Opdracht 1

2. Request

3. Server

4. Response

Internet

1. Client

Opdracht 2

https://www.bol.com/nl/p/hoe‐werkt‐dat‐nou/9200000057347012/?country=BE&suggestionType=browse#product\_alternatives

protocol

subdomein

host

path

parameters

fragment

Opdracht 3

Network tab

Afbeelding met tekst, schermafbeelding, binnen, monitor

Automatisch gegenereerde beschrijving

Response body

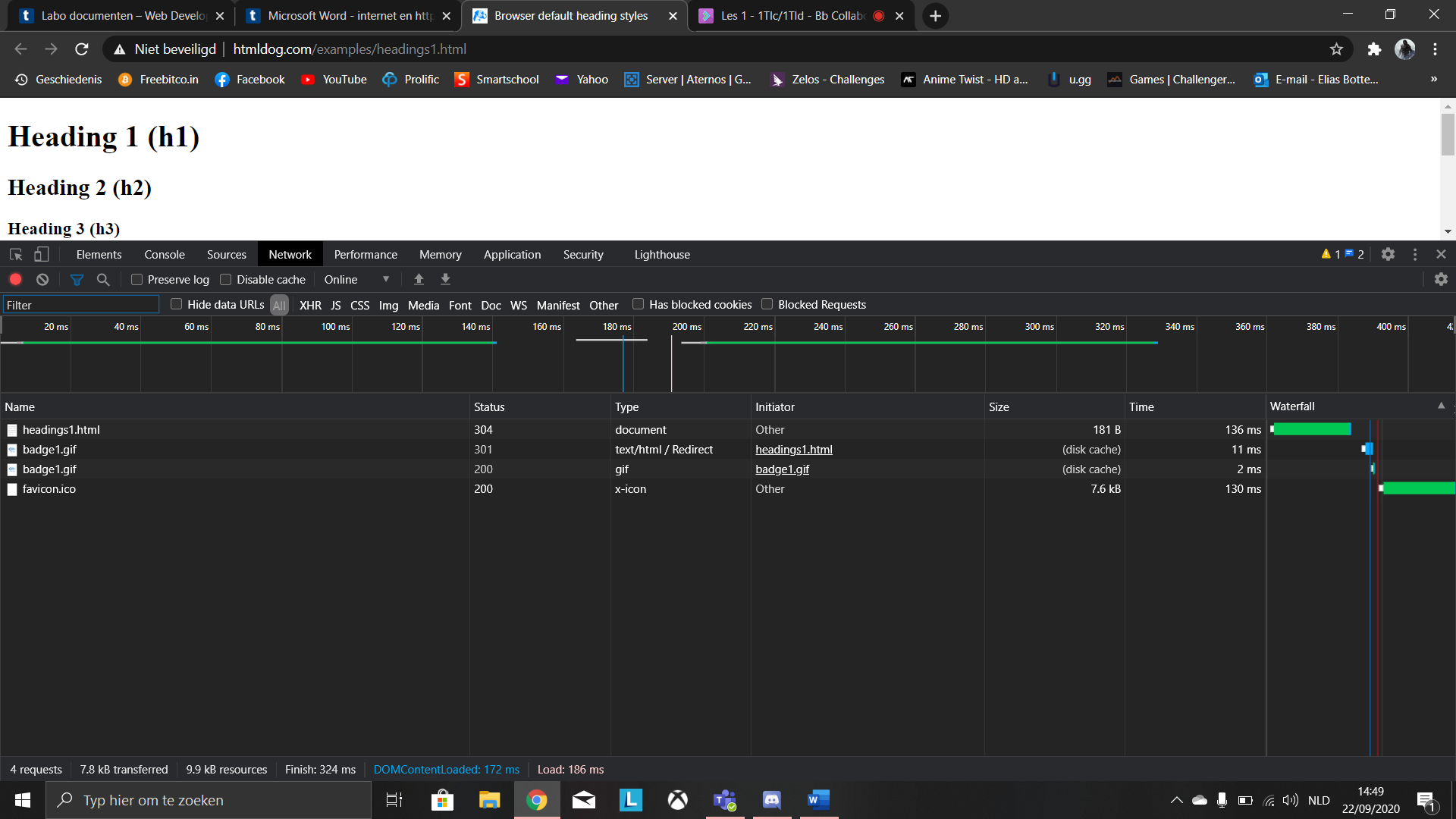
Afbeelding met tekst, schermafbeelding, binnen, elektronica

