PXL DIGITAL	**Hogeschool PXL** Departement Digital Academiejaar 2022-2023 – semester II **EXAMEN**

WebAdvanced (DEEL 1 van 1)

Student	
Naam student **Voornaam student** **Klas** **Lector** **Examenlokaal**	…… …… **1 GRAD PRO** **LUC DOUMEN – JASPER BEULS** ……
Examen	**WebAdvanced – Praktijkexamen**
Resultaat	**/ 20**
Periode	~~Kwartaal 1~~ | ~~Kwartaal 2~~ | ~~Semester 1~~ | **Semester 2** | ~~Herexamen~~
Datum	**13/06/2023**
Tijdstip (aanvang + einde)	**08u30 – 11u30**
Klas	***1 GRAD PRO A- B- C- D- E- F- G- H- I- J- K- L***
Lectoren	**LUC DOUMEN – JASPER BEULS**
Samenstelling examen	
Puntenverdeling	**Vraag 1 (11ptn) – Vraag 2 (9ptn)**
Tijdsverdeling	**NIHIL**
Samenstelling bundel + praktijkexamen	**Examendeel 1 van 1**
Pagina's	**xx pagina's**, inclusief voorbladpagina's
Totaalscore	**/20p**
Toegelaten hulpmiddelen	~~Rekenmachine~~ - **laptop** – ~~internet~~ – **cursusmateriaal** - …
Opmerkingen	**NIHIL**
Digitaal beginbestand	**JA**/~~NEE~~
Digitale indiening	**JA** / ~~NEE~~

Noteer hier de MD5 filehash van je digitale oplossing.*

Een digitale oplossing is geldig als:

- ☐ ***de oplossing is geüpload naar de centrale FTP-server.***
- ☐ ***de MD5 op deze pagina overeenkomt met je examenoplossing****
- ☐ ***de naam van de file = "Achternaam_Voornaam_Klas_Vak_Examenlokaal" ****

**In de examenbundel vind je meer info over upload, md5 en naamgeving van de digitale oplossing.*

Checklist voor het afleggen van examen

- ☐ **Gsm, smartphone en andere smart devices liggen uitgeschakeld op tafel**
- ☐ **Geen gsm of smartphone in jassen of tassen**
- ☐ **Jassen en tassen vooraan of achteraan het lokaal leggen**
- ☐ **Bij laptopexamen strikt de opgelegde regels volgen**
- ☐ **Tijdens het examen staat de vliegtuigstand aan**
- ☐ **Bundel met examenvragen blijft steeds samengeniet**
- ☐ **Toegestaan: 1 drankje in hersluitbaar flesje**
- ☐ **Iedereen zwijgt tijdens het examen**
- ☐ **Niet vergeten op je examenkopij te vermelden**
    - ☐ **de naam van lector**
    - ☐ **jouw naam**
    - ☐ **plaatscode (vermeld op je tafel of op het overzicht in het lokaal)**
    - ☐ **indien digitaal examen: *MD5 checksum***

Hogeschool PXL
Departement PXL-Digital
Elfde-Liniestraat 24 · 3500 Hasselt
digital@pxl.be · www.pxl.be

Instructies voor digitale examens met FileZilla

Gegevens FTP-server

IP adres: **10.50.50.161**
Gebruikersnaam: **41PRO1070**
Wachtwoord: **Ex%130623**

Start examen

Voor dit examen moet **een** startbestand afgehaald worden.

- Download het startbestand van de opgave folder op de FTP-server.
- Zie stap 2 'uploaden van je examenoplossing' voor de werkwijze.

Tijdens het examen

- Tijdens het examen staat je laptop in vliegtuigmodus.
- Connectie met een netwerk is niet toegestaan.

Einde examen = uploaden examenoplossing

Stap 1: Lokaal op je computer

- Comprimeer je examenoplossing tot één zip-bestand
- Controleer of het zip-bestand jouw oplossing bevat!
- Bereken de MD5-checksum van je zip-bestand. *
- Noteer de MD5-checksum op de voorpagina van je examenblad. Verander vanaf nu NIETS meer aan de inhoud van je examenoplossing!
- Verander de bestandsnaam: **"Achternaam_Voornaam_Klas_Vak_Examenlokaal"**
    - Examenlokaal staat vermeld op je examensteekkaart.

