Daan Sperling Erik Knaake

605908 598368

Fletnix

Een dynamische website

Inhoud

[Inhoud 1](#_Toc503992946)

[Inleiding 2](#_Toc503992947)

[Functionele eisen 3](#_Toc503992948)

[Gerealiseerde functionaliteit 4](#_Toc503992949)

[Gegevens ophalen uit de database 4](#_Toc503992950)

[Filteren op films 4](#_Toc503992951)

[Beveiliging 4](#_Toc503992952)

[Schermafdrukken 5](#_Toc503992953)

[Feedback en knelpunten 8](#_Toc503992954)

[Feedback 8](#_Toc503992955)

[Knelpunten 8](#_Toc503992956)

Inleiding

Na het realiseren van een statische website. Is het nu de opdracht geworden om hiervan een dynamische website te maken. Om dit te realiseren, hebben we PHP code nodig gehad. Wat terug te vinden is op vele pagina’s in onze website.

Om een paar voorbeelden te noemen van de dynamiek op de website:

* Een directe link met een database.
* Een goed functionerende inlog functie/pagina.
* Alle films en hun details worden direct uit de database gehaald.
* Aanmelden wordt mogelijk gemaakt met een database.
* Alle abonnement vormen staan geregistreerd in een database.

Qua styling hebben we hetzelfde kleurenpalet gebruikt als de statische website. In het kort: een donker rode kleur en een donker blauwe kleur voor achtergrond. Voor de tekst hebben wij een witte kleur en lettertype Arial.

Functionele eisen

Van onze opdrachtgever hebben we een aantal functionele eisen gekregen voor de website om te realiseren. Al deze eisen zijn verwerkt en toegepast in de website. Hieronder vind u een lijst met deze eisen en onze oplossingen:

* *Minimaal 30 films zijn volledig uitgewerkt (titel, regisseur, cast, duur, jaar, poster, trailer, samenvatting).*

We hebben in totaal 31 films volledig uitgewerkt in de database, deze kunnen via de website worden opgevraagd

* *De gegenereerde HTML code valideert bij validator.w3.org.*

Alle code is gevalideerd en is zonder fouten of errors getest.

* *De filmposters staan in de eigen website, de trailers worden via externe links binnengehaald.*

Alle filmposters staan offline in een map en worden met PHP opgehaald. De trailers worden binnengehaald vanaf youtube.com en worden afgespeeld op onze site.

* *Er wordt gebruik gemaakt van PDO en sessionvariabelen.*

Alle gegevens die uit de database moeten worden opgehaald, worden met behulp van een *PDO* en *prepared statements* opgehaald. Als het nodig is om gegevens door te geven naar andere pagina’s maken we hiervoor gebruik van *sessionvariabele*n.

* *Terugkerende HTML code (b.v. voor de header of de footer) wordt met includes toegevoegd.*

Alle code die op meerdere pagina’s voorkomen, worden opgehaald met require\_once. Er is gekozen voor require\_once in plaats van include vanwege de snelheid en het voorkomen van problemen als een pagina vaker ‘geinclude’ zou moeten worden binnen een pagina.

* *Voor de databaseverbinding wordt een eigen bestand gebruikt die met require\_once daar waar nodig opgeroepen wordt.*

Zoals het eerder genoemde punt. Dit is gebruikt en werkt naar behoren.

* *PHP code wordt zoveel mogelijk van HTML code gescheiden.*

De standaard HTML tags worden geopend in de header en gesloten in de footer. Verder worden binnen andere pagina’s HTML code alleen gebruikt om het resultaat van de PHP code te weergeven. Functies en variabelen zijn zo veel mogelijk bovenaan een pagina verklaard.

* *Eigen variabelen en functies hebben een duidelijke Nederlandse naam.*

Alle gebruikte variabelen en functies zijn in het Nederlands en hebben een toepasselijke naam.

