

Webprogrammering Client-side

Akademiuddannelsen i Informationsteknologi



Kristoffer Surrow

Erhvervsakademi Aarhus

17-01-2018

Indhold

INDLEDNING	2
OVERVEJELSER	2
LØSNINGEN	2
HJEMMESIDEN	3
DYNAMISK INDHOLD	4
PERSPEKTIVERING	5
KONKLUSION	7
LITTERATURLISTE	8

INDLEDNING

Som del af et projekt hos smedevirksomheden *Netto Perforering*, laves en ny hjemmeside henvendt til virksomhedens kunder (primært byggefirmaer og arkitekter).

Hjemmesidens fokus er at danne et overblik over hvilke produkter virksomheden kan tilbyde/producere, samt at give kunderne mulighed for at søge i virksomhedens komplette lageroversigt, for her at sende forespørgsler på specifikke varer.

Virksomhedens primære omsætning består i salg af Perforerede Plader, der fås i et utal af forskellige varianter, hvilket gør det en kompliceret proces for kunder uden stor teknisk forståelse for Perforerede Plader, at finde de rette.

Formålet med lageroversigten er derfor at henvende sig til kunder med høj såvel som lav teknisk forståelse for produkterne, og give begge grupper mulighed for at finde præcis det produkt de søger på hjemmesiden. Dette mhp. at minimere mængden af telefonopkald, der allerede udgør en stor del af dagligdagen hos *Netto Perforering*.

OVERVEJELSER

Inden projektet kunne indledes var der flere faktorer at tage højde for. Bl.a. havde virksomheden specifikke krav til hjemmesiden:

- Laves i WordPress, så fremtidige opdateringer og ændringer kan foretages hurtigt og af dem selv.
- Lagerprogram findes i excel-format, og skal være let at indlæse til siden.
- Hjemmesiden skal være mobil-venlig, da mange af kunderne leder efter løsninger imens de står på byggepladsen.
- Lageroversigten skal henvende sig til kunder med høj såvel som lav teknisk forståelse for produkterne.
- Der skal være mulighed for at sende online-forespørgsler på specifikke varer.

At arbejde med et CMS som WordPress gør det let, overskueligt og ikke mindst hurtigt at opbygge en hjemmeside fra bunden, da man oplever stor støtte i opsætning af de forskellige elementer, og da det hele foregår i deres "live-editor".

På trods af de mange muligheder Content Management Systems tilbyder, sætter disse imidlertid også store begrænsninger, i form af rammerne disse systemer sætter op. Ønsker man som bruger at udvikle mere specifikke løsninger igennem WordPress foregår dette med PHP, hvilket åbner for et hav af muligheder, heriblandt at sende forespørgsler til en MySQL-server.

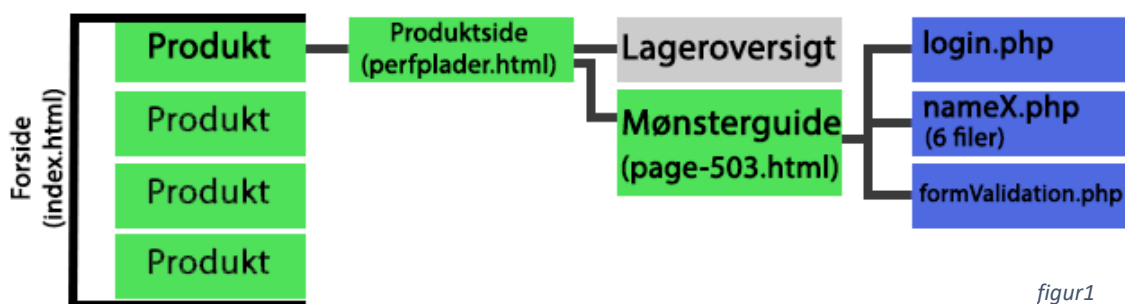
LØSNINGEN

Med dette projekt har jeg valgt at bruge en kombination af en WordPress-løsning på de simple sider, med brug af PHP på de mere komplicerede sider hvor jeg behøver at hente data fra MySQL-serveren, hvor lagerprogrammet i excel-format uploades.

Jeg har valgt at designet skal være simpelt, og siden let at navigere rundt på, med målet om hurtigt at nå frem til lageroversigten, og finde den vare der søges. Såfremt en kunde står på byggepladsen med sin

telefon, og ikke kan finde den vare de søger, har jeg valgt at fastgøre et telefonlink i toppen på mobil devices, således kunder i tvivl hurtigt kan ringe op til *Netto Perforering*.

