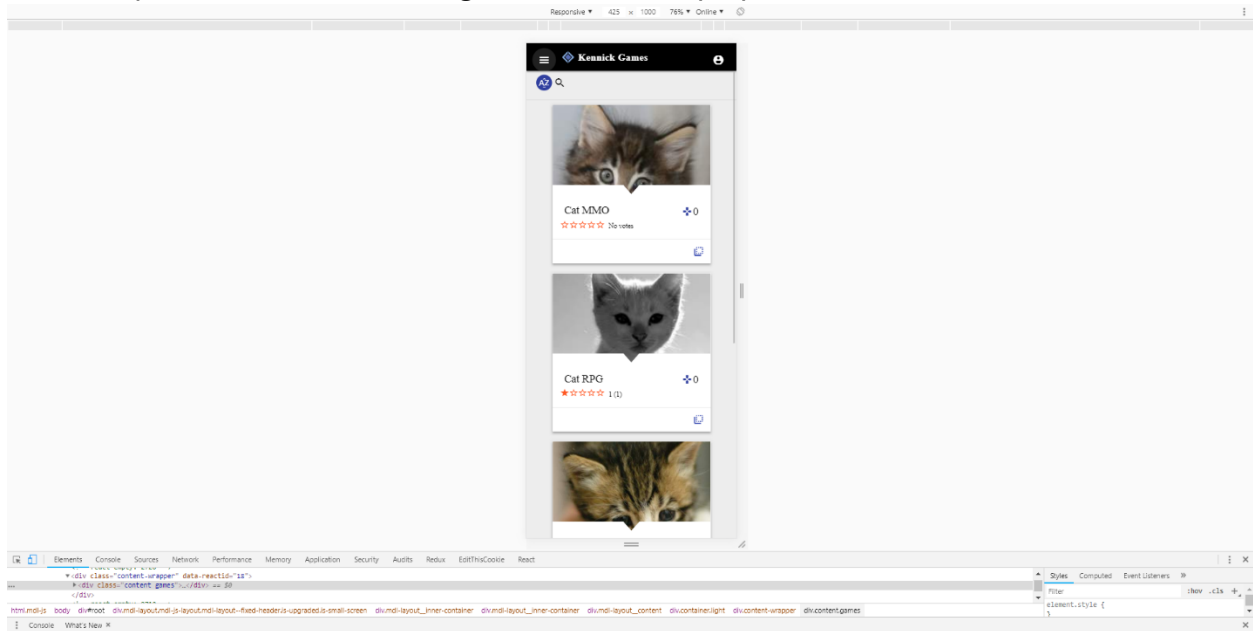
Responsive

1 van de eisen van dit project is dat het zelfde in gebruik moet worden genomen op elk soort scherm.

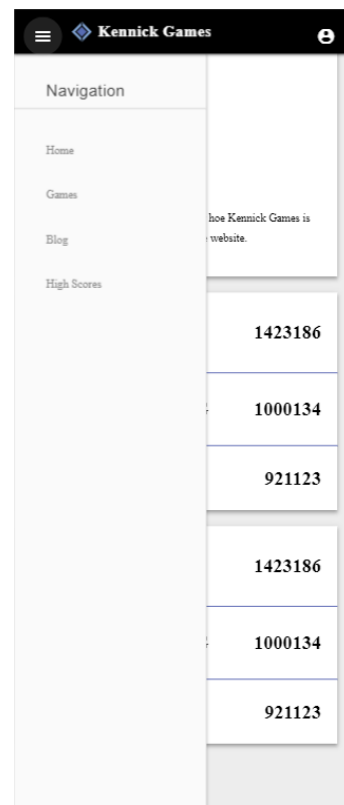
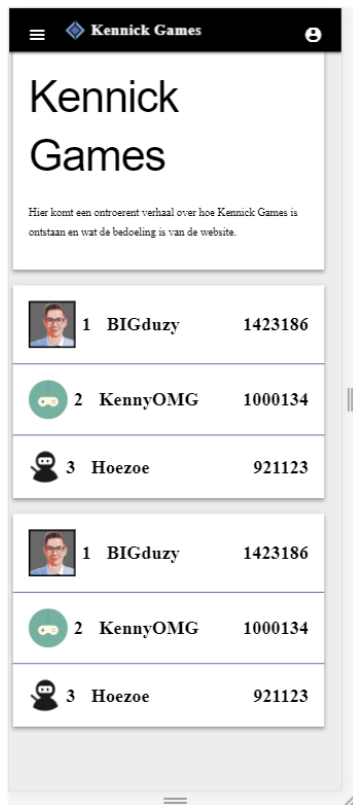
Ook moet dit makkelijk in onderhoud zijn en migratie naar een PWA (progressive web app) moet kunnen met het minst mogelijke wijzigingen. Hiervoor word onderandere Flexbox gebruikt.



Dit is het speloverzicht hoe het weergegeven word voor laptops met een klein scherm of tablets



Dit is het speloverzicht hoe het weergegeven word voor mobiele telefoons



Dit is het mobiele variant van de homepage. De spelers worden weergegeven in een lijst vorm. Als het menu wordt uitgeklaapt gaat het menu over de content heen ipv ernaast zoals bij grotere schermen.

Conclusie

Door gebruik van nieuwe technieken is responsiveness makkelijk te realiseren. Het overstappen van een applicatie in de browser naar een smartphone app is hierdoor ook gemakkelijk en onderhoud is minimaal sinds je de broncode maar 1 keer hoeft te schrijven.

Gebruik van een richtlijn document voor de design zoals Material Design zorgt ervoor dat je veel vrijheid hebt in je implementatie en dus kan implementeren exact zoals je wilt en zoals wanneer je de vormgeving helemaal zelf ontwikkeld. Al bespaar je veel tijd in ontwikkeling als je gebruik maakt van al gemaakte oplossingen zoals Bootstrap.