Gerealiseerde functionaliteit

# Gegevens ophalen uit de database

Zoals beschreven is in de opdracht beschrijving moeten de films uit de database worden opgehaald, en moet dit gebeuren met behulp van een PDO. Om dit te realiseren is er een bestand *database\_verbinding.php* gemaakt die de connectie naar de database tot stand brengt. Deze database bestaat uit de eerste 85 films, die binnen de genres: Sci-Fi, actie of komedie vallen, geproduceerd zijn na 1974, uit de TSV bestanden die te downloaden zijn op: <https://datasets.imdbws.com/>. Het grootste deel van de informatie is door C++ van deze TSV bestanden gehaald. Er is ook informatie, zoals de afbeeldingen, van <https://www.themoviedb.org/> gehaald. Van deze films zijn bij 31 films de trailers en de ontbrekende regisseurs en/of acteurs gezocht. In de database zijn ook de klantgegevens en de abonnementsvormen opgeslagen.

# Filteren op films

Er is gekozen om het filteren van films op meerdere manieren mogelijk te maken. Zo kan er gezocht worden op titel, regisseur. En kan er gefilterd worden op jaartal en genre. Bij deze laatste twee filters kan er ook worden gekozen om de bijbehoren *operators* te veranderen. Zo kun je zeggen film die als genre Sci-Fi én Actie hebben of films die Sci-Fi óf Actie als genre hebben. Bij het jaartal kun je bepalen of het geselecteerde jaartal voor, na of op (of een combinatie hiervan is) het publicatiejaar van een film is. De zoekterm(en) worden boven de resultaten weergegeven.

# Beveiliging

## Gebruikers

Om het wachtwoord van de gebruiker te beveiligen is er gekozen om een *salt* met een lengte van 64 byte te gebruiken en dit achter het ingevoerde wachtwoord te plakken. Dit resultaat wordt *gehasht* met het algoritme sha2 met een blok grootte van 512 bit. Dit algoritme is langzamer dan MD5 en daardoor aanzienlijk moeilijker te kraken, het is wel snel genoeg om als gebruiker geen last te hebben van het snelheidsverlies. De *salt* zorgt ervoor dat dezelfde wachtwoorden toch anders worden opgeslagen, terwijl er nog wel kan worden bewezen dat het goede wachtwoord is ingevoerd.

