


Naam:

Web 2 – Test op 25 mei 2023 (100% van je eindscore)

Deze test gaat door van thuis uit, we rekenen op ieders eerlijkheid en spreken het volgende af :

- **Je werkt de opdracht individueel uit, zonder hulp van anderen!**
 - communicatie met medestudenten of externen over de opdracht/oplossing is ten strengste verboden
 - zelfs als je klaar bent communiceer je niet met elkaar voor 22u00 (er kunnen nog studenten bezig zijn!)
 - je zit dus niet samen op dezelfde locatie, stuurt elkaar geen berichten, wisselt geen bestanden uit, deelt geen cloud drive folders, geeft elkaar geen hints, etc.
 - elk vermoeden van fraude zal bij de fraudecommissie worden gemeld (zie OER)
- De opdracht komt om 18u30 automatisch beschikbaar op Chamilo, bij onderdeel 'Opdrachten'
 - Je uploadt je oplossing bij diezelfde opdracht vóór het einde van de test
- **Het einde van de test is om 21u30**, op dat tijdstip moet je oplossing dus al op Chamilo staan
 - voor studenten met extra tijd (volgens hun IOEM) is het einde van de test om 22u00
 - Het exacte tijdstip van indienen is dat wat Chamilo rapporteert
 - let op, misschien loopt je eigen klok wel voor of achter!
 - **dien voor de zekerheid al 10 minuten eerder in**
 - zelfs een beetje te laat kost al punten
 - **oplossingen die meer dan 5min te laat zijn, worden niet meer aanvaard**
- Tijdens de test is er ondersteuning via Microsoft teams
 - je kunt vragen stellen aan de docent via de 1-op-1 chat
 - (dus niet via de gemeenschappelijke chat, dat stoort de anderen)
 - klik op icoontje  bovenaan of selecteer “Chat” in de linkerbalk
 - hou ook de gemeenschappelijke chat in de gaten.
 - (voor het geval je docent een belangrijk bericht post tijdens de test)
- Je mag het internet raadplegen tijdens de test
 - in zoverre dit geen communicatie met medestudenten of externen oplevert.
- Je oplossing gebruikt enkel code die je zelf schrijft tijdens de test (en die de standaard Javascript mogelijkheden benut). Je gebruikt geen libraries of eerder geschreven code.

Bij deze opdracht hoort een .zip file met de volgende inhoud

- dit .pdf document dat de opdracht beschrijft
- video 'movies demo short.mp4', demonstreert wat je moet bouwen
- afbeelding 'movies screenshot.png', een screenshot
- **folder 'beginsituatie movies'**

Stop eerst de inhoud van de **folder 'beginsituatie movies'** in een nieuw Webstorm project (of folder) met naam

'familienaam voornaam test movies'

Bijvoorbeeld : 'janssens jan test movies'

Bekijk nu het demonstratie filmpje 'movies demo short.mp4' zodat je weet wat je moet bouwen.

Je opdracht is de webapplicatie uit de video zo getrouw mogelijk na te bouwen.

Je krijgt ook een screenshot, voor het geval je exacte afmetingen nodig hebt en de kwaliteit van de video onvoldoende is.

De **folder 'beginsituatie movies'** (en bijgevolg nu ook je Webstorm project of folder) bevat de volgende inhoud:

- Index.html **(NIET AANPASSEN)**
 - dit is het HTML document dat deel is van je oplossing. Dit bestand pas je niet aan, HTML elementen zullen namelijk enkel via je JavaScript oplossing aangepast / toegevoegd / verwijderd etc worden. Binnen dit HTML bestand is aangeduid waar er verwacht wordt dat er elementen gemanipuleerd zullen worden (dit enkel binnen de elementen met ID "likeList" en "movieList").
- hardgecodeerd voorbeeld.html **(NIET AANPASSEN)**
 - dit is een statische HTML-pagina die je bij de opdracht krijgt, zodat je zou zien welke HTML-elementen en attributen je nodig hebt voor de films en de likes in de pagina. Het is zinloos dit document aan te passen, het hoort bij de opdrachtbeschrijving en doet niks in je oplossing.
- /script/movies.js **(NIET AANPASSEN)**
 - dit is een gegeven Javascript bestand dat een globale variabele '**movies**' definieert waarin de informatie over alle films zit. Per film is er een unieke id, een titel, de url van de bijbehorende afbeelding en een korte beschrijving.
- **/script/script.js**
 - **hier schrijf je al je Javascript code in.**
- /style/style.css **(NIET AANPASSEN)**
 - dit bevat alles nodige CSS regels voor de oplossing. Je kan in 'hardgecodeerd.html' zien welke CSS-classes e.d. er gebruikt worden om de gewenste layout te krijgen.
- /images **(NIETS AANPASSEN IN DEZE FOLDER)**
 - een folder met afbeeldingen van films, eentje per film. In de globale '**movies**' variabele (zie bestand 'movies.js') vind je de urls terug (relatieve paden naar deze afbeeldingen).

