

IJsselmeer Yachts

Max Harmsen - 2335778, Tom Huisman - 1484397,
Mika Kalshoven - 0000000, Matt ter Steege - 9932003

Universiteit Utrecht
Utrecht, Nederland

CONTENTS

Contents	1
1 abstract.....	1
2 Achtergrond onderzoek	1
3 Onderzoek.....	3
3.1 Toegankelijkheid	4
4 implementatie	4
4.1 Bezoekersreis.....	6
5 Reflectie.....	6
5.1 Communicatie	6
5.2 Versie management	7
5.3 Persoonlijke reflecties	7

1 abstract

Dit verslag beschrijft het project waar Max, Matt, Mika en Tom een website hebben gemaakt voor het bedrijf IJsselmeer Yachts. De website is gericht op het verhuren van luxe jachten waarmee de huurder niet alleen het IJsselmeer op kan, maar ook naar bestemmingen buiten Nederland (te vinden op de website). Het doel is om luxe uit te stralen op de website. Dit wordt gerealiseerd door het kleurgebruik. Daarnaast is het uitstralen van betrouwbaarheid ook belangrijk. Om dit voor elkaar te krijgen is er een overzichtelijke botenpagina gemaakt waar de gebruikers in één oogopslag de boten kunnen vergelijken en zo dus een eerlijke keuze kunnen maken. Voor de ontwikkeling van deze website zijn HTML, CSS, JavaScript en Bootstrap gebruikt. Deze vormgeving is herleid uit het onderzoeken van andere verhuursites, optimale kleurpaletten en verschillende lay-outs die samen leiden tot de beste gebruikerservaring. Om de website interactief te maken, is er gebruikgemaakt van 2 externe API's en een filtersysteem dat gebaseerd is op een JSON-tabel. De eerste API is van DummyJSON en geeft gesimuleerde reviews terug. Dit om te simuleren alsof er reviews ontvangen zouden worden van bijvoorbeeld een externe instantie als TripAdvisor. De tweede API is van Open-Meteo en geeft het weer terug gedurende een bepaalde periode. De gebruiker bepaalt aan de hand van het filtersysteem het gewenste tijdsbestek en daarna vragen we het weer van die dag op via de API-endpoint van Open-Meteo. Voor de locatie van deze weersvoorspellingen hebben we het midden van het IJsselmeer gekozen. Na ± 10 weken hebben we als resultaat een werkende website die gebruikmaakt van de 2 genoemde API-endpoints en een filtering op boten heeft door middel van een JSON-tabel. Onze website is terug te vinden op: <https://webpace.science.uu.nl/9932003/>.

2 Achtergrond onderzoek

Binnen ons groepje hebben we redelijk veel ervaring met HTML, CSS en JavaScript. De een wat meer dan de ander, maar we hebben allemaal op de middelbare school informatica gehad of we hebben in onze vrije tijd geëxperimenteerd. Dit geeft onze groep een enorm voordeel, omdat we de benodigde basis voor ons project hebben. We hoeven niet zozeer in te gaan op de werking en syntax, omdat we dit zo goed als uit ons hoofd kunnen doen. Als we toch tegen problemen aanliepen, hebben we de website "W3Schools" gebruikt om onze problemen op te lossen. Dit is een website die alles rondom het leren van webdesign aanbiedt (W3Schools, n.d.). W3Schools was vooral handig voor als we niet bij elkaar waren. Het merendeel van de tijd hebben we samen aan onze website gewerkt, waardoor meestal iemand anders het antwoord wist op bepaalde vragen.

Dit achtergrondonderzoek is meer gericht op het onderzoek achter de gemaakte keuzes voor de opmaak, lay-out en API-implementatie voor onze website, doordat wij als team de kennis voor het ontwikkelen van een website al bezitten, waardoor we ons meer richten op de keuzes eromheen.

Voor onze webapplicatie hadden we bedacht dat we een website wilden maken waarmee we een probleem voor mensen kunnen

oplossen. Al brainstormend tijdens de het eerste werkcollege kwamen we erachter dat we iets wilden doen met verhuur; we wisten alleen nog niet wat we wilden verhuren. Ons eerste plan was om een verhuursite voor huisjes te maken. Toen we onderzoek naar dit idee gingen doen, kwamen we erachter dat het lastig was om voor dit idee een goede API te vinden die we konden gebruiken. Daarnaast kwamen we ook al heel erg veel verhuursites tegen die huisjes verhuurden, dus het was ook niet een origineel idee. Vervolgens kwam het idee op om een verhuursite voor boten te maken, maar dan wel specifiek op het IJsselmeer. Nog steeds hadden we het probleem dat we het geen origineel idee vonden, want er zijn ook meer dan genoeg bootverhuursites, en daarom besloten we alleen luxe jachten te verhuren op het IJsselmeer. Dit bestaat namelijk nog zeker niet.

Nu is het belangrijk dat we goed onderzoek doen naar hoe we onze website willen weergeven, welke kleuren we willen gebruiken en welke API's we zouden kunnen gebruiken. Volgens het Nederlands Kleurinstituut hebben kleuren een diepgaande invloed op diverse aspecten, waaronder marketing en ontwerp. Daarom is het belangrijk om onderzoek te doen naar wat de website zou moeten uitstralen en welke kleuren daarvoor gebruikt moesten worden. In het onderzoek kwam al snel naar boven dat jachten geassocieerd worden met luxe. Daarom is het van belang dat er een royale en betrouwbare website gemaakt wordt, waar klanten een vertrouwelijk en luxueus gevoel bij krijgen. De kleuren paars en blauw stralen deze aspecten uit (nederlandskleurinstituut, 2024). Daarom is er gekozen om deze 2 kleuren als basis te gebruiken voor de website. Als er veel boten verhuurd moeten worden, moet de website beter zijn dan de website van de concurrent. Daarom is het noodzakelijk dat er onderzoek gedaan wordt naar hoe de website van de concurrentie eruitziet. Als groep is het van belang om de websites goed te evalueren. Door de evaluatie kunnen we bedenken wat we wel en niet moeten implementeren in onze website. We denken dat de concurrenten in twee groepen onderscheiden kunnen worden: verhuurbedrijven van boten buiten het IJsselmeer, en verhuurbedrijven op specifiek het IJsselmeer.