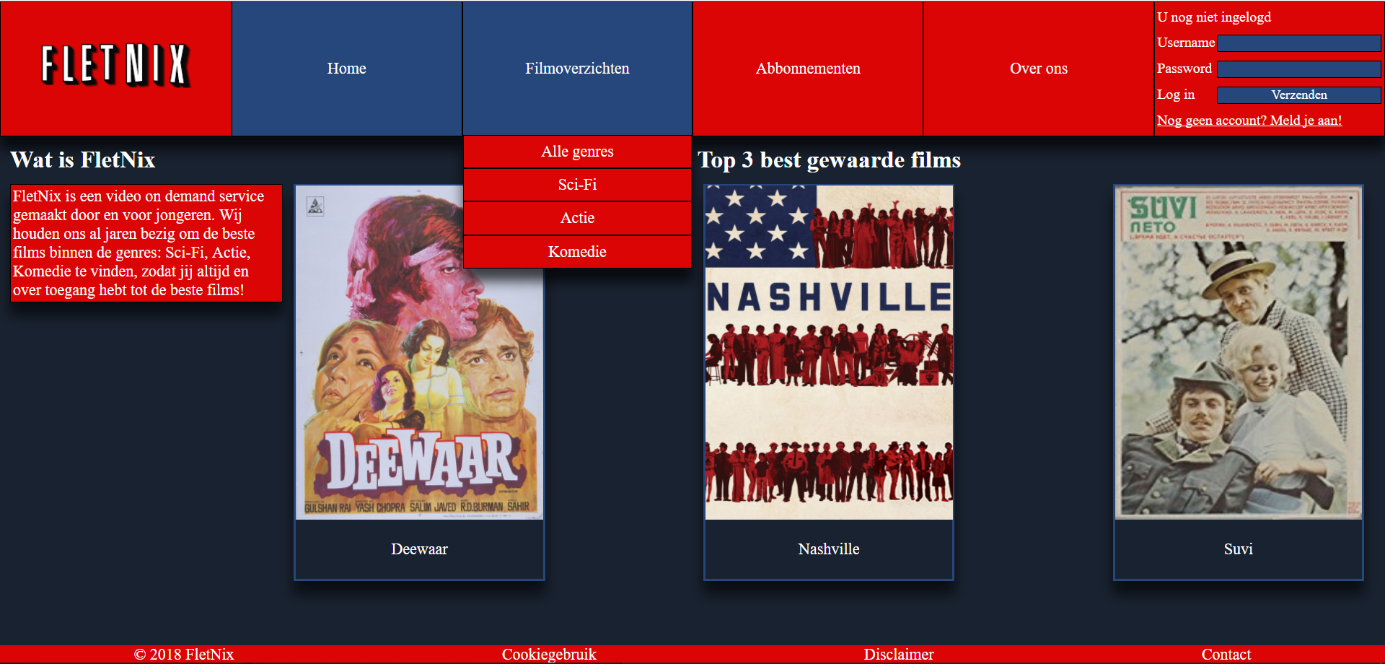
## Filters

Omdat er gekozen is de mogelijk te bieden om te kiezen dat een jaartal groter dan, gelijk aan, kleiner dan etc. is dan het publicatiejaar van een film, oftewel *operators* kunnen worden veranderd moet er voor worden gezorgd dat we wel zeker weten dat deze *operators* wel een valide operator zijn en geen SQL-injectie. Om dit te voorkomen gebruiken we een variabele met een operator als standaard waarde, die alleen veranderd wordt als de verzonden *operator* valide blijkt te zijn.

# Schermafdrukken

In het algemeen is een pagina opgebouwd uit drie tot vier delen, een header, footer, het hoofdonderdeel en eventueel een kleiner onderdeel ernaast. In de header vindt de gebruiker de meest belangrijke navigatie. Het is de link tussen de hoofdpagina’s, maar het is ook de plaats waar een gebruiker kan inloggen. Verder is een belangrijk deel van de navigatie de filmposter met titel, door hierop te klikken gaat u naar de details van deze film. Op deze pagina kunt u weer op het filmposter klikken om naar de afspeelpagina te gaan.