Stap 2: Uploaden van je examenoplossing

- Maak verbinding met het **PXL-wifinetwerk.**
- Open FileZilla client.
- Maak verbinding met de FTP-server. Gebruik bovenstaande gegevens.
- Upload je examenoplossing in de folder 'inzending'.
- Uitzondering: Voeg 'versie2' toe aan je bestandsnaam als je een nieuwe versie wil uploaden. Noteer juiste MD5 op het examenblad!

Stap 3: Controle

1. Je examenoplossing is zichtbaar tussen de files op FTP-server.

* MD5 van je oplossingsfile (zip) bepalen

- OSX: via het commando **md5** in een terminal-venster.
- Linux: via het commando **md5sum** in een terminal-venster.
- Win: Powershell ***Get-FileHash*** *-Path .\path\bestandsnaam.zip -Algorithm md5*

Een digitale oplossing is enkel geldig als

- ***de oplossing is geüpload naar de centrale FTP server***
- ***de MD5 op voorpagina overeenkomt met je examenoplossing.***
- ***de naam van de file = "Achternaam_Voornaam_Klas_Vak_Examenlokaal"***

V O O R W O O R D

- Je krijgt code toegestuurd via FileZilla bij de aanvang van het examen
- Je **moet** hier **gebruik** van **maken** om de opdracht op te lossen
- Na het examen geef je af via FileZilla door je digitale oplossing door te zenden
- De naam van het af te geven bestand moet in de naamgeving van de vorm zijn:

  Achternaam_Voornaam_Klas_WebAdvanced_Examenlokaal.zip

- Controleer na het maken van de **.zip** dat alle bestanden aanwezig zijn en dat je het juiste bestand via FileZilla doorstuurt
- *Noteer de MD5 filehash van je digitale oplossing hierboven in het daarvoor voorziene vakje*
- Plaats in elk **.js**-bestand je naam en voornaam in commentaar. Aangevuld met de (javascript) code die jouw oplossing is. Het spreekt dus voor zich dat enkel dit bestand verbeterd wordt.
- De **verbetering van het examen** zal beoordeeld worden aan de hand van werkende (onder)delen van het programma. Niet-werkende code zal geen punten opleveren.
- Dit examen is opgesteld om op 3h tijd opgelost te worden.

VEEL SUCCES !!!

Oefening 1 DOM & Event handling 7pnt

Maak de oplossing in de map oefening1. Hierin vind je ook de file index.html. Aan deze file mag je niets wijzigen. Alle code schrijf je in js/oefening1.js.
In dit Javascript bestand staat er al code die je moet gebruiken.
De bedoeling van deze website is om de gebruiker de kans te geven om een landenquiz te genereren. Hiervoor krijg je een array met vragen, mogelijke antwoorden per vraag en het antwoord op de vraag.
De gebruiker kan in de website het aantal vragen ingeven in het inputveld.

Website:

Als de gebruiker op de knop 'Genereer vragen' klikt moet volgende gebeuren:

- Indien het getal groter is dan 5 wordt er een foutmelding weergegeven in een messagebox (hint: alert) en met als boodschap "Er zijn max 5 vragen".
- Bij een juiste ingave genereer je random het opgegeven aantal vragen.
  Tip: let keuzeIndex = parseInt(5*Math.random());
  De vragen zet je binnen het output veld en je voorziet telkens 3 buttons met mogelijke antwoorden.

Één vraag kan meerdere keren voorkomen in de gegenereerde vragen, dit moet je niet filteren.

De vragen van de quiz worden in het javascript oefening1.js meegegeven en deze moet je gebruiken.