Een verhuurbedrijf dat buiten het IJsselmeer aanbiedt, is: YachtCharter Westerdijk (Westerdijk, n.d.). Dit bedrijf heeft een duidelijke website, met een overzichtelijke navigatie. Er is zichtbaar moeite gedaan om een goed ontwerp en een goede lay-out van de website. Dit maakt het overzichtelijk en biedt de mogelijkheid om makkelijk door de website te navigeren. Ook maakt de website gebruik van uitgebreide, maar vanzelfsprekende filtermogelijkheden, met visuele plaatjes van hun aanbod. Uit deze analyse volgde de conclusie dat de website van IJsselmeer Yachts deze aspecten ook zou moeten bevatten. Wel is er veel tekst te vinden op de website, wat naar onze mening te veel is. Een voorbeeld hiervan is de beschrijving van de boten, die meteen weergegeven wordt op de homepage. Wij vinden dit onnodig. Dit zal daarom ook niet op deze manier geïmplementeerd worden op de website van IJsselmeer Yachts, omdat dit naar onze mening het doel van de website weghaalt. Verder maakt de website het gemakkelijk om het bedrijf te contacteren. Dit wordt dus ook bij onze website geïmplementeerd.

Een ander verhuurbedrijf buiten het IJsselmeer is YachtCharter Sneek (Yachtcharter Sneek, 2025). Deze website is vergelijkbaar met de website van YachtCharter Westerdijk. De website oogt wederom goed, duidelijk ontworpen en goede lay-out. Wij vinden de manier waarop de recensies van de klanten weergegeven worden wel overzichtelijk. Dit zal dus op zo'n zelfde manier worden geïmplementeerd op onze site. Ook oogt deze website een stuk 'schoner'. Dit komt omdat er

geen grote stukken tekst te zien zijn op de homepage, in tegenstelling tot YachtCharter Westerdijk. Ook zien we dat dit bedrijf verschillende vaarroutes aanbiedt en dus kan dit ook voor ons een slimme implementatie zijn om bezoekers nog meer te kunnen bieden.

Een verhuurbedrijf specifiek gericht op onder andere het IJsselmeer is: Andijk Jacht Verhuur (Andijk jacht Verhuur, n.d.). Deze website is een wereld van verschil met de website van YachtCharter Westerdijk. De website oogt op het eerste gezicht veel minder professioneel. Dit komt door de onoverzichtelijkheid en het ontwerp en de lay-out van de website. We zien dat er weinig gebruik is gemaakt van CSS voor de opmaak van de website. Dit kunnen we beter doen, zodat we een concurrentievoordeel kunnen krijgen. We zien veel overeenkomsten tussen de websites. Zo hebben ze allemaal een navigatie-header en -footer, ze laten allemaal boten zien op de homepage en de recensies van oude klanten. Ook hebben alle websites mogelijkheden om te filteren op verschillende voorkeuren van de gebruiker. Denk dan aan de datum van het boeken, de hoeveelheid mensen waarmee ze willen varen en de lengte van de boot. Alle websites maken het de gebruiker ook heel makkelijk om contact op te nemen. Deze aspecten van de website zijn essentieel voor onze website en zullen dus ook geïmplementeerd worden.

Verschillend tussen de websites is de opmaak en lay-out. De ene is professioneler dan de andere. Ook zien we verschillen in de positie van verschillende secties op de websites. Denk dan bijvoorbeeld aan de informatie van de boten, klantenreviews. Filterpagina of de plek waar contact op genomen kan worden.

Voor onze website wilden we het groots aanpakken. Zo worden er niet alleen boten aangeboden, maar ook een platform bieden waar bezoekers inspiratie kunnen vinden voor hun bestemmingen, net zoals YachtCharter Sneek. Daarom zijn er ook bestemmingen toegevoegd aan deze website, om de bezoekers net dat beetje extra te geven.

De website moet gebruikmaken van API's, wat Application Programming Interface betekent. Dit zorgt ervoor dat applicaties gegevens met elkaar kunnen uitwisselen (Logius, n.d.). De API zorgt voor een communicatieverbinding tussen input en output (NCSC, n.d.). Voor deze website zijn 2 API-endpoints nodig. We gaan voor deze website gebruikmaken van een API die recensies kan leveren. Dit geeft klanten meer vertrouwen in het bedrijf, en zorgt voor een professionele look and feel. Ook wordt er een API gebruikt die het weer voor de aankomende dagen kan aanleveren. Hierdoor kunnen de bezoekers het weer checken voordat ze een boot huren. Zo wordt de kans op negatieve reviews door externe factoren beperkt.

Om klantrecensies te implementeren, is het wel van belang om een API-endpoint te vinden dat ook daadwerkelijk recensies teruggeeft. Een probleem hiervoor is wel dat, omdat we een nieuwe website hebben, we nog geen recensies hebben. We kiezen er daarom voor om een DummyJSON API te gebruiken. Dit is een zogenaamde "Mock" API die voor gegenereerde data kan ophalen (Ovi, 2025). Deze API haalt externe informatie op en levert deze data in een JSON-format. Mocht dit namelijk een echte website (van een verhuurbedrijf) zijn geweest, dan waren de reviews ook extern opgevraagd. Dit kan bijvoorbeeld uit een database of een platform als Trustpilot zijn. Het gebruik van deze API komt door de kennis van groepsgenoten in voorgaande webprojecten. Hierdoor hoefden we niet uitbundig onderzoek te doen naar hoe we deze API zouden kunnen integreren in onze website. De andere API die we wilden gebruiken is een weer-API. Hierdoor zullen de klanten het actuele weer zien voor de komende dagen wanneer ze een zoekopdracht

hebben ingevuld bij onze filtering. De API die deze service zal leveren is de API open-meteo. Deze API geeft de mogelijkheid om het weer op het IJsselmeer te tracken (door middel van coördinaten in het midden van het IJsselmeer), en weer te geven op onze website. Ook kan Open-Meteo actuele informatie geven over de wind, maximale en minimale temperatuur en of er neerslag komt (Open Meteo, n.d.). Om inspiratie op te doen is de website van Buienradar bekeken. Deze website geeft het actuele weer op een bepaalde plaats weer (Buienradar, n.d.). Buienradar geeft de belangrijkste informatie van het weer op een overzichtelijke manier weer, wat wij ook moeten implementeren op onze website. Gezien er veel positieve reviews waren over de app van Buienradar, vonden we dit een goeie plek om inspiratie op te doen.