For at henvende lageroversigten til dem med høj såvel som lav teknisk forståelse, har man to muligheder at søge Perforerede Plader på: "Lagerprogram" og "Mønsterguide". Lagerprogrammet for de tekniske, med en totaloversigt og filtersøgning til de individuelle kolonner og et link til at sende en forespørgsel, lavet i WordPress med et tabel-plugin. Mønsterguiden for de mindre tekniske, fører folk igennem de forskellige produkter ved at vælge: Leveringstid-Materiale-Perforeringstype-Hulstørrelse og Deling for til sidst at vælge den enkelte plade med en formular til at sende en forespørgsel.



figur1

Som det ses på *figur1* præsenterer forsiden de 4 mest populære produkter virksomheden producerer, hvoraf Perforerede Plader præsenteres først. Herfra kan man klikke ind på en *produktside* der præsenterer det valgte produkt, hvorfra man kan vælge *lageroversigten* eller *mønsterguiden*.

Mønsterguiden afhænger af flere PHP-sider (blå bokse i *figur1*), for at hente data fra serveren. Siden *Mønsterguide* var i første omgang også skrevet i PHP for at indgå på det samlede WordPress-site, men i det opgavemateriale jeg afleverer har jeg ændret denne til en HTML-fil så den kan åbnes offline. Fra mine andre sider linker jeg dog stadig til mønsterguiden på WordPress-sitet, så der kan hentes data fra serveren og I får den rette oplevelse af funktionen.

For at opfylde kravene til denne opgave har jeg derfor skrevet 3 sider fra WordPress-sitet i HTML (grønne bokse i *figur1*), som består af en forsider (index.html), produktsiden for perforerede plader (perfplader.html), samt mønsterguiden for perforerede plader, og de tilhørende 8 PHP-filer.

For at opnå den bedste oplevelse at mit projekt da: 1) Vælg "Se Mere" for perforerede plader 2) Prøv "luftprocentberegneren" 3) Vælg "Mønsterguide" 4) Vælg: "Alle varer", "Aisi 304", "RT", "8.00mm", "12.00mm", vælg en vilkårlig plade og derefter afprøv formularen.

HJEMMESIDEN

I hjemmesidens head-tag findes den overordnede information om siden, bl.a. hvilket charset der bruges, den overordnede titel på siden, og hvilke stylesheets der skal bruges. Da jeg på siden bruger flere stylesheets har jeg bevidst placeret de eksterne stylesheets over mit primære, for på den måde sikre mit primære stylesheet den største magt.(Duckett. J. 2011. s239)

Af eksterne stylesheets har jeg bl.a. *animate.css* for at animere flere elementer på siden, samt Bootstrap v3 og v4. V3 bruges udelukkende til *Glyphicons*, mens v4 bruges i flere omgange. Hvad jeg bruger Bootstrap v4

til beskrives senere i dette afsnit. (Hjemmeside:Bootstrap, Hjemmeside:Animations) Det kan diskuteres om tilføjelsen af endnu et Bootstrap library, udelukkende for *Glyphicons* kan forringe hastigheden hvormed siden loader, men da både HTML og CSS indlæses væsentligt hurtigere end fx JavaScript, der først skal igennem en interpreter, burde dette ikke give noget særligt "performance-hit". (Duckett. J. 2013 s.40, s.356)

Strukturen for min hjemmeside følger i store træk den nye HTML5-struktur, med flere nye layout elementer som erstatning for brug af div-elementet, som det ses på *figur2*. Ved brug af den nye HTML5-struktur kan sidens opdeling bedre beskrives, hvilket kan gøre det mere overskueligt at navigere rundt i koden, samt forbedre hvordan søgemaskiner vægter indholdet. (Duckett. J. 2011 s.432)

På min side har jeg til forskel fra layoutet på *figur2* opdelt indholdet i sections med asides i hver section. Jeg har valgt at bruge section-elementet fremfor article-elementet, da jeg mener denne løsning er mere passende til netop denne sides behov. Dette bl.a. fordi jeg ser article-elementet være mere tekstpræget og høre til i eksempelvis blogs eller fora.