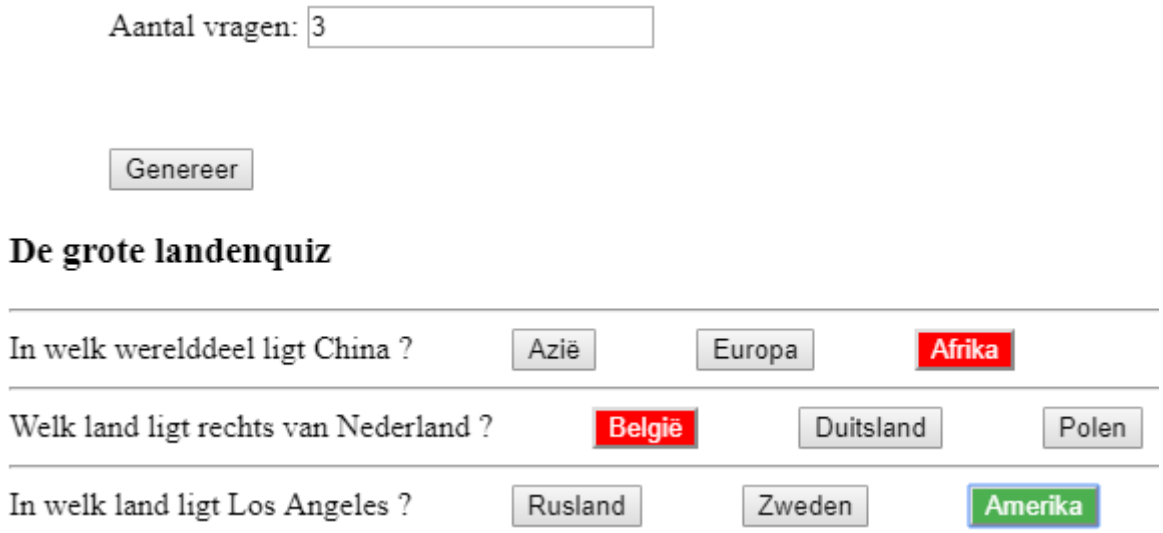
De HTML-code die in JavaScript gegenereerd wordt is van de vorm:

```
<h3>De grote landenquiz </h3>
<hr>
<label> Van welk land is Moskou de hoofdstad ?</label>
  <button name="Kroatie" value="Rusland">Letland</button>
  <button name="Rusland" value="Rusland">Rusland</button>
  <button name="Spanje" value="Rusland">Spanje</button>
<hr>
<label> Welk land heeft de vorm van een laars ? </label>
  <button name="Engeland" value="Italie">Engeland</button>
  <button name="Italie" value="Italie">Italie</button>
  <button name="Denemarken" value="Italie">Denemarken</button>
<hr>
….
```

In de hierboven gegenereerde HTML-code wordt **bij elke button** het juiste antwoord op die vraag in het **<u>attribuut</u> value** geplaatst.

Aan elke gegenereerde button wordt dezelfde eventlistener voor het click event geplaatst. Als dit event zich voordoet, controleer je het antwoord. (**<u>Tip: attributen name en value van de button vergelijken</u>**). Indien het antwoord fout is kleur je dit met een rode achtergrond en witte letters, indien het antwoord juist is kleur je

het met een groene achtergrond en witte letters. (Tip: event.target en classen uit de css).
Eénmaal geklikt en gekleurd, verandert de knop niet meer van kleur. (dus indien eerst fout en dan juist heb je een rode knop en een groene knop).



Oefening 2: JavaScript functions 4pt

De output van de volgende functies log je steeds in de console.

Functie 1:

Schrijf een JavaScript-functie genaamd filterLetters die een naam en een array met te filteren letters ontvangt.
De functie moet een nieuwe string retourneren waarbij alle letters uit de oorspronkelijke naam zijn verwijderd die voorkomen in de array lettersToFilter. De functie moet altijd werken, het mag niet uitmaken of er nu een hoofdletter of een kleine letter in de naam of de array staat.

Voorbeeld: filterLetters(‘Charlie’, [‘a’ , ’e’ , ‘c’]);
Output: hrli

Functie 2:

Schrijf een JavaScript-functie genaamd nameAndCitySortedByAge die een **nieuwe** array retourneerd op basis van de students array. Deze nieuwe array moet gevuld zijn met objecten die enkel de name, city en age properties uit de students array hebben, maar in het Nederlands. En de array moet gerangschikt zijn van oud naar jong.