Ook was er inspiratie nodig voor de website. Voor het ontwerp hebben we Dribbble gebruikt. Dit is een platform dat een grote collectie aan designideeën aanbiedt (Dribbble, n.d.). Met de keywords: “Boat”, “Boats”, “Yacht”, “Boat Rental” en “Yacht Rental”, hebben we kunnen sparren over hoe we willen dat onze website eruit komt te zien. We hebben gekozen voor een minimalistisch en overzichtelijk design. Op onze website moeten de boten duidelijk, zonder te veel informatie, worden weergegeven. De informatie wordt pas gegeven zodra de gebruiker op de boot klikt. Dit voorkomt een cognitieve overload.

3 Onderzoek

In dit deel wordt het ontwerp van de website van IJsselmeer Yachts beschreven. Voor de website worden HTML, CSS en JavaScript gebruikt. Deze technieken vormen de basis van de website. Met HTML, een afkorting van HyperText Markup Language, wordt de structuur van de webpagina's opgebouwd. Deze programmeertaal maakt het mogelijk om inhoud zoals tekst, hyperlinks en afbeeldingen in de website te plaatsen. CSS, een afkorting van Cascading Style Sheets, wordt ingezet voor de opmaak van de inhoud, waaronder kleuren, lettertypen en lay-outs. CSS bepaalt daarmee de visuele uitstraling van de webpagina. JavaScript wordt gebruikt om de website interactief en dynamisch te maken. Deze programmeertaal wordt toegepast bij de implementatie van de gekozen API's, DummyJSON en Open-Meteo. Hierover wordt later in dit document uitgebreider ingegaan.

De website van IJsselmeer Yachts is gericht op het verhuren van boten waarmee bezoekers kunnen varen op het IJsselmeer. Er is een breed aanbod aan boten beschikbaar voor verhuur. De focus ligt op het aanbieden van luxeproducten en het aantrekken van een bijbehorende klantendoelgroep. Om die reden is gekozen voor een luxe uitstraling van de website. Donkerblauw en paars zijn geselecteerd als hoofdkleuren, omdat onderzoek aantoonde dat deze kleuren een luxe en betrouwbare indruk wekken (nederlandskleurinstituut, 2024). Daarnaast worden wit, grijs en zwart gebruikt voor tekst, lichtblauw voor knoppen, wit voor de randen rondom afbeeldingen en roze voor het selecteren van afbeeldingen.

Voor het bouwen van de website wordt gebruikgemaakt van het webframework Bootstrap. Dit framework versnelt en vereenvoudigt het ontwikkelproces door het aanbieden van kant-en-klare componenten. Deze componenten zorgen bovendien voor een consistente uitstraling binnen de website.

De website is opgebouwd uit een header met navigatiebalk, een body en een footer. In de header bevinden zich hyperlinks naar de verschillende HTML-pagina's. Deze verwijzen onder andere naar het botenaanbod, de mogelijke vaarroutes en bestemmingen, de contactpagina en de pagina met achtergrondinformatie. Onderaan elke

HTML-pagina bevindt zich een footer waarin dezelfde hyperlinks terugkomen als in de navigatiebalk.

Bij het openen van de hoofdpagina wordt direct een afbeelding van een boot getoond. Daarnaast is er een filtersysteem aanwezig waarmee boten kunnen worden gefilterd op basis van voorkeuren. Hierdoor wordt de functie van de website meteen duidelijk. Ook wordt de tekst “Welkom tot het goede leven!” weergegeven, waarmee het gewenste gevoel van luxe en ontspanning wordt overgebracht. Verder naar beneden op de pagina zijn meerdere reviews zichtbaar van klanten die gebruik hebben gemaakt van de dienstverlening van IJsselmeer Yachts. Deze reviews vergroten het vertrouwen in zowel de service als de website (Hu et al., 2014). Voor het tonen van de reviews wordt gebruikgemaakt van de DummyJSON API, zoals besproken in het achtergrondonderzoek. De reviews bewegen dynamisch van rechts naar links, waardoor telkens nieuwe beoordelingen zichtbaar zijn.

Wanneer op “Onze boten” wordt geklikt of wanneer via het filtersysteem een gewenst formaat of capaciteit wordt geselecteerd, verschijnt een overzicht van alle boten die worden verhuurd. Op deze pagina kan verder worden gefilterd op lengte en het aantal zitplaatsen. De opgegeven wensen worden licht verruimd, zodat ook boten die dicht bij de gewenste criteria liggen worden getoond. Daarnaast wordt op deze pagina een weerbericht weergegeven met actuele weerdetails voor het IJsselmeer, inclusief een advies over het meest geschikte type boot voor die dag. Hiervoor moet wel een gewenste vertrekdatum zijn ingevuld. Bij verder scrollen verschijnen alle boten die binnen de ingestelde filters vallen. Per boot worden een afbeelding, het maximale aantal personen, de lengte, het bouwjaar en de huurprijs per week getoond. De boten worden weergegeven in een grid, wat zorgt voor een overzichtelijke en strakke uitstraling. Door op een boot te klikken, worden de detailgegevens van deze boot geopend. Bovenaan deze detailpagina wordt, als een hurendatum is ingevoerd, ook een weerbericht voor die specifieke dag getoond. Deze gegevens worden opgehaald via de Open-Meteo API. De API levert informatie zoals minimale en maximale temperatuur, verwachte regenval tussen 09:00 en 21:00 en windkracht. Op basis hiervan wordt een praktisch advies gegeven, bijvoorbeeld over voldoende water drinken en zonnebescherming bij warm weer of het dragen van warme kleding bij lage temperaturen. Deze informatie wordt gepresenteerd in een duidelijke en verzorgde tekst.

Na het selecteren van een specifieke boot wordt via een hyperlink doorgestuurd naar een vervolgpagina. Op deze pagina worden opnieuw de specificaties van de boot getoond, aangevuld met een grotere afbeelding. Dit biedt de mogelijkheid om de keuze nogmaals te controleren. Daarnaast is er een rekentool beschikbaar waarmee een indicatie kan worden berekend van de totale huurprijs op basis van het aantal dagen en het aantal gasten. Onder de specificaties en reserveringsmogelijkheden staat een uitgebreide beschrijving van de boot.