Automatisch gegenereerde beschrijving

Network code



Opdracht 4



**Welke resources heeft je browser nog meer opgevraagd? Hoe zie je dit?**

Gifs and icons

**Kun je in het HTML document in de response body terugvinden waarom net die resources werden opgevraagd?**

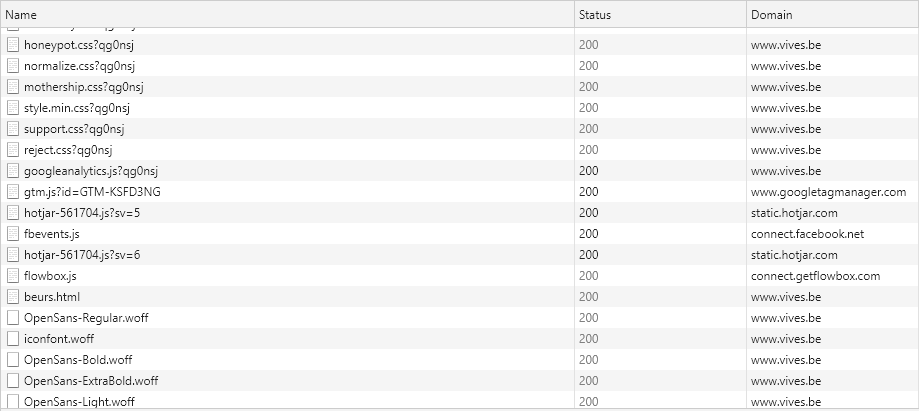
Omdat in de html een href is naar een img, deze moet dus opgehaald worden.

Opdracht 5

**Welke andere soorten resources worden opgevraagd door het inladen van deze pagina?**

Documenten, stylesheets, jpeg’s, png’s, scripts, gif’s, font’s, fetch, vnd.microsoft.icon en xhr’s

**Werden alle requests naar dezelfde server verstuurd?** Om dit te zien kun je best een 'domain' kolom toevoegen aan de tabel. Rechtsklik hiervoor op de hoofding van de tabel en vink 'Domain' aan. Rechts naast elke request zie je de timing informatie die weergeeft wanneer de request verstuurd werd en wanneer de response werd ontvangen.

Neen, 

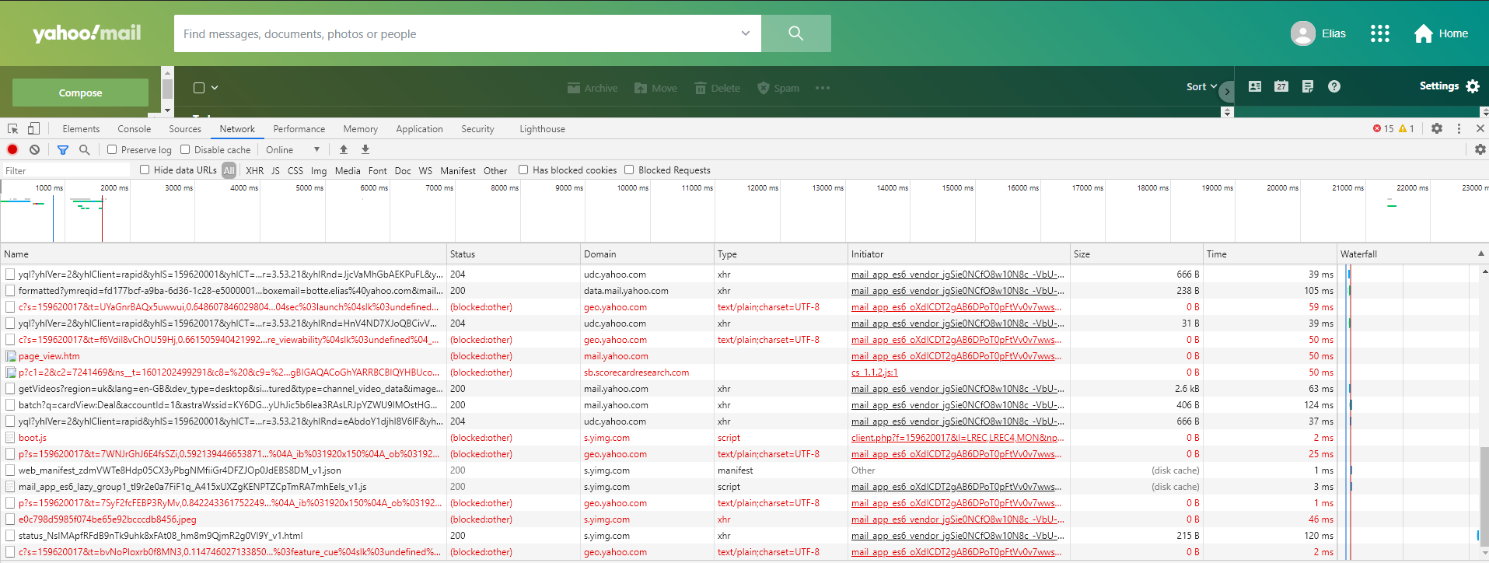
Het opvragen van 2 resources van een webpagina kan normaliter onafhankelijk van elkaar gebeuren, de browser kan dus een pagina sneller kunnen inladen door een volgende request te versturen nog voor de response op een vorige request werd ontvangen. **Hoe kun je dit uit de timing informatie afleiden?**

