

Beoordelings Formulier: **WEBS3**  
Samenvatting eindopdracht: **MonsterZoo**  
Inleveren als .zip op blackboard

Datum:

Examinator:

	Primaire Functionaliteit
	Ondersteunende functionaliteit
	Complexe functionaliteit

Knockout criteria	k1	De applicatie is volledig met HTML/CSS en-of SASS)/Javascript gemaakt. (geen typescript, coffeescript)
	k2	De applicatie moet op elke machine werken, getest met Google Chrome
	k3	Er mogen geen Console errors voorkomen in de applicatie
	k4	Er moet minimaal 1 API call in de applicatie zijn verwerkt en worden aangeroepen dmv Fetch()
	k5	De applicatie bevat geen inline javascript
	k6	Er zijn geen frameworks gebruikt zoals Angular, React etc.

Nummer opbouw: Sectie.Week nummer.Toetsitem

Leerdoel(en):	Wat lever je op:	Nr.	Max. Score	Wanneer is het goed?	Eigen Score	Docent Score	Feedback / Motivatie
Je verrijkt, door middel van javascript, een statisch html document met dynamisch gedrag zodanig dat je een interactieve client side webapplicatie kan ontwikkelen	<b>1 - Applicatie Code</b>						
	Code (docent controle)	1.1.1	5	Je code is goed leesbaar, overdraagbaar, met juiste benaming en vrij van onnodig commentaar.	5		
	Code (docent controle)	1.1.2	3	Er wordt op een logische manier gebruik gemaakt van functies en ze zijn niet afhankelijk van elkaar	3		
	Code (uitleg student)	1.2.1	3	Je maakt correct gebruik van Eventlisteners zodanig dat de applicatie kan reageren op de verschillende interacties van de gebruiker.	3		
	Code (uitleg student)	1.2.2	6	Er is DOM manipulatie aanwezig via Javascript zodanig dat er alleen de elementen op de pagina aanwezig zijn die op dat moment nodig zijn. (Monster creatie, velden worden stap voor stap getoond of verwijderd / voor ingevuld bij aanpassen)	6		
	Code (uitleg student)	1.2.3	3	Je kan het gebruik van callbacks uitleggen en gebruik ervan in de code tonen	3		
	Code (uitleg student)	1.3.1	5	Je identificeert de belangrijkste plekken waar iets fout kan gaan zodanig dat er een correcte afhandeling (systeem & gebruiker) is geïmplementeerd.	5		
	Code (uitleg student)	1.4.1	5	De applicatie maakt gebruik van object georiënteerde programmeer technieken. Je kan uitleggen waar, waarom en welke technieken gebruikt worden	5		
	Code (uitleg student)	1.4.2	8	De applicatie heeft een Model-View-Controller (of vergelijkbaar) software architectuur. Je kan uitleggen wat de verschillende lagen doen / hoe je dit hebt geïmplementeerd	8		
	<b>2 - Functionaliteit Applicatie</b>						
	Func. Test (Docent controle)	2.2.1	6	Het is mogelijk om een eigen monster visueel te creëren (tekenen/uploaden) doormiddel van het Canvas element / Canvas wordt gebruikt voor andere visuele oplossingen zoals achtergronden etc.	6		
	Func. Test (Docent controle)	2.2.2	8	Het is mogelijk om monsters op het grid te plaatsen/verwijderen/aan te passen doormiddel van Drag&Drop inclusief gestelde restricties/specificaties.	8		
	Func. Test (Docent controle)	2.3.1	3	Het is mogelijk tussen regio's te wisselen en daarbij de verschillende geplaatste monsters te zien. (onthouden van de locatie)	3		
	Func. Test (Docent controle)	2.3.2	8	Er is interactiviteit tussen de geplaatste monsters, de regio en een enkele monster (zie specificaties)	8		
	Code (Uitleg Student)	2.4.1	3	Er wordt gebruik gemaakt van Localstorage om de benodigde data op te slaan	3		
	Func. Test (Docent controle)	2.x.1	9	De uitwerking voldoet aan de specificaties van de eindopdracht zoals beschreven in het webdictaat. (denk aan, dynamisch formulier, het weer, extra kracht etc.)	9		
	Student uitleg	2.5.1	10	De applicatie is modulair opgebouwd en maakt gebruik van Webpack om alles netjes te bundelen	10		
	Student uitleg	2.6.1	10	Je hebt een Webcomponent volgens de Shadow Dom principes geïmplementeerd			
	Student uitleg	2.6.2	5	Je hebt een extra library / WebAPI gebruikt om de applicatie interactiever/visueel aantrekkelijker te maken.	5		
<b>Eindcijfer</b>					90		