Op de pagina “Bestemmingen” wordt een breder aanbod aan vaarroutes en bestemmingen gepresenteerd dan alleen het IJsselmeer. Aangezien de aangeboden boten vaak groot zijn, bestaat er bij klanten regelmatig de wens om meer te ontdekken dan uitsluitend het IJsselmeer. Daarom wordt ook de mogelijkheid geboden om naar andere bestemmingen te varen.

Op de “Contact”-pagina is het mogelijk om contact op te nemen met IJsselmeer Yachts. Deze pagina bevat een formulier waarin een naam, e-mailadres en bericht kunnen worden ingevuld. Het formulier is bewust eenvoudig gehouden om de drempel voor contact zo laag mogelijk te maken.

De “About”-pagina biedt achtergrondinformatie over de website en het bedrijf IJsselmeer Yachts. Hier wordt onder andere het verhaal over het ontstaan en verleden van het bedrijf verteld. Ook worden de oprichters voorgesteld, zodat duidelijk is wie achter de website en de organisatie staan. Dit draagt bij aan het opbouwen van vertrouwen.

Voor het tonen van reviews wordt gebruikgemaakt van de DummyJSON API. Deze API maakt het mogelijk om fictieve data op te halen zolang echte klantgegevens nog niet beschikbaar zijn (Ovi, 2025). De ontvangen data worden verwerkt om deze visueel aantrekkelijk en duidelijk te presenteren. Zo wordt een numerieke waardering die via de API binnenkomt, met behulp van JavaScript omgezet naar ster-iconen, zodat de beoordeling direct en intuïtief te begrijpen is voor bezoekers.

3.1 Toegankelijkheid

De toegankelijkheid van de website is beoordeeld en verbeterd op basis van zowel automatische als handmatige controles. Als uitgangspunt is het “Lighthouse” toegankelijkheids-rapport gebruikt. Dit rapport behaalt een score van 0.97 (op de schaal 0-1), wat aangeeft dat de website op de meeste geautomatiseerde toegankelijkheidscontroles goed scoort. Hierbij zijn onder andere controles uitgevoerd op correcte ARIA-attributen, geldige rollen, taalinstellingen (html lang), documenttitels, labels bij formulieren, kleurcontrast en semantische structuur zoals lijsten en koppen.

Omdat “Lighthouse” slechts een deel van de toegankelijkheidsproblemen automatisch kan detecteren, is aanvullend handmatig getest. De focus lag daarbij op toetsenbordtoegankelijkheid en logische navigatie. Tijdens deze tests bleek dat niet alle interactieve elementen standaard bereikbaar waren zonder muis. Dit is gezien als een concreet toegankelijkheidsprobleem.

Om dit op te lossen zijn zo veel mogelijk knoppen, links en invoervelden expliciet tab-indexed gemaakt. Alle formulieren, navigatie-elementen en interactieve componenten zijn gecontroleerd op bereikbaarheid via de Tabtoets. Daarbij is niet alleen gekeken of een element focus kon krijgen, maar ook of de tab-volgorde logisch en voorspelbaar was. Onjuiste focusvolgordes en elementen die focus kregen zonder functionele waarde zijn aangepast of verwijderd.

Daarnaast is gecontroleerd of de visuele volgorde overeenkomt met de DOM-structuur en of focus duidelijk zichtbaar blijft tijdens navigatie. Dit sluit aan bij Lighthouse-checks zoals logical tab order, focusable controls en visual order follows DOM, die geen fouten rapporteerden maar wel handmatige bevestiging vereisen.

De combinatie van een hoge Lighthouse-score en gerichte handmatige verbeteringen zorgt ervoor dat de website beter bruikbaar is voor gebruikers die moeite hebben met een muis of volledig afhankelijk zijn van toetsenbordnavigatie. Accessibility is hiermee geen bijzaak gebleven, maar aantoonbaar meegenomen in de technische uitwerking van de website.

Naast toetsenbordtoegankelijkheid is ook het kleurgebruik van de website beoordeeld (Ng, 2017). Het Lighthouse toegankelijkheidsrapport controleert expliciet op kleurcontrast en kent hier een significante weging aan toe. De hoge totaalscore van 0,97 geeft aan dat tekst en interactieve elementen in de meeste gevallen voldoende contrast hebben ten opzichte van hun achtergrond. Dit is belangrijk voor gebruikers met verminderd zicht, kleurenblindheid of bij gebruik van schermen in ongunstige lichtomstandigheden.

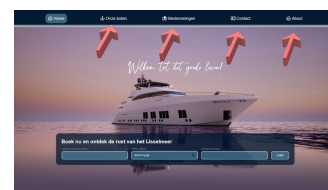
Op basis van deze controle is vastgesteld dat tekst, knoppen en links goed leesbaar blijven zonder afhankelijk te zijn van subtiele kleurverschillen. Waar kleur wordt gebruikt om betekenis over te brengen, zoals bij links of interactieve elementen, is dit ondersteund

door duidelijke visuele vormgeving en niet uitsluitend door kleur. Hierdoor blijft de interface begrijpelijk voor gebruikers die kleuren minder goed kunnen onderscheiden. Het kleurgebruik ondersteunt daarmee de leesbaarheid en bruikbaarheid van de website, zonder dat informatie verloren gaat wanneer kleurwaarneming beperkt is.

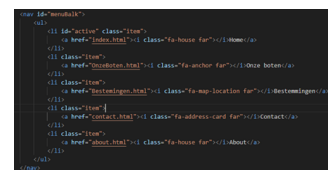
4 implementatie

Ondanks het feit dat er geen grootte, nieuwe design keuzes zijn gemaakt tussen het ontwerp en de implementatie, zijn er wel kleine verbeterpunten gevonden tijdens het realiseren van het ontwerp. Een punt waar dit goed te zien is, is bij de “bestemmingen” pagina. Om duidelijk het verschil te laten zien tussen het assortiment boten, en de beschikbare locaties, is er gekozen om meer gebruik te maken van blauwtinten.

De website wordt gestart op de index.html, en vanaf daar kan via de menubalk worden genavigeerd naar alle andere webpagina's



Figuur 1



Figuur 2

Dit is de HTML-code voor onze menubalk. De “back-end” werkt met <a> links, die verwijzen naar de andere html-bestanden. Ook wordt er gebruik gemaakt van twee klassen: active en item. Dit wordt gebruikt om de actieve webpagina te weergeven in de menubalk. Daarnaast hebben we gekozen om in de menubalk specifieke iconen te gebruiken voor specifieke pagina's. Dit is gedaan met de FA (Font Awesome).