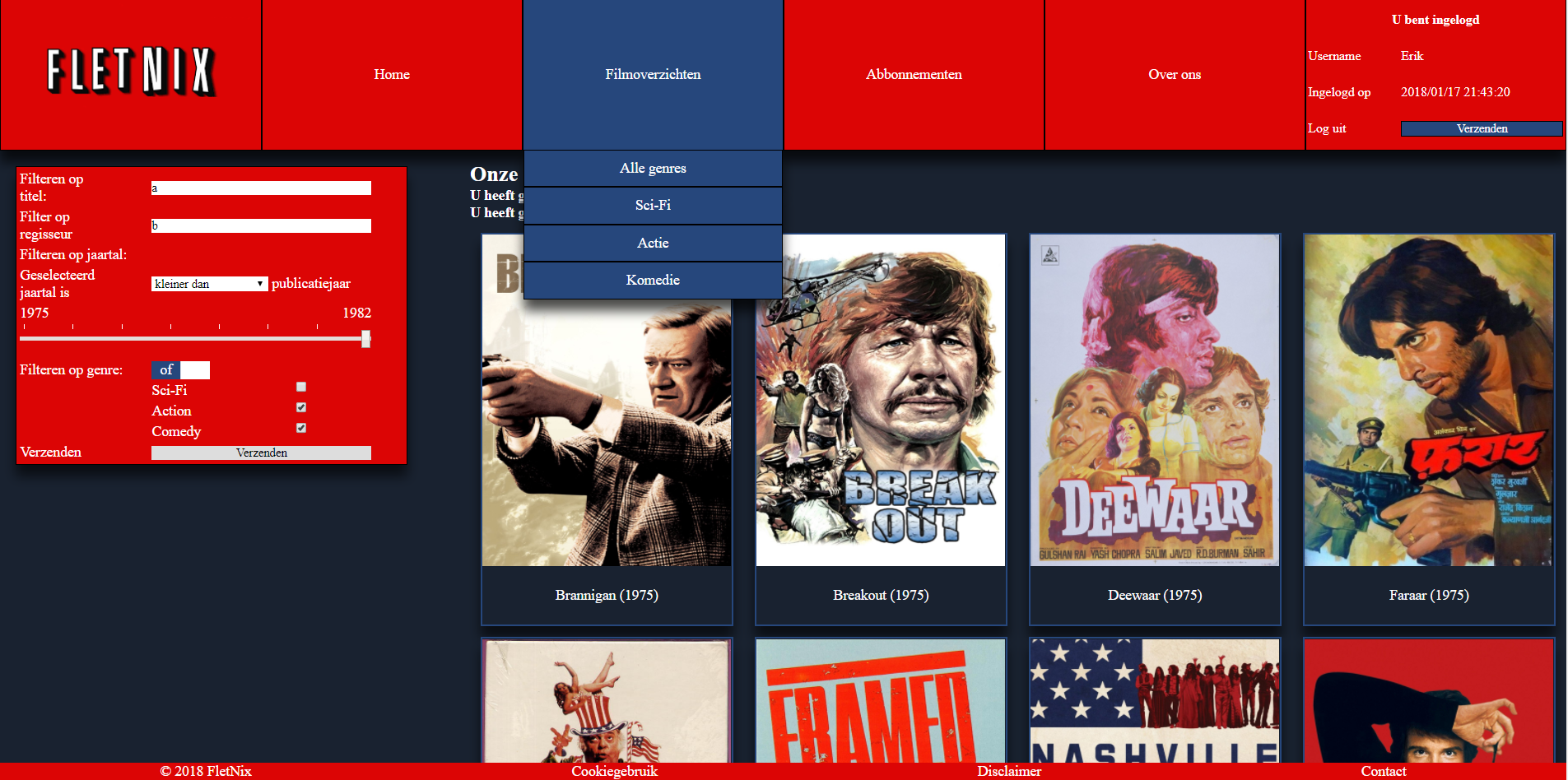
## Home



Figuur

De pagina waar de gebruiker voor het eerste zal komen, hier ziet uw de drie films die op dat moment het beste gewaardeerd worden. Verder staat hier een korte beschrijving van wat FletNix is.