De requests gebeuren praktisch op hetzelfde moment, de responses niet. Sommige resources hebben meer tijd nodig om te downloaden.

**Lijkt het erop dat het aantal gelijktijdige 'onafgewerkte' requests beperkt is?** (onafgewerkt, in de zin dat de request verstuurd is maar nog geen response werd ontvangen).

Nee, de requests gebeuren bijna instant, maar volgens een prioriteitranking.

Opdracht 6

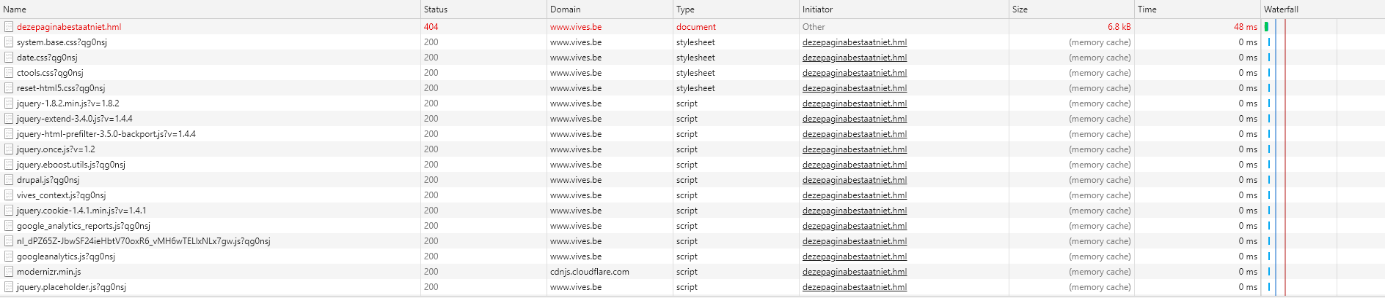


De requests dienen om de mailbox te refreshen en mogelijke nieuwe mail op te halen.

Opdracht 7

**En merk op dat er wel degelijk een response teruggestuurd wordt alhoewel de pagina niet bestaat. Hoe komt dit?**

Omdat het de foutpagina template moet ophalen.



**Wat betekent de status code 404 in de response header?**

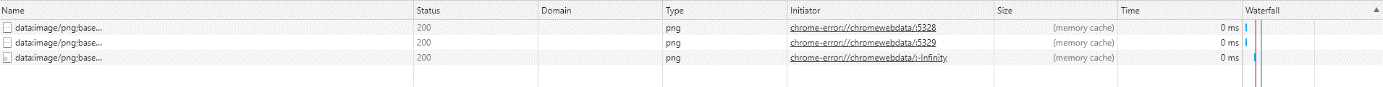
De browser kon communiceren met een bepaalde server, maar deze kon niet vinden wat wij zochten. Vandaar dat aangegeven wordt dat de pagina niet gevonden kon worden.