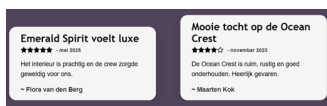
Figuur 3

Dit is hoe de menubalk er nu uit ziet, omdat het alleen HTML gebruikt. Met behulp van CSS is de decoratie verbeterd.

```
#menuBalk {
  width: 100%;
  background-color: #001a30;
  position: sticky;
  height: 100px;
  top: 0;
  z-index: 9999;
  display: flex;
  align-items: center;
  justify-content: space-around;
}
#menuBalk ul {
  width: 90%;
  display: flex;
  justify-content: space-around;
  font-size: 1.5rem;
}
#menuBalk ul li {
  float: left;
  list-style-type: none;
}
#menuBalk ul .item a {
  display: block;
  color: white;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}
#menuBalk ul .item i {
  margin-right: 6px;
}
ul .item a:hover {
  border-radius: 20px;
  background-color: #111111;
}
```

Figuur 4

Dit is de code die zorgt voor de UI van de menubalk. We benoemen alleen de belangrijke code hiervan. Bovenaan is te zien dat de menubalk een sticky position heeft, dit zorgt ervoor dat ook als de gebruiker naar beneden scrolt, de menubalk nog steeds bovenaan het scherm blijft. In de menubalk ul .item a, is te zien dat de tekst-decoratie op none staat. Dit zorgt ervoor dat er geen blauwe/paarse lijnen onder de link staat. De .item a:hover, zorgt dat de achtergrond verandert als de gebruiker met de muis over een pagina hoovert.



Figuur 5

Een ander punt wat te zien is op de hoofdpagina, is de review functie. Dit is gemaakt met een API, die willekeurige reviews genereert. Het API-endpoint dat we hiervoor gebruiken is: <https://dummyjson.com/c/40df7598-493f-b177>

```
const function loadReviews() {
  const res = await fetch("https://dummyjson.com/c/40df7598-493f-b177");
  const products = await res.json();

  const container = document.getElementById("reviews");

  const set = products
    .map(p => {
      <div class="review">
        <div class="stars">
          <CountStars/>
          <span class="line"> <p.date/> </span>
        </div>
        <p><p.description/>
        <p class="user"> <p.user/>
        </div>
      </div>
    })
    .join("");

  container.innerHTML = set + set;
}
```

Figuur 6

Dit is de JavaScript code die hiervoor zorgt. Allereerst wordt via een fetch de data van het API-endpoint gehaald. De code daaronder

gebruikt de verzamelde data en weergeeft het op de webpagina. Het API-endpoint levert alleen een score van 0 – 5, maar de visualisatie moeten wij zelf maken.

```
function countStars(stars){
  let star = parseInt(stars);
  if (star == 3) star = 5;
  let emptystar = 5 - star;
  let html = "";

  for (let i = 0; i < star; i++) {
    html += <i class="fa-solid fa-star"></i>;
  }

  for (let i = 0; i < emptystar; i++) {
    html += <i class="fa-regular fa-star"></i>;
  }

  return html
}
```

Figuur 7

Dit is de code die de score van 0-5 in een aantal sterren omzet. Naar HTML-code die wordt geprojecteerd op de pagina.

```
const fetchBoats = async () => {
  const res = await fetch("https://dummyjson.com/boats");
  const boats = await res.json();

  const container = document.getElementById("boats");

  const set = boats
    .map(b => {
      <div class="boat">
        <div class="header">
          <div class="name"> <b><b.name/></b></div>
          <div class="image"> <img alt="boat image" />
        </div>
        <div class="details">
          <div class="builder"> <b><b.builder/></b></div>
          <div class="length"> <b><b.length_meters/></b></div>
          <div class="years"> <b><b.years/></b></div>
          <div class="capacity"> <b><b.capacity/></b></div>
          <div class="price"> <b><b.price_per_week/></b></div>
        </div>
      </div>
    })
    .join("");

  container.innerHTML = set + set;
}
```

Figuur 8

Dit is het meest ingewikkelde stuk code van de website: het algoritme achter de zoekbalk. Dit werkt als volgt:

Allereerst wordt via een fetch het JSON bestand boats.json opgevraagd. Dit is het bestand wat wordt gebruikt om de informatie van de bestemmingen en de boten op te slaan. Dit bevat in ieder geval een id-nummer, naam, plaatje, en nog specifieke details voor ofwel de bestemming of de boot. Hieronder is een van de boten te zien in het JSON-bestand.

```
{
  "id": 1,
  "name": "Blue Horizon",
  "image": "images/OnzeBoten/Blue Horizon.jpg",
  "builder": "Dutch Shipbuilders BV",
  "length_meters": 60,
  "years": "2008/2018",
  "capacity": 12,
  "price_per_week": 2450
}
```

Figuur 9

De verzamelde informatie wordt onder handen genomen, en gefilterd op een aantal details. Elke boot wordt doorgenomen, en bij elke boot wordt er vervolgens bekeken of er wel iets ingevuld in de tekstboxen van de lengte van de boot, en het aantal mensen wat erop moeten kunnen. Daarnaast wordt er bekeken of de lengte van de boot, binnen een marge van +- 10 meter valt. Ook het aantal passagiers wat op de boot moet kunnen, moet binnen een marge van +- 2 mensen vallen. Als een van deze details niet binnen de filter past, wordt de boot "weggegooid" deze wordt niet bij de zoekresultaten laten zien, omdat de details van deze boot te ver van de verzochte specificaties lagen.