Voorbeeld: nameAndCitySortedByAge();
Output:
[

{ naam: 'Frank', stad: 'Houston', leeftijd: 23 },

{ naam: 'Bob', stad: 'San Francisco', leeftijd: 22 },

{ naam: 'Isaac', stad: 'Dallas', leeftijd: 22 }, …
]

Functie 3:

Schrijf een JavaScript-functie genaamd subjectCount. Deze functie gaat over de students array en retourneerd een nieuwe array met voor elk uniek subject een object met de naam en het aantal studenten die ingeschreven zijn voor dat vak.

Voorbeeld: subjectCount();
Output:
[

{ subject: 'Math', count: 4 },

{ subject: 'Physics', count: 5 },

{ subject: 'Chemistry', count: 5 }, …

]

Functie 4:

Schrijf een JavaScript-functie genaamd getStudentByGradeAndNameLength eerst in de students array gaat zoeken naar de student met de hoogste grade.
Vervolgens zal de lengte van de naam van die student als index dienen om de returnwaard van de functie te bepalen.
Indien de index die net berekend werd binnen de array valt wordt de student op die index geretourneerd. **Indien Bob de hoogste score heeft moet de derde student uit de array geretourneerd worden. Want Bob is 3 letters lang**. Indien de index buiten de array valt moet de student met de hoogste grade geretourneerd worden.

Voorbeeld: getStudentByGradeAndNameLength();
Output:
{

name: 'Emily',

age: 20,

subjects: ['English', 'Art', 'Music'],

city: 'Boston',

grade: 88

}

Oefening 3 VueJS 9pnt

Aan het einde van het document staan foto's van hoe het eindresultaat er uit moet zien. Ga hier eerst eens naar kijken.
Alle functionaliteiten die hier worden gevraagd moeten dynamisch werken. Al worden er morgen 20 extra katten toegevoegd moeten de gevraagde functionaliteiten nog altijd werken. Werkt je code dan niet meer zal je punten afgetrokken krijgen. Gebruik waar nodig de correcte directives.

1. In de startbestanden in de map vue staat een map assets. Plaats deze afbeeldingen in jouw project onder src/assets.
2. In de startbestanden in de map vue staat een file cats.js, dit is een pinia store. Kopieer de file naar je stores map.

CatsView:
Maak een nieuwe file CatsView.vue aan.

1. Zorg ervoor dat er een navigatielink bijgemaakt wordt in App.vue die linkt naar de CatsView view.

2. In CatsView bereken je op basis van de lijst katten in de pinia store de gemiddelde prijs van een kat.

3. Maak een CatsList component aan en render deze in de template van CatsView.
   De CatsList component krijgt als prop een titel mee vanuit de CatsView view. Namelijk de tekst "Een gemiddelde kat kost… " (zie screenshot). De gemiddele prijs moet berekend worden in de pinia store en moet daarna opgehaald worden uit de pinia store door de CatsView view, die op zijn beurt de rest van de zin eraan plakt en deze doorgeeft aan de CatsList component.

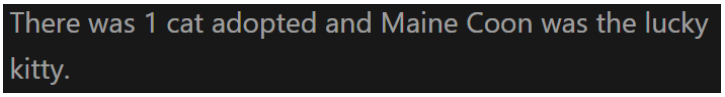
CatsList:

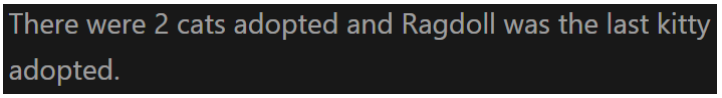
1. De CatsList component toont bovenaan een <h2> element met daarin de tekst met de gemiddelde prijs die uit de parent component komt.

2. Voor elk van de katten in de cats array in de pinia moet er een element gerenderd worden. Deze div toont links de foto van de kat, met daarnaast de naam en de prijs van de kat. En een knop om de kat te adopteren. De tekst voor de knop staat in commentaar in de pinia store, deze mag je gewoon kopiëren naar de template van CatsList.