Opdracht 8

**Wat is het verschil met de vorige opdracht?**

We krijgen hier zelfs geen statuscode



Opdracht 9

|  |  |  |  |
| --- | --- | --- | --- |
| Succescodes | Omleidingscodes | Client errors | Server errors |
| 200=OK, succesvolle HTTP request. | 301=Moved Permanently, deze en alle toekomstige aanvragen moeten verwezen worden naar de gegeven URI | 400=Bad request, de server kan of wil niet de aanvraag volbrengen door een client error. (Bv te groot bestand) | 500= Internal server error, de standaard error code wanneer een onbekende fout is opgetreden en er is geen geschikte specifiekere code. |
| 204=No Content, De aanvraag was succesvol maar geeft geen content weer. | 302=Found, de client moet naar een andere URL surfen of kijken | 401=Unauthorized, er is een authenticatie verreist die nog niet is ingegeven of is mislukt. | 503=Service Unavailable, de server kan de aanvraag niet behandelen (overbelast of down for maintenance) |
|  | 303=See Other, de response op de request kan gevonden worden onder een andere URI met de GET methode. | 404=Not Found, De gevraagde source kon niet gevonden worden, maar is misschien beschikbaar in de toekomst. |  |

Opdracht 10

**Zoek op het internet welke HTTP request methods er bestaan en schrijf ze neer.**

GET, HEAD, POST, PUT, DELETE, CONNECT, OPTIONS, TRACE, PATCH

**Waarvoor dienen de vaak gebruikte GET en POST methods?**

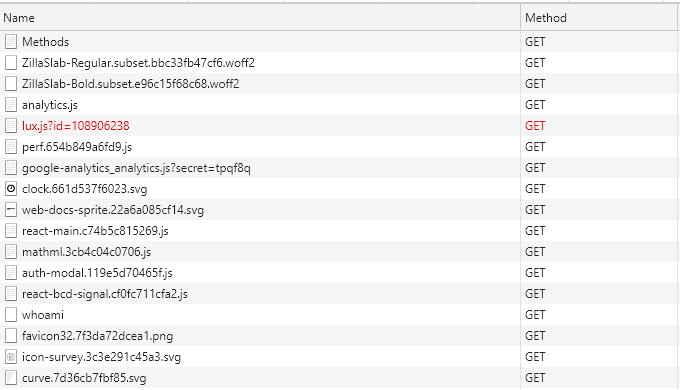
GET: vraagt een representatie van de specifieke resource aan, aanvragen met GET zouden enkel data moeten ontvangen.

POST: Deze methode is gebruikt om een entity aan te geven aan de gespecifieerde resource, dit veroorzaakt vaak een verandering aan de staat of neveneffecten aan de server.

**Waar in een request staat aangegeven om welke request method het gaat, en hoe vind je dit**

**terug in de Chrome developer tools (zie uitleg bij opdracht 3)?**

In de method tab.