Als een boot wel aan al deze eisen voldoen wordt er een HTML-code gegenereerd.

```
const fetchBoats = async () => {
  const res = await fetch("https://dummyjson.com/boats");
  const boats = await res.json();

  const container = document.getElementById("boats");

  const set = boats
    .filter(b => {
      b.length_meters >= 50 & b.length_meters <= 70 & b.capacity >= 10 & b.capacity <= 15 & b.price_per_week >= 2000 & b.price_per_week <= 3000
    })
    .map(b => {
      <div class="boat">
        <div class="header">
          <div class="name"> <b><b.name/></b></div>
          <div class="image"> <img alt="boat image" />
        </div>
        <div class="details">
          <div class="builder"> <b><b.builder/></b></div>
          <div class="length"> <b><b.length_meters/></b></div>
          <div class="years"> <b><b.years/></b></div>
          <div class="capacity"> <b><b.capacity/></b></div>
          <div class="price"> <b><b.price_per_week/></b></div>
        </div>
      </div>
    })
    .join("");

  container.innerHTML = set + set;
}
```

Figuur 10

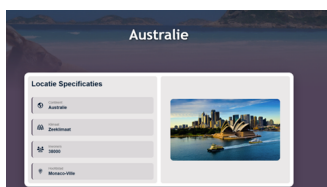
Dit is hoe dat wordt gedaan. Door middel van een InnerHTML kan de HTML-code rechtstreeks in de JS kunnen worden gezet (tussen aanhalingstekens uiteraard). Om te zorgen dat deze code werkt voor alle boten, worden een aantal details gehaald uit de JSON van de geselecteerde boot. Boat.image, boat.builder en nog een aantal details kunnen zo worden geselecteerd uit de JSON. Helemaal aan het einde van de zoekalgoritme functie, is ook nog een regel die eventuele fouten opvangt.

```
..catch(err -> console.error("Error loading boats:", err));
```

Figuur 11

Mocht er iets misgaan, zorgt deze zin ervoor dat niet de hele website vastloopt, maar dat de site zo ver mogelijk doorgaat zoals het hoort.

Een ander ingewikkeld deel van de website is het linken tussen de boten/bestemmingen, en de extra informatie die wordt weergegeven als erop wordt geklikt.



Figuur 12

Dit wordt op eenzelfde manier gedaan als het laden van de boten. Over alle boten/bestemmingen zit een link verstopt die, als er op wordt gedrukt, wordt gelinkt naar een andere HTML-pagina, en daarbij wordt ook meteen een nummer van de boot/bestemming meegegeven. Op deze pagina wordt weer de JSON-bestand door middel van een fetch opgevraagd. Vervolgens wordt het id-nummer die door de link werd meegegeven gebruikt om de juiste boot/bestemming onder handen te nemen.

Hierna wordt er weer via een InnerHTML een hele HTML-code gegeven voor de layout van de pagina.

Wat betreft de implementatie van de front-end kant van het ontwerp, volgens het ontwerp zouden onze hoofdkleuren paars en blauw worden, om een luxe uitstraling te geven die past bij het verhuren van jachten. In de praktijk is dit ook gedaan, maar wel met een paar andere kleuren erbij. Dit is uiteindelijk de kleur palette geworden.



Figuur 13

Het gebruik van bootstrap bleek een ingewikkeld punt te zijn voor dit team. Omdat iedereen al ervaring had met CSS, HTML en

JavaScript, lag de voorkeur erg sterk bij het zelf opbouwen van componenten van de site. Dat gezegd hebbende is er wel gebruik gemaakt van het framework, voornamelijk bij de "contact" webpagina.

4.1 Bezoekersreis

De bezoekersreis binnen de website van IJsselmeer Yachts is bewust logisch en stapsgewijs opgebouwd, met als doel bezoekers op een natuurlijke manier te begeleiden richting het huren van een boot. De reis begint op de hoofdpagina, waar direct duidelijk wordt wat de website aanbiedt. Door het tonen van een grote, sfeerbepalende afbeelding van een boot in combinatie met een zichtbaar filtersysteem wordt meteen gecommuniceerd dat de kern van de website draait om het huren van luxe boten.

Na binnenkomst op de homepage volgt de oriëntatiefase. In deze fase krijgt de bezoeker de mogelijkheid om via filters alvast een eerste selectie te maken op basis van wensen zoals capaciteit en formaat van de boot. Tegelijkertijd zorgen de getoonde reviews voor sociale bevestiging. Door ervaringen van eerdere klanten zichtbaar te maken, ontstaat sneller vertrouwen in de aangeboden service. De dynamische weergave van deze reviews trekt de aandacht zonder de bezoeker af te leiden van het hoofddoel.

Wanneer een bezoeker besluit verder te gaan, leidt de navigatie of het filtersysteem naar de pagina "Onze boten". Hier verschuift de focus van oriëntatie naar vergelijking. Het overzicht van beschikbare boten, weergegeven in een strak grid, maakt het eenvoudig om verschillende opties naast elkaar te bekijken. Filters blijven beschikbaar, zodat de selectie verder verfijnd kan worden. Tegelijkertijd wordt aanvullende context geboden in de vorm van een weerbericht en praktisch vaaradvies, wat bijdraagt aan een realistisch en betrouwbaar beeld van de huurervaring.

Na het selecteren van een specifieke boot komt de bezoeker terecht op de detailpagina. In deze fase staat bevestiging centraal. Door grotere afbeeldingen, uitgebreide specificaties en een diepgaandere beschrijving kan gecontroleerd worden of de gekozen boot daadwerkelijk aansluit bij de verwachtingen. De prijsindicatie op basis van huurperiode en aantal gasten helpt om mogelijke twijfel weg te nemen en voorkomt verrassingen. Ook het opnieuw tonen van weersinformatie voor de gekozen datum ondersteunt het gevoel van voorbereiding en controle.

Tot slot biedt de website aanvullende pagina's, zoals bestemmingen, contact en about, die niet direct gericht zijn op conversie maar wel een belangrijke ondersteunende rol spelen. Deze pagina's versterken het vertrouwen in het bedrijf, geven extra informatie en verlagen de drempel om contact op te nemen. Op deze manier vormt de bezoekersreis een samenhangend geheel waarin oriëntatie, vergelijking, bevestiging en vertrouwen elkaar logisch opvolgen.

5 Reflectie

Onze groep bestond uit 2 personen die al wat meer ervaring hadden met webontwikkeling en 2 personen die wat minder, maar nog steeds ervaring, hiermee hadden. Deze verdeling bleek achteraf heel erg prettig te zijn, omdat op deze manier niemand lang met problemen hoefde te blijven zitten. Er was altijd wel iemand in de groep die een oplossing wist, waardoor de ontwikkeling van de website heel soepel verliep.