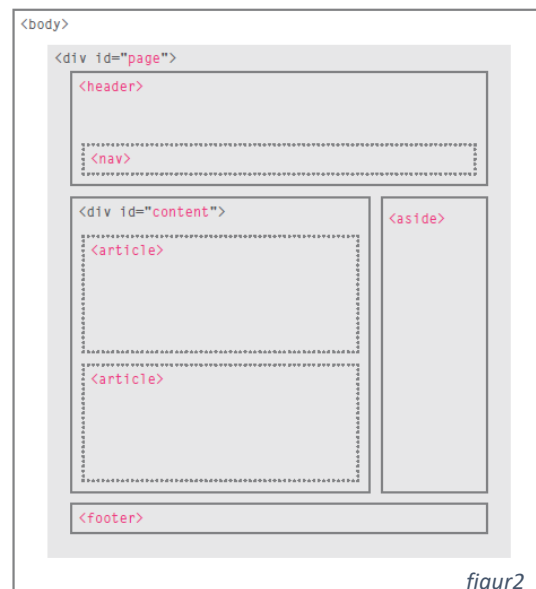
Navigations-barren er designet ved brug af Bootstrap v4, da dette library allerede har defineret en overskuelig, responsiv navigations-bar der på mindre skærme fungerer som en "toggle" knap der åbner og lukker for navigations-menuen. Samtidig indeholder Bootstrap v4 også et JavaScript library der bl.a. animerer en "glidende" visning af navigations-menuen ved brug af "toggle-knappen" på mindre skærme. (Duckett. J. 2011 s.471, Hjemmeside:Bootstrap)

Det førnævnte JavaScript library for Bootstrap v4 er placeret som det sidste før body-tagget lukker, sammen med det for JQuery og den interne (main.js). Disse placeres til sidst da JavaScript som tidligere nævnt skal interpreteres, og derfor er længere tid om at blive "læst" af browseren. Når disse placeres til sidst læses alt HTML og CSS før der interpreteres JavaScript. Dette gør at siden er klar til brug allerede før og mens JavaScript indlæses, hvilket giver en følelse af en god ydeevne på sitet. Samtidig er det vigtigt at alt HTML og CSS er læst af browseren inden JavaScript, såfremt der gennem JavaScript og JQuery oprettes *Event listeners* eller skal laves direkte ændringer i HTML/CSS. (Duckett. J. 2013 s.356)

For at gøre siden responsiv har flere elementer på siden fået MediaQueries. På denne måde kan et element få nye CSS properties og values afhængigt af eksempelvis skærmstørrelsen. På sidens Aside-elementer er yderligere benyttet Bootstraps responsive "12-column" design, der justerer kolonnens bredde i en række ud fra x antal 12.-dele. (UsingMediaQueries.pdf, Hjemmeside:Bootstrap)

DYNAMISK INDHOLD

Det dynamiske indhold på hjemmesiden igangsættes ved at aktivere event handlers eller listeners. På denne side er næsten alle events bundet til elementer som event handlers skrevet *inline* i HTML'en. Denne metode ses i dag som den forkerte måde at binde events til elementer, men som ny indenfor JavaScript var dette den mest overskuelige løsning. Sidens event handlers indeholder bl.a. onclick og onchange, der ved aktivering igangsætter en JavaScript funktion, der gennem kommandoer ændrer indholdet på siden og gør



figur2

dette dynamisk. Til validering af formularen er benyttet event listeners, der betragtes som den rette metode til dette formål. (Duckett. Jon. 2013 s.250-251)

Ser vi først på siden perfplader.html findes et select input med en inline event handler der "onchange" igangsætter funktionen "hideDiv". Denne funktion er skrevet i ren JavaScript-kode, og bruger statements til bl.a. at vise luftprocentberegneren, der ved ændring af input udregner luftprocenten ud fra det givne valg af beregner.

Ser man derefter på siden Mønsterguide, bruges funktionen "Choose" med 3 parametre til at komme igennem guiden. Ved klik på det valgte knap/billede defineres to variabler "title" og "alt" ud fra de respektive attribut-værdier. Herefter sendes disse variabler til en PHP-fil gennem en JQuery AJAX funktion, hvorefter PHP-filen kontakter serveren og henter data ud fra attribut-variablerne, der indsættes som html i næste skridt af mønsterguiden. (Duckett. J. 2013 s.388-395)

Funktionen Choose tilføjer derefter klasser til elementerne i næste skridt af mønsterguiden, for at disse ser "aktiverede" ud. Yderligere udføres et tjek ved brug af et loop, der fjerner "aktiverings"-klasserne fra elementer længere fremme i guiden, hvilket er aktuelt hvis brugeren skifter mening med et tidligere valg i guiden.