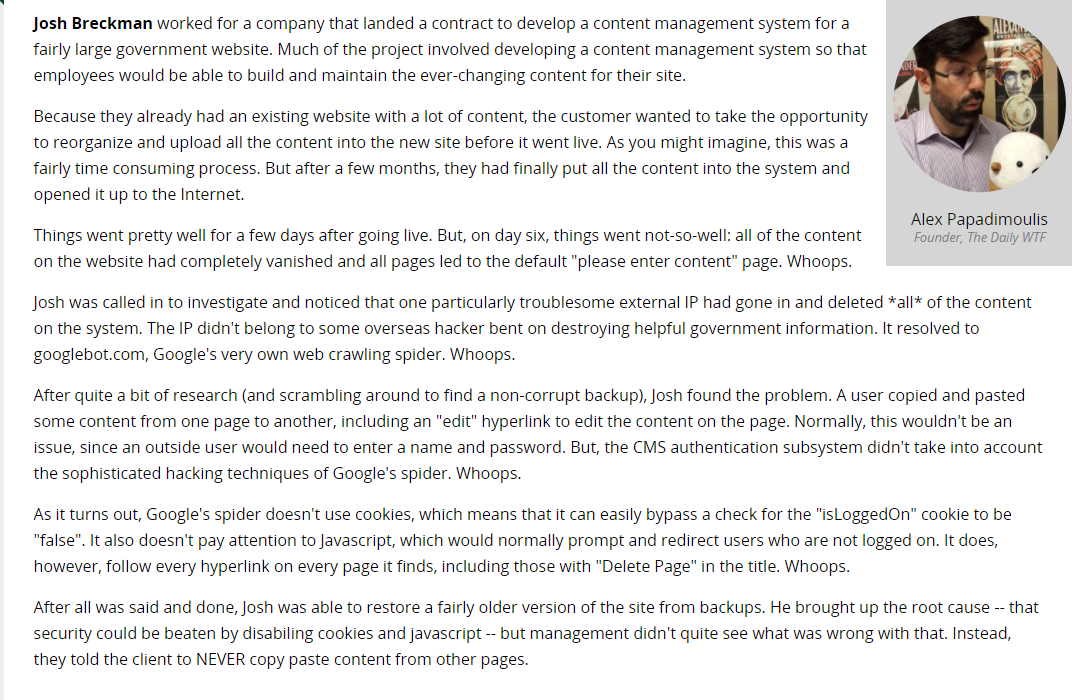
**Als je een url in de adresbalk van je browser typt en op enter drukt, wat voor request method gebruikt de browser dan om die resource op te vragen bij de server?**

GET

**Als je in een webpagina op een gewone hyperlink klikt, welke request method wordt er dan**

**gebruikt?**

GET



Stel, iemand schrijft een webapplicatie om producten te beheren en realiseert de"wis productgegevens" functionaliteit door middel van een gewone hyperlink met'wis' opschrift. Om een product te wissen moet een gebruiker dus gewoon op de 'wis' link klikken bij dit product.

Op een bepaald moment komt bv. de google‐bot langs die (programmatorisch) alle links uitprobeert om te zien wat dit oplevert aan nieuwe pagina's om te indexeren. Of stel dat de browser een accelerator plugin bevat die proactief gelinkte pagina's inlaadt zodat de gebruiker niet hoeft te wachten bij het klikken op een link.

**Wat zou er dan gebeuren met de productgegevens?**

Ze zouden allemaal verwijderd worden door de bot.

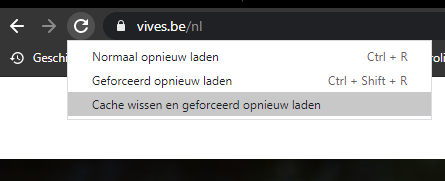
Opdracht 11

**Bekijk de vele requests die het inladen van die ene pagina heeft**

**veroorzaakt. Hoeveel request waren er in totaal?**

132+

Opdracht 12



**Hoeveel kilobytes of megabytes aan data werd er verstuurd om alle nodige resources in te laden?**

3.8mb

**Hoelang duurde het vooraleer alle resources van de pagina waren ingeladen?**

2,03s

**Klik nu gewoon op de refresh knop. Kijk nogmaals hoeveel data er werd verstuurd. Waarom is dit zoveel minder? Laadde de pagina sneller?**

63,9kb, we vragen niet alle resources op, waardoor de pagina sneller laad (788ms)

**Waar kun je zien welke documenten daadwerkelijk verstuurd werden en welke niet?**

Als ze een response body hebben, dan zijn ze verstuurd geweest.

**Waar vindt de browser dan de inhoud van de documenten die niet bij de server werden**

**opgevraagd?**

In de memory cache.