5.1 Communicatie

De communicatie binnen onze groep was een van de sterkste punten. In plaats van meteen te beginnen, werd er eerst besproken wat de algemene designkeuzes waren voor de website. Door dit goed van

tevorens te bespreken, was het eigenlijk niet nodig om later hele stukken van de website opnieuw te maken, omdat het niet paste bij de overall look and feel van de website.

Het brainstormen over het uiterlijk en de functionaliteiten van de website deden we vooral tijdens de werkcolleges. We deden dit omdat we tijdens de werkcolleges elkaar makkelijker dingen kunnen laten zien die we gevonden hadden, en ook kun je elkaar makkelijker concepten uitleggen door het bijvoorbeeld even snel te schetsen.

Al snel bleek dat we niet met z'n allen dezelfde mening hadden wat betreft het uiterlijk. Zo vond de een dat de website "clean" moest blijven, wat vooral wit betekende met een paar accentkleuren, en de ander vond dat we juist veel kleur moesten toepassen op onze site. De oplossing die we hiervoor vonden, was om bronnen te zoeken die beide keuzes zouden moeten ondersteunen. Na de bronnen met de hele groep vergeleken te hebben, kwamen we erop uit dat de keuze voor kleur toch de betere keuze was. Juist doordat we ook bronnen hebben gebruikt, en niet gewoon 1 kant hebben gekozen, was iedereen in de groep oké met de keuze. Nu konden we namelijk als groep beargumenteren waarom 1 optie beter was dan de andere. Daarbij kwam ook nog het feit dat we niet 1 kant gekozen hebben, wat frictie binnen de groep voorkwam.

Gezien er thuis vooral ontwikkeld werd, is er ook een WhatsApp-groepsapp gemaakt. Hier konden vragen gesteld worden en werden nog twijfels over het uiterlijk van de website besproken. De reden voor thuis ontwikkeling was simpel, als je alleen werkt dan werk je efficiënter. Gezien de belangrijkste dingen al overlegd waren, was efficiëntie voor ons dus belangrijk tijdens de ontwikkeling.

Wat betreft de communicatie waren er ook wat verbeterpunten. Zo is elke pagina ontwikkeld door 1 persoon. Deze persoon had dan ook de vrije keuze om alles te doen op die pagina wat hij wilde, zolang het maar aan de designkeuze voldaan werd. Hierdoor werd er dus alleen bij de merge (later meer hierover) gekeken naar de pagina inhoudelijk. Gezien de pagina tijdens de merge al klaar is, kwam het erop neer dat er eigenlijk niet zo veel veranderd meer veranderd werd aan de pagina. Hierdoor hadden we als groep minder invloed op de inhoud van de pagina's. Dit zou voor een volgende keer beter kunnen. Dat betekent dat we de volgende keer niet alleen de designkeuzes met ons allen goed moeten bespreken, maar ook dieper op de inhoud van elke pagina in moeten gaan.

Gelukkig bracht het zojuist genoemde verbeterpunt niet veel problemen met zich mee. Meestal kon de hele groep zich god vinden in wat de ontwikkelaar op zijn pagina gemaakt had, wat ervoor zorgde dat ook hier geen frictie ontstond.

5.2 Versie management

Gezien de gehele groep bestond uit dubbele bachelors (Informatica / Informatiekunde) had iedereen al ervaring met GitHub. Dit kwam goed uit, want voor dit project besloten we ook snel dat het gebruik van GitHub belangrijk was.

Aan het begin van het project pushten we al onze commits op de main branch. Dit werd gedaan omdat er een pipeline aangemaakt was op de main branch, die alle wijzigingen meteen zou doorsturen naar de webspace van Matt ter Steege. Al snel ontdekten we dat dit toch niet handig was. Er ontstonden onnodig complexe merge conflicts en iedereen werkte met de verkeerde versie, omdat er niet constant gepulled werd. Daarnaast verloren we ook voortgang, gezien de merges sommige aanpassingen weer volledig ongedaan maakte.

Vanwege dit probleem werd er besloten om met branches te werken. Voor elk onderdeel werd er een nieuwe branch aangemaakt en pas als de ontwikkeling van dat onderdeel helemaal klaar was, werd

de branch overgezet naar main. Het voordeel hiervan was dat de merge conflicten een stuk minder complex waren. Het nadeel was dat je de veranderingen niet direct kan zien op de webspace. Dit nadeel was acceptabel, omdat dit project vooral bestond uit HTML- en CSS-files. Deze bestanden kun je direct openen in de browser (lokaal) en daarom kun je op deze manier je veranderingen direct zien.

Een afspraak die we maakten als groep was dat we onze veranderingen zo snel mogelijk zouden pushen. Aan het begin werd dit niet gedaan, waardoor de groep geen goed beeld had van de dingen die al klaar waren, en de onderdelen waar nog tijd besteed aan besteed moest worden.

Voor een volgend project is het belangrijk dat we meteen beginnen met het gebruiken van branches. Hierdoor verliezen we geen voortgang en ontstaan er minder frustraties aan het begin.

5.3 Persoonlijke reflecties

Max Harmsen

Al voordat ik met dit project begon, had ik al best wat ervaring met HTML en CSS. Hierdoor hadden deze 2 talen weinig verrassingen voor mij. Toch wou ik graag wat nieuws leren en besloot daarom wat meer gebruik te maken van JavaScript. Deze taal had ik namelijk niet tot nauwelijks gebruikt in mijn andere projecten, dus hier wist ik nog niet heel erg veel van. Ik kwam erachter dat JavaScript wat betreft structuur redelijk overeenkomt met C (een taal die ik in het vorige blok geleerd heb), maar dat de syntax net anders is. Hierdoor kreeg ik het gevoel dat ik niet weer helemaal vanaf 0 moest beginnen, wat me motiveerde om niet alleen het hoogstnodige JavaScript te gebruiken, maar ook JavaScript te gebruiken om de site visueel aantrekkelijker te maken. Een voorbeeld hiervan is het omzetten van de het getal dat de sterren aangeeft (vanuit de API-endpoint) naar daadwerkelijke ster-icons die de gebruiker kan zien. Daarnaast heb ik ook geleerd hoe ik door middel van JavaScript een API-endpoint kan bevragen en de informatie die het endpoint teruggeeft. Ik had namelijk nog nooit gewerkt met het principe van async-functies. Dit JavaScript-principe gaat ervan uit dat er op sommige regels code gewacht moet worden voordat de rest uitgevoerd kan worden. In het geval van het bevragen van een API-endpoint is dit uiteraard het geval. Zodra je het endpoint bevraagt, dan heb je niet meteen de informatie van je endpoint ontvangen; hier gaat tijd overheen. Gezien die informatie vanuit het endpoint verwerkt wordt in die functie, moet je dus wachten tot het endpoint zijn informatie heeft teruggegeven en daarvoor moet je async gebruiken.