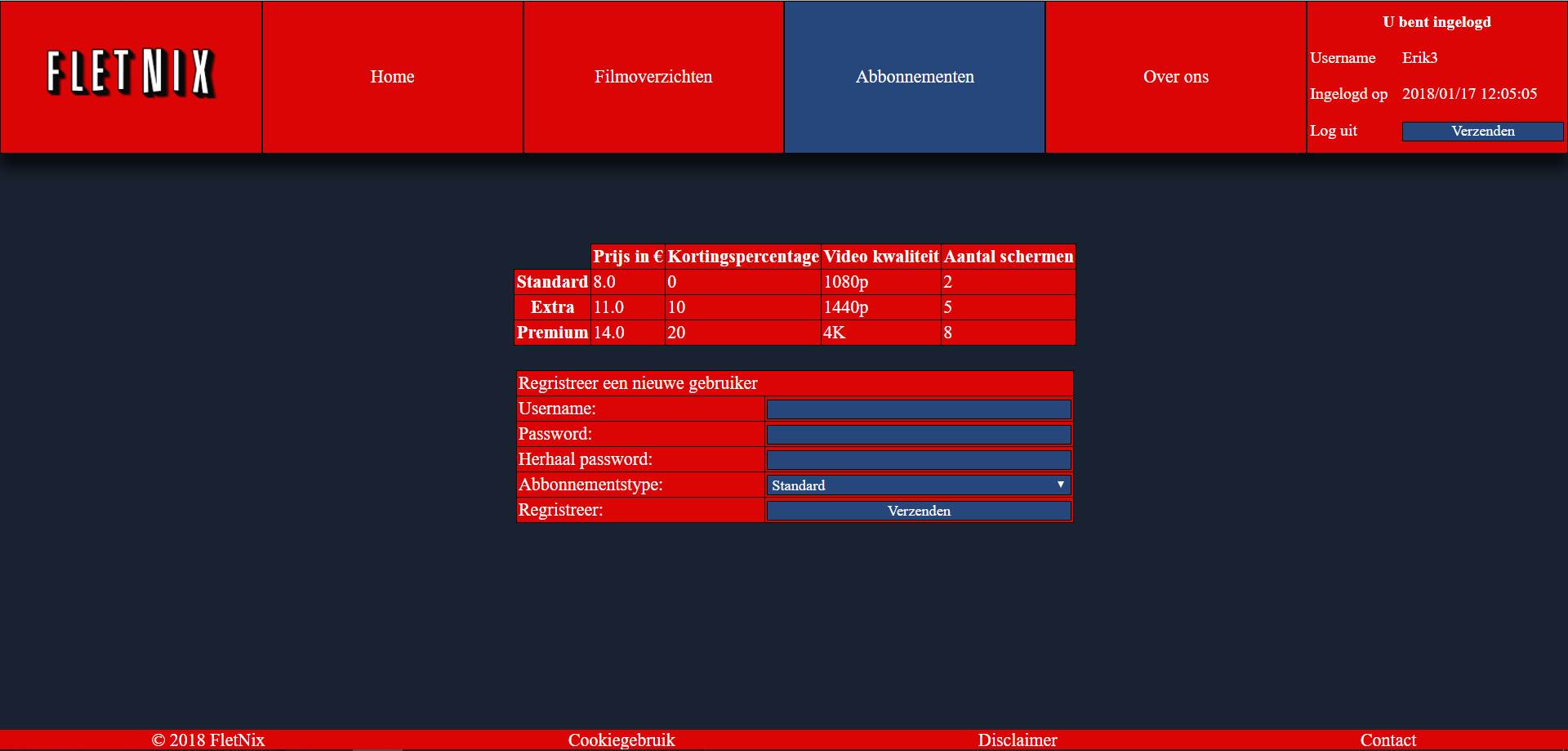
## Filmoverzicht



Figuur

Op deze pagina kan er worden gezocht naar een film. Er kan worden gefilterd zoals eerder beschreven. U kunt hier doorklikken om naar de detailpagina van een film te gaan.

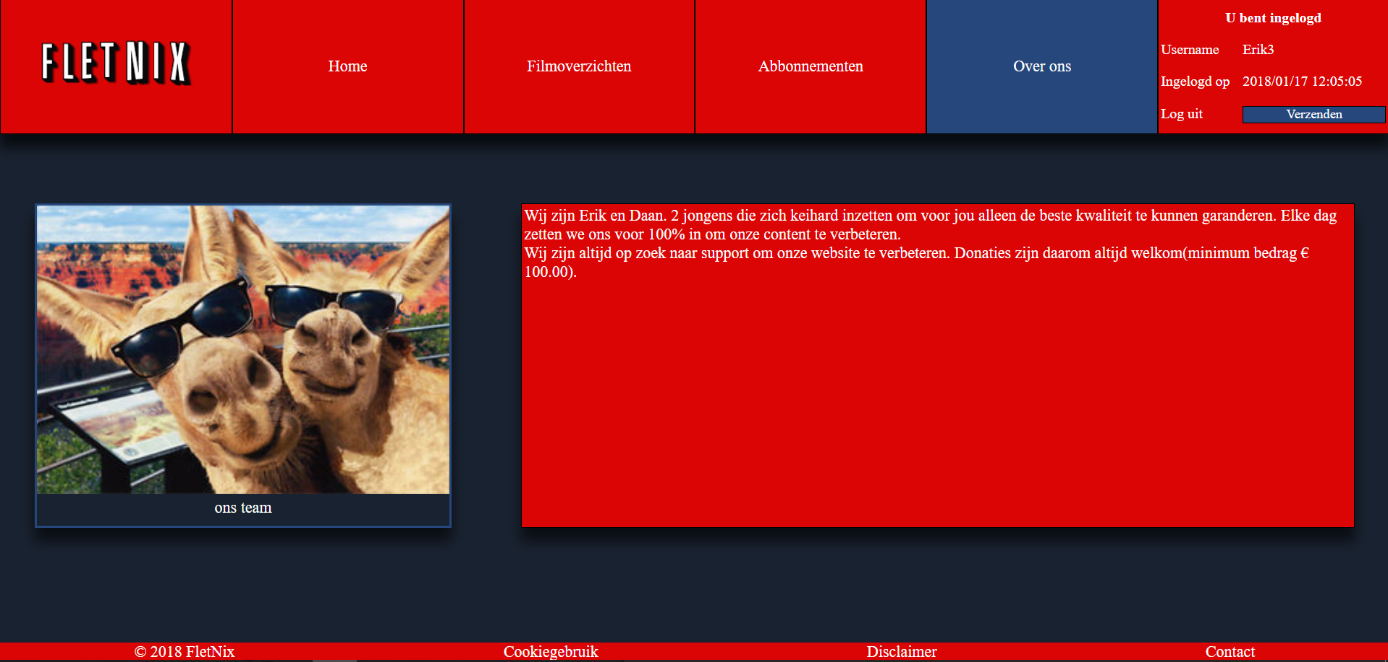
## Abonnementen



Figuur

Dit is de pagina waar een gebruiker kan zien welke abonnementen er zijn, en kan hij er een afsluiten.