3. Als er op de knop geklikt wordt moeten er 2 dingen gebeuren:

Als eerste:
In **CatsView.vue** moet boven de CatsList component een <h1> element gerenderd worden.
Dit element mag ENKEL gerenderd worden als er minstens 1x op een knop geklikt is.
De tekst die in de h1 komt te staan verschilt afhankelijk van hoeveel katten er al geadopteerd zijn (dus hoe vaak er al op een knop is geklikt).



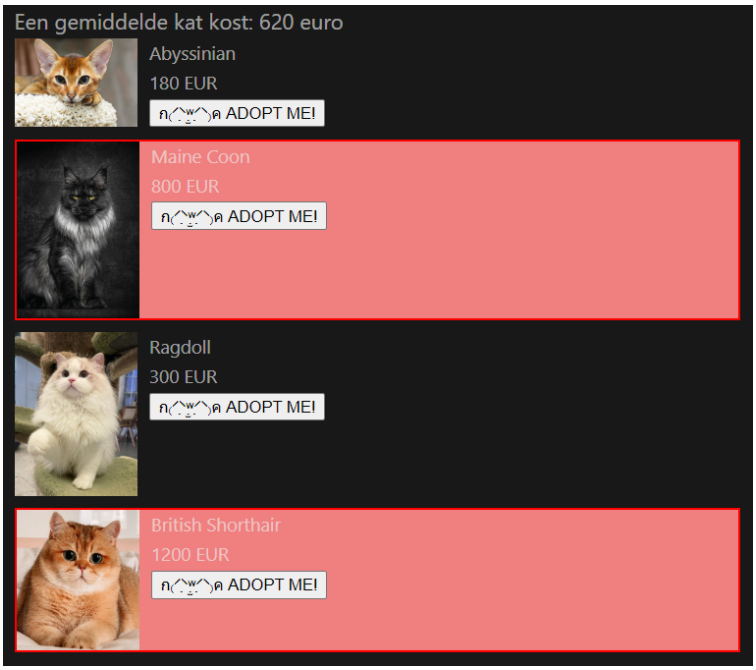


De naam van de laatst geadopteerde kat moet mee in de h1 komen te staan zoals in de screenshot hierboven.

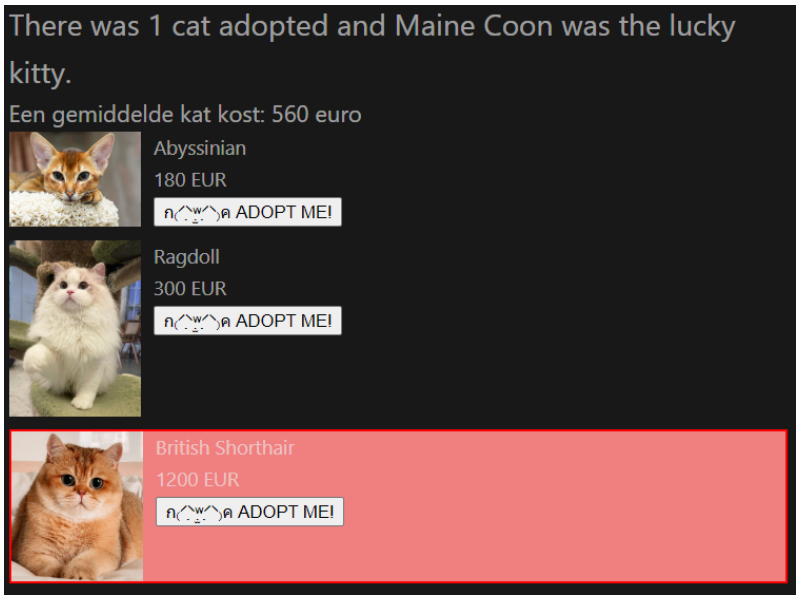
Ten tweede:
Als op de knop geklikt werd moet de kat waar de knop bij hoorde uit de array in pinia worden verwijderd.

4. Als een kat meer dan 500 euro kost krijgt de div met de info van die kat een background-color: light-coral en een rode volle border van 2px dik.

Screenshots:
Standaard view:



Als de Maine Coon geadopteerd is:



Als er 2 katten geadopteerd zijn:

Heel veel succes!!