**Hoe weet de browser welke documenten best opgevraagd (moeten) worden en welke niet? M.a.w. hoe lang mag de browser een bepaalde resource als 'vers' te beschouwen? (Hint: kijk eens naar de response headers)**

De documenten die een bepaalde functie bevatten krijgen voorrang

Opdracht 13

 <https://www.nieuwsblad.be/> 8

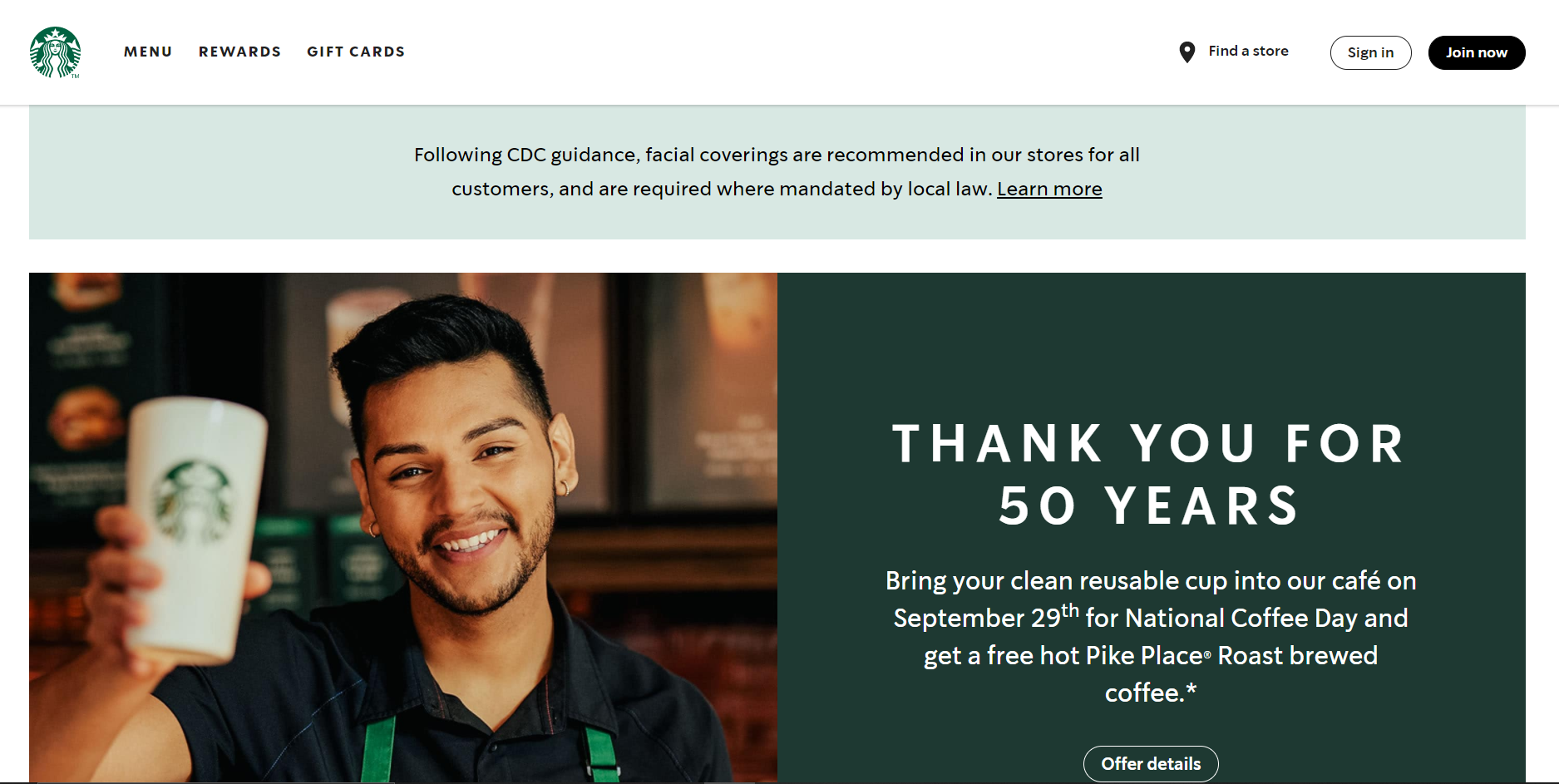
 <https://www.cnn.com> 11

 <https://www.vives.be> 7

 <https://www.vrt.be/vrtnws/nl/> 6

Advertising, essentieel, sociale media, website analytics, onbekend

Opdracht 14



Afbeelding met tekst, schermafbeelding, persoon

Automatisch gegenereerde beschrijving

**Hoeveel verschillende layouts tel je?**

2 verschillende layouts