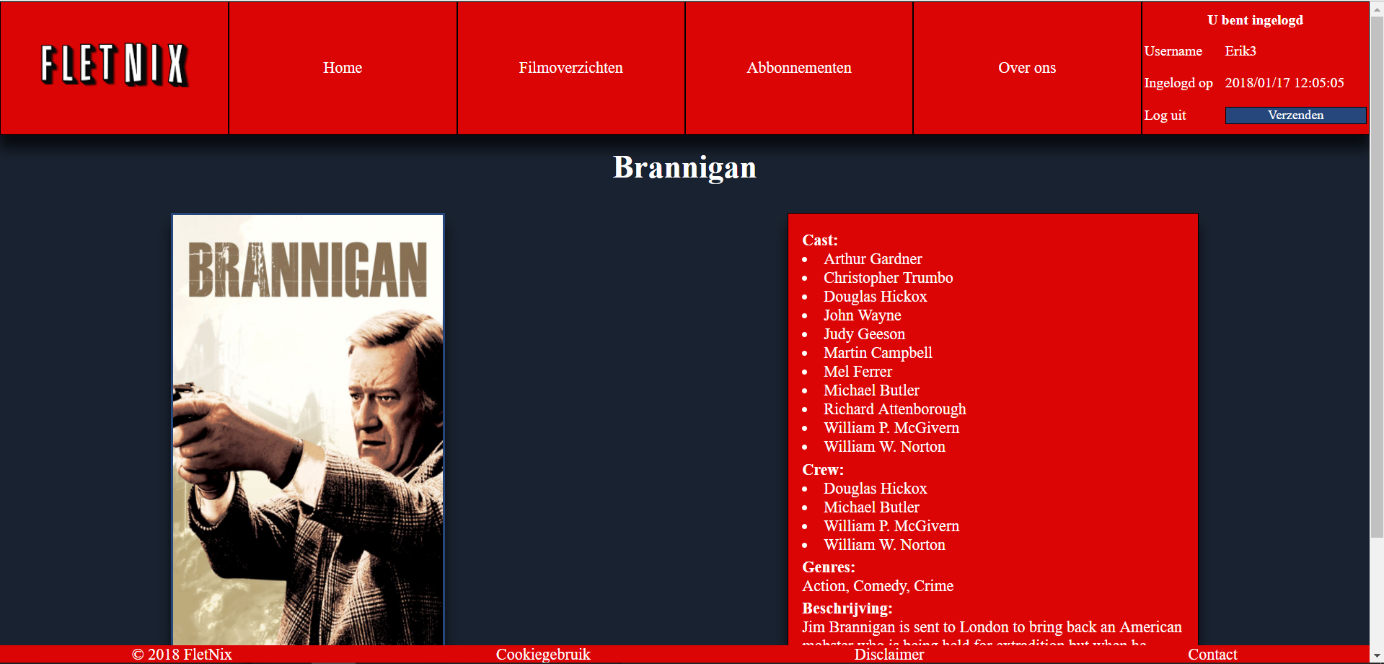
## Over ons



Figuur

Op deze pagina is te zien wie de website hebben gemaakt.