Afslutningsvis på funktionen Choose tilføjes klassen "selected", der markerer det klikkede element for bedre visuel feedback. Samtidigt fjernes klassen fra alle elementets "siblings" ifølge DOM-træet, såfremt brugeren skifter mening og klikker på et andet element. (Duckett. J. 2013 s.210)

Når mønsterguiden er gennemgået og den rette plade er valgt, ender man ved formularen hvor en forespørgsel på varen kan sendes. For at sikre indtastningerne i formularen er korrekte tilføjes event listeners i JQuery til input-felterne, der aktiverer "check-funktionerne" når fokus fjernes fra det aktive input-felt. Samtidigt returnerer disse funktioner en rød fejlmeddelelse, såfremt inputtet ikke stemmer overens med kravene. Når der trykkes på "Submit-knappen" udføres alle "check-funktioner", og formularen sendes kun hvis alle input stemmer overens med det forventede. (Hjemmeside:Form Validation)

Disse funktioner sikrer en glidende, dynamisk gennemgang af mønsterguiden, der samtidig håndterer gennemgangen hvis brugeren ombestemmer sig og ændrer valg, og ikke mindst hvis der laves tastefejl.

PERSPEKTIVERING

Sidens generelt simple opbygning, den glidende og dynamiske måde indholdet fremstår på, og ikke mindst den måde brugeren håndteres visuelt og dynamisk gennem brugen af JavaScript, gør siden både intuitiv at følge, og let at bruge for en eventuel kunde der leder efter perforerede plader. Mønsterguiden giver en visuel tilbagemelding på de forskellige parametre en perforeret plade besidder, hvilket bør lette produktøgningen for de mindre teknisk stærke kunder hos Netto Perforering.

Den samlede hjemmeside der skal offentliggøres i den nærmeste fremtid, er grundlæggende et WordPress-site, med brugerdefinerede PHP-sider i "baggrunden", der alle bruges af mønsterguiden (page-503.php). Der er som tidligere nævnt mange fordele ved at bygge et website op omkring WordPress, og i denne omgang var den primære årsag til dette valg at hjemmesiden fremover skulle være let at ændre og opdatere på stedet, således virksomheden undgår at erhverve en webudvikler igen.

Samtidigt er den måde ændringer håndteres på i et WordPress-site smart. WordPress indlæser en side ved at loope igennem indholdet, og indhente siden i fragmenter. Eksempelvis hentes header footer og de

forskellige sektioner hver for sig, da det hele foregår i PHP over en server. Dette resulterer i at en ændring i eksempelvis headeren kun skal laves en enkelt gang.

Skulle man skrive et site fra bunden burde dette gøres på samme måde igennem PHP, hvor alle fragmenterne laves individuelt, og siden opbygges ved at inkludere fragmenterne hver for sig, for at sikre kontinuitet igennem de elementer der vises på flere sider.

Personligt har det været en udfordring at dette projekt krævede både client-side og server-side programmering, da jeg inden projektet ingen erfaring havde med server-side programmering i aldelshed, ej heller med PHP-sproget i særdeleshed.

Processen bag udviklingen af projektet har dog været yderst succesfuldt og jeg kan ikke forestille mig nogen bedre måde at lære programmering på end at sidde med det i en længere periode, og langsomt lære ved at kæmpe sig igennem problemerne der opstår undervejs, kombineret med et kursus i webprogrammering client-side, hvor man kan støtte sig op ad et par gode lærebøger, fremfor at søge al viden over internettet.

At omskrive WordPress-sitet til html igennem denne opgave har været en udfordring, i og med alt design og opsætning i WordPress gøres automatisk, var det en udfordring at opsætte elementerne på præcis samme måde. I WordPress var designet på plads inden denne opgave blev udleveret, hvilket der både var fordele og ulemper ved. Fordelene ved dette er at man udelukkende behøver at koncentrere sig om programmeringsdelen, og da designet allerede var på plads, skulle man ikke bekymre sig om det så godt ud eller ej. En stor ulempe ved denne fremgangsmåde er dog at det er svært at overføre mange af de mindre effekter, funktion og finurligheder man har mulighed for at lave med designet igennem WordPress. Dette er bl.a. den primære årsag til mit brug af Bootstrap på sitet, da dette tilbyder nogle af de samme effekter, med gode resultater, uden et enormt tidsforbrug.

Havde jeg haft mere tid til at udføre denne opgave var mine største ønsker at skille mig af med Bootstrap, og skrive både navigationsmenuen og billedkarrusellerne fra bunden. Som tidligere nævnt er der flere fordele ved at bruge Bootstrap på en side, men man ender med kun at bruge stumper af koden, hvoraf resten er overflødig, og især Bootstraps JavaScript library er tungt for browseren at indlæse. Yderligere kan Bootstraps stylesheet mere eller mindre ødelægge designet, hvis det først tilføjes til siden efter dele af designet er færdiggjort, da der pludseligt er elementer der får flere properties. Det kan være tidskrævende at ændre alle disse properties, der før tilføjes af Bootstrap stylesheetet, havde en standardværdi der var tilfredsstillende.