Tom Huisman

Toen ik hoorde dat we voor dit project een website mochten gaan maken, was ik heel blij. Dit vind ik namelijk heel leuk om te doen. Ik had al wel enige ervaring met HTML, CSS en JavaScript, maar in vergelijking met mijn groepsleden was dit nog weinig. Het voordeel van groepsleden met veel ervaring is: dat als ik tegen problemen aanliep, kon ik dat meteen vragen en had ik eigenlijk binnen 10 seconden wel een oplossing. Dit was soms heel fijn, maar kon ook enorm frustrerend werken. Dit komt omdat ik over sommige dingen wel 10 minuten deed, terwijl ik wist dat het binnen een korte tijd gefixt kon zijn, alleen zou ik er zelf minder van leren. Daardoor koos ik er soms voor om het zelf uit te zoeken, want daar heb ik uiteindelijk het meeste aan. Wel heb ik nog veel kunnen leren van mijn groepsleden, omdat hun ervaring met het bouwen van websites veel handige tips en tricks met zich meebracht. Zo leerden

ze me dat ik de webpagina lokaal op Google kon openen, element kon inspecteren, waardoor mijn wijzigingen meteen zichtbaar werden. Dit was vooral handig met het finetunen van bepaalde posities. Ik heb mijn webpagina-skills echt kunnen verbeteren door dit project, waar ik enorm blij mee ben. Dit gaat mij later in mijn loopbaan weer verder helpen!

Matt ter Steege

Dit was niet de eerste keer dat ik een website bouw. Ik heb al aardig wat eerdere ervaring met het maken van verschillende websites die allemaal weer verschillende doelen hebben, dus mijn kennis over HTML, JS en CSS zijn al aardig goed. Echter was dit wel de eerste keer dat ik een website heb gemaakt waar Bootstrap gebruikt (moest) worden, dus dat was zeker een nieuwe ervaring en heeft me op dat gebied ook al wat meer kennis opgeleverd. Ook ben ik gewend normaal alleen te werken, dus dit was ook een goede oefening om te leren hoe een groepsproject het best gecoördineerd kan worden door goede afspraken en elkaar niet in de weg zitten door het gebruik van verschillende branches op Git. Ik heb hier uiteindelijk veel van geleerd. Meer op het gebied van hoe werken in een team werkt dan op het gebied van nieuwe kennis op programmeer niveau.

Mika Kalshoven

Ik vond dit project erg goed gaan. De meeste van ons groepje hadden al ervaring met het maken van websites. Dit hele groepje doet ook de dubbele bachelor met informatica, dus dit is niet de eerste keer dat we hebben geprogrammeerd. Dit zorgde dat het verloop van het ontwikkelen van de site erg veel soepeler ging. Waar ik zelf wel moeite mee had, was het schrijven van het verslag. Gelukkig hebben een aantal andere teamleden juist daar weer veel ervaring in. Dit zorgde dat iedereen elkaars zwakte aanvulde.

Wat voor mij wel helemaal nieuw was, is het gebruik van Bootstrap. Dit is iets wat ik zelf nog nooit heb gebruikt, en dit was heel interessant om voor deze opdracht wel te leren, en te gebruiken.

De communicatie voor dit project vond ik zelf zeer aangenaam. Als ik een appje naar iemand stuurde, of een vraag had, kreeg ik snel een hulpvolle reactie terug. Dit maakt het samenwerken heel erg aangenaam, omdat, ook al heb ik ervaring met het ontwikkelen van websites, ik wist zeker niet het meest. Het is daarom heel fijn als je “domme” vragen kan stellen aan je teamgenoten, en een fijne reactie terug krijgen.

Al met al, dit project was voor mij heel erg leerzaam. Ik heb heel veel nieuwe dingen over HTML, en vooral CSS geleerd. Daarnaast, heb ik het ook enorm naar mijn zin gehad tijdens dit project en heb ik mij op vele momenten helemaal stuk gelachen, wat natuurlijk ook belangrijk is tijdens een groepsproject.

References

- Andijk jacht Verhuur. (n.d.). *Zeilboot huren ijsselmeer & waddenzee – andijk zeiljacht verhuur* [Andijk jacht Verhuur]. <https://www.andijkjachtverhuur.nl>
- Buienradar. (n.d.). *Buienradar* [Buienradar]. <https://www.buienradar.nl/>
- Dribbble. (n.d.). *Dribbble - discover the world's top designers & creative professionals* [Dribbble]. <https://dribbble.com>
- Hu, N., Koh, N. S., & Reddy, S. K. (2014). Ratings lead you to the product, reviews help you clinch it? the mediating role of online review sentiments on product sales. *Decision Support Systems*, 42–53.
- Logius. (n.d.). *Logius | api-standaarden* [Logius]. <https://www.logius.nl/onze-dienstverlening/gegevensuitwisseling/api-standaarden>
- NCSC. (n.d.). *Ncsc – api security* [NCSC]. <https://www.ncsc.nl/webapplicaties/api-security>
- nederlandskleurinstituut. (2024, July). *Kleurvoorkeuren* [nederlandskleurinstituut]. <https://nederlandskleurinstituut.nl/blog/2024/07/16/kleurvoorkeuren/>
- Ng, C. (2017). A practical guide to improving web accessibility. *Journal of Library User Experience*, 7.
- Open Meteo. (n.d.). *Open meteo* [Open Meteo]. <https://open-meteo.com>
- Ovi, M. (2025, November). *Dummyjson* [GitHub]. <https://github.com/Ovi/DummyJSON>
- W3Schools. (n.d.). *W3schools* [W3Schools]. <https://www.w3schools.com>
- Westerdijk. (n.d.). *Yachtcharter westerdijk* [Westerdijk]. <https://westerdijk.com/motorboot-huren-friesland>
- Yachtcharter Sneek. (2025, March). *Boat rental sneek | yachtcharter sneek* [Yachtcharter Sneek]. <https://yachtchartersneek.nl>