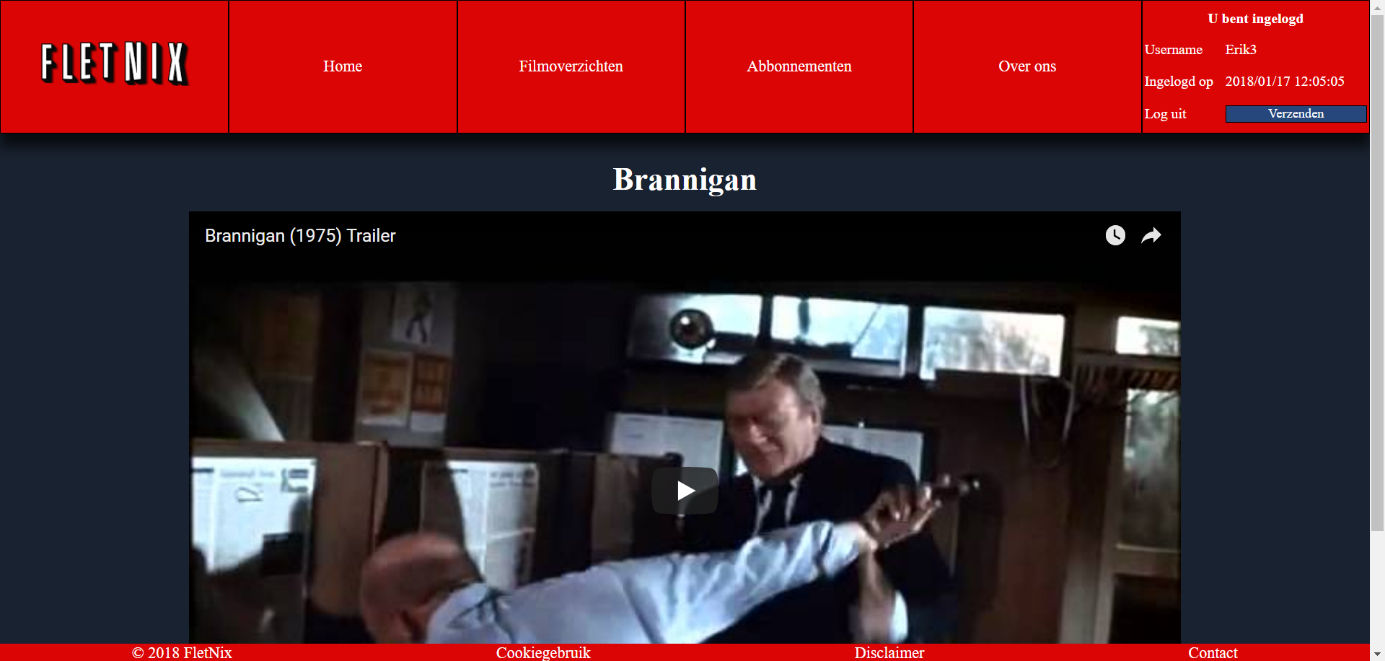
## Filmdetails



Figuur

Op deze pagina ziet u de details van een film, hieronder vallen de regisseur(s), acteurs, genre(s), beschrijving, waardering, publicatiejaar, titel en speelduur. U kunt hier doorklikken om de trailer te bekijken.

## Afspelen



Figuur

Op deze pagina kan de trailer van een film worden afgespeeld, deze trailer wordt opgehaald vanaf het *embed* gedeelte van <https://www.youtube.com/>.

Feedback en knelpunten

# Feedback

Vanuit de statische website hadden we een paar punten feedback gekregen om toe te passen op de dynamische website. Het eerste punt van kritiek was dat de knop filmoverzicht bedoeld was als dropdown box. In plaats daarvan hadden wij een die veranderd in 3 knoppen zodra de muis er op boven komt. In dynamische versie is dit een dropdown box geworden.

Verder is er gevraagd om meer geavanceerde CSS selectoren te gebruiken, zoals een child selector of een atribute selector. Deze zijn nu vaker gebruikt dan in de statische website.

Als laatste was er een advies gegeven om meer padding te gebruiken bij tekst, dit is gedaan zoals vooral te merken in de over ons en homepagina.

# Knelpunten

Ons belangrijkste knelpunt tijdens de realisatie van de website was de compatibiliteit met verschillende browsers. Veelal was deze compatibiliteitsproblemen te vinden bij de styling. Zoals de positionering van een element wat wel werkte binnen Google Chrome maar vervolgens niet werkte in Internet Explorer of Mozilla Firefox.

Verder hebben we enige moeite gehad met het schrijven van een databaseverbinding die op beide systemen werkte. Na een aantal pogingen is dit wel gelukt.

Conclusie

Zoals u kunt zien aan de scherm afdrukken, is het gelukt om de website dynamisch te maken.

# Figuren

[Figuur 1 5](file:///C:\Users\Erik%20Knaake\Documents\HBO-ICT\WT\BP%202\WT_BP210\Fletnix_Opleverdocument.docx#_Toc504060081)

[Figuur 2 5](#_Toc504060082)

[Figuur 3 6](#_Toc504060083)

[Figuur 4 6](#_Toc504060084)

[Figuur 5 7](#_Toc504060085)

[Figuur 6 7](#_Toc504060086)