At ændre de inline event handlers til event listeners ville samtidig være en prioritet hvis jeg havde mere tid med opgaven, da dette er den rette måde at koble events til et element, da man på denne måde ikke blander JavaScript ind i sit HTML, men adskiller disse.

Havde jeg mere tid ville jeg også have tilføjet en lille feature til luftprocentberegneren. Der gennem flere statements kunne tilføjes en beskrivende tekst hvis kun det ene af de to felter var udfyldt, eksempelvis *"indtast deling"*. Ydermere kunne en reduktion af decimaler i resultatet være på sin plads.

Et problem med en mønsterguide lavet med billeder er at man ikke kan se nøjagtig hvordan hulstørrelsen og delingen ser ud i 1:1. Det var et stort ønske fra chefen hos Netto Perforering at billederne skulle vises i 1:1 størrelse på skærmen, men jeg måtte skuffe ham og sige at det ikke kunne lade sig gøre. For det første skulle billederne på skærmen være så store at man ville miste overblikket ved at skulle vælge mellem eksempelvis 20 plader, men for det andet og nok mest problematisk, så kan det ikke lade sig gøre fordi kunderne vil se mønsterguiden fra et hav af forskellige skærme, med forskellig opløsning og størrelse. Det er muligt at styre størrelsen på billederne afhængigt af skærmopløsningen, men det er desværre umuligt at

tilpasse det efter størrelsen på skærmen. Løsningen jeg kom frem med til sidst blev at man, når pladen er valgt, kan åbne en printbar pdf-fil der kan vise perforeringen i 1:1 format.

Når websitet går live, bliver det spændende at se om de hos Netto Perforering kan mærke en forskel, og at analysere på antallet af forespørgsler ifht. før. Der er samtidig rig mulighed for at analysere brugernes adfærd på hjemmesiden via google analytics, og derved eksempelvis undersøge hvilke sider kunder forlader sitet fra, eller hvor stor en del der åbner mønsterguiden, som rent faktisk finder en plade og sender en forespørgsel. (Duckett. Jon. 2011 s.483-486)

Uanset hvad er denne hjemmeside en klar opgradering fra deres tidligere hjemmeside, og derfor håber jeg at de vil modtage flere forespørgsler over hjemmesiden, og samtidig reducere antallet af telefonopkald i løbet af dagen. Det kan også være de bare får flere opkald hvor kunderne allerede har undersøgt hvad de vil have, men bare har et par mindre spørgsmål i forlængelse af bestillingen.

KONKLUSION

Smedevirksomheden Netto Perforering havde et stort behov for at få en ny og opdateret hjemmeside, med mulighed for at søge i virksomhedens produkter, og efterfølgende sende en forespørgsel på dette produkt. Denne hjemmeside skulle have en produktsøgningsmulighed for kunder med høj såvel som lav teknisk viden om perforerede plader.

Det var samtidig et krav at hjemmesiden skulle laves i WordPress, da det fremover skulle være let at ændre og opdatere indhold på siden uden at kunne programmere. At bruge WordPress til at opbygge en hjemmeside er i den indledende fase en stor fordel, og programmet har mange funktioner der hjælper webudviklere hurtigt i gang med at opbygge en hjemmeside. Ønsker man at arbejde udenfor de rammer WordPress tilbyder, skrive de enkelte sider i server-side sproget PHP.

For at imødekomme virksomhedens behov har jeg udviklet en hjemmeside igennem WordPress, med enkelte sider skrevet i PHP, alle for at imødekomme løsningen med en mønsterguide. Hjemmesiden er simpelt opbygget, og gøres intuitiv og dynamisk gennem brug af JavaScript. Forhåbentligt vil denne nye hjemmeside føre til et øget antal forespørgsler over hjemmesiden, og reducere antallet af opkald i løbet af en dag, for at give de få ansatte lidt ekstra tid i løbet af dagen, til at få lavet de småting der altid mangler.

LITTERATURLISTE

HJEMMESIDER

Bootstrap : www.getbootstrap.com

Animations : <https://daneden.github.io/animate.css/>

Form Validation : <https://jsfiddle.net/sonalin123/gkz8nLrx/>

LÆREBØGER

Duckett. Jon. 2011. *HTML & CSS design and build websites*. John Wiley & Sons, Inc

Duckett. Jon. 2013. *JAVASCRIPT & JQUERY interactive front-end web development*. John Wiley & Sons, Inc

ANDET MATERIALE

UsingMediaQueries.pdf