Kortom, je hebt alle bestanden die je nodig hebt voor de oplossing en **je mag ENKEL de volgende bestanden aanpassen:**

- script/script.js
 - let op, je mag 'script/movies.js' NIET aanpassen

Het bedenken van de nodige HTML- en CSS-code maakt GEEN onderdeel uit van deze opdracht, dat is allemaal al aanwezig in de beginsituatie van het project. Je moet enkel Javascript code schrijven.

De styling regels die nodig zijn vind je in 'style/style.css' en in '**hardgecodeerd voorbeeld.html**' zie je hoe de HTML gestructureerd moet worden om de gewenste layout te verkrijgen. In jouw oplossing zal je Javascript code dezelfde HTML-elementen moeten inlassen voor de films en likes, dus kijk goed in het hardgecodeerde voorbeeld wat er daar gebruikt werd per film en per like.

Je moet dus de Javascript code schrijven om de voorgedefinieerde films in de globale 'movies' variabele in de HTML-pagina te krijgen. De definities van deze variabele vind je in het bestand '**script/movies.js**'. Per film zal je (na het inladen van de pagina!) de nodige HTML-elementen aan de DOM-tree moeten toevoegen, precies zoals de voorbeelden in file 'hardgecodeerd voorbeeld.html'.

Je zult ook de like/unlike functionaliteit moeten schrijven, zoals het filmpje demonstreert.

Voor alle duidelijkheid: **je 'index.html' bestand mag geen voorgedefinieerde films of likes bevatten**, alles moet via Javascript code gerealiseerd worden.

Mocht je toch iets moeten aanpassen dat eigenlijk niet voorzien was in deze opdrachtbeschrijving (bijvoorbeeld extra HTML-elementen, extra CSS-classes, extra CSS-regels, etc), **dan documenteer je dit duidelijk via commentaar bovenaan het 'script/script.js' bestand**. Als je dit in een ander bestand schrijft zal het over het hoofd gezien worden.

Je oplossing moet zeker deze functionaliteit bevatten (let op puntenverdeling), zoals gedemonstreerd in de video:

- voor elke film uit **de globale variabele 'movies'** verschijnt de juiste inhoud in de pagina bij het inladen (niet hardgecodeerd in de HTML maar dynamisch ingelast d.m.v. Javascript!) **(6/20)**
- een klik op 'like' voegt de film toe in de like bar (zijkant) en verandert de link naar 'unlike' **(4/20)**
- een klik op 'unlike' verwijdert de film uit de like bar (zijkant) en verandert de link naar 'like' **(3/20)**
- de like bar verschijnt indien nodig **(2/20)**
- de like bar verdwijnt als er geen likes meer zijn **(1/20)**
- een klik op een like scrollt de pagina naar de juiste film **(1/20)**

En het moet er ook uitzien zoals in de video + screenshot (met de gegeven CSS-styling dus) **(3/20)**.

Een paar tips:

- Je kan een element verbergen door de 'display' CSS- property op 'none' te zetten. Als je de originele waarde terugzet, bv. 'block' in het geval van een <nav> element, dan wordt het element terug zichtbaar.
- Voor het scrollen moet je gewoon een unieke <a> anchor 'xyz' definiëren per film, en dan vanuit de like (in de like bar aan de zijkant) linken naar een url die eindigt op #xyz
 - je kan dit uitproberen door het bestand 'hardgecodeerd voorbeeld.html' in je browser te openen en op een afbeelding in de like bar (zijkant) te klikken.
- De layout is volledig gegeven, maar je oplossing kan er natuurlijk enkel correct uitzien als je de juiste HTML-elementen en/of CSS-classes en/of id's gebruikt zoals bedoeld was. Als je oplossing er niet goed uit ziet, vergelijk dan eens heel aandachtig de DOM-tree van je oplossing met die van de gegeven file 'hardgecodeerd voorbeeld.html' (bv. inspecteer hun DOM-tree via het 'Elements' tabblad in de Chrome Developer Tools).

Indienen

Helemaal bij het begin van deze opdracht maakte je een Webstorm project of folder met naam

'familienaam voornaam test movies'

Zip deze folder in een .zip file en geef deze de volgende naam :

'familienaam voornaam test movies.zip' (met weerom je eigen data natuurlijk)

Bijvoorbeeld : 'janssens jan test movies.zip'

Open voor alle zekerheid nog eens je .zip file en controleer dat je volledige oplossing erin zit. Als ik je ingediende file unzip, dan moet alles gewoon werken zonder hernoemen of kopiëren van files!

Dien tot slot je .zip file in op Chamilo bij de opdracht.

Denk eraan :

- **je oplossing moet absoluut vóór het einduur op Chamilo staan!**
- contacteer je docent via de 1-op-1 chat functie als je vragen hebt
(zodat je de andere studenten niet stoort)

Veel succes!