

Eksamensdokument for Webutvikling

2023

1. Vanskelighetsgrad

1.1 Vanskelighetsgrad A

Oppgaven beskriver ulike vanskelighetsgrader og spesifiserer krav for disse. Etter nøye vurdering av de ulike vanskelighetsgradene ble konklusjonen at A-nivået er passende for mitt kompetansenivå. Noen av kravene til A-nivået spesifiserer at alle sidene skal ha microdata som validerer i Rich Results, hover-effekter på desktop men ikke mobile, at nettsiden skal ha et favicon, og en «tilbake til toppen»-knapp på alle sider. I tillegg til alle A-kravene skal de andre vanskelighetsgradenes krav inkluderes i prosjektet.

2. Oppgaveprosess

2.1 Redegjørelser/forutsetninger

Oppgavebeskrivelsen var krevende siden det var nødvendig å grundig gjennomgå alle vanskelighetsgradenes krav og se demovideoen flere ganger for å få en passende forståelse av hva oppgaven forventet. Dette var et essensielt steg før utviklingsprosessen av nettsiden kunne starte, og dannet et godt grunnlag for hva slags forutsetninger og strategiske valg som ble avgjort når utviklingsprosessen var satt i gang. Takket være dette ble utviklingsprosessen relativt problemfri.

Først måtte oppgaven tolkes på flere steder der oppgavebeskrivelsen og demovideoen manglet informasjon. Eksempelvis ble det besluttet å utvikle nettsiden basert på mobile-first – altså begynte nettsideutviklingen for mobile-versjonen før desktop-versjonen. Mobile-first utvikling i forhold til desktop-first utvikling hadde ingen klare begrensninger eller rammer beskrevet i oppgavebeskrivelsen, og derfor var det en antagelse at nettsideutviklingen stod fritt ut fra hva man selv foretrekker. Å utvikle nettstedet ut ifra mobile-first var derfor en beslutning basert på kunnskap tilnærmet fra forelesninger, der mobile-first og dets viktighet for nettsidens tilgjengelighet og brukervennlighet ble vektlagt av foreleseren. Denne beslutningen effektiviserte utviklingen av nettsiden, da det oppleves lettere å tilpasse designet fra mobile til desktop enn motsatt.

Videre var det ingen spesifikke størrelser til media queries angitt i oppgavebeskrivelsen, og dermed måtte dette antas basert på hva som kan observeres i demovideoen. Media queries ble iverksatt i nettsiden for å tilpasse designet fra mobile til desktop, men også for å justere på størrelsene på skrift, bilder og andre diverse elementer. Dette tiltaket hadde som formål å sikre responsivitet og

brukervennlighet, samt å oppnå en nettside mest mulig lik demovideoen. Noe verdt å bemerke i kontekst med responsiv design er at viewport width måleenheten (vw) har på noen steder blitt benyttet, men dette i mindre grad enn i tidligere personlige prosjekter med mer kreativ frihet. I stedet har pixel måleenheten vært hoved-måleenhet under utviklingen, da som et bevisst valg for å opprettholde en konsistens presentasjon med hva som demonstreres i demovideoen.

2.2 Utfordringer

Utviklingen av nettsiden var en prosess med relativt få utfordringer, men noen områder var definitivt mer krevende enn andre. Spesielt var det to problemområder under oppgaveprosessen, nemlig implementering av Schema microdata og bruk av Flex-grow på Portfolio.html. Disse utfordringene ble løst til slutt, men en var mer suksessfull enn den andre.

Microdata ble iverksatt jevnlig under oppgaveprosessen i alle html-sider til beste evne, da hovedsakelig med hjelp av Google Structured Data Markup Helper. Utfordringen med microdataen oppstod på grunn av manglende ressurser på flere nivåer. Aller først manglet mesteparten av kilder rundt Schema Markup informasjon om bruk av mikrodata i stedet for JSON-LD. Denne informasjonsmangelen kan antas å stamme fra at Schema selv anbefaler å bruke JSON-LD formatet. Dette førte til komplikasjoner med oppgaveprosessen siden oppgavebeskrivelsen begrenser formatet til microdata, som sett her: «Alle sidene skal ha en form for microdata, vurder innholdet på de ulike sidene og velg fornuftig microdata basert på det.» (Eksamensoppgave ITF10511, 2023). I tillegg var det ekstra krevende å legge inn microdataen, mest fordi det var få kilder som ga konkrete eksempler av riktig og passende microdata-bruk for oppgave-nettsiden. Mye tid ble derfor investert i å forsøke å finne relevante kilder som kunne bidra til oppgaven, men dessverre viste det seg å være ganske få. ChatGPT ble forsøkt brukt, men den fant bare opp Schema data som ikke eksisterte, og dermed ett mindre hjelpeverktøy for situasjonen. Dette resulterte i at mengden microdata var mindre enn ønsket, selv om microdata ble lagt inn til beste evne med de tilgjengelige ressursene. Dette var også uheldig siden tidligere kunnskaper rundt temaet var på mange måter begrenset til produkt-side microdata, ut ifra hva som ble lært i forelesninger og brukt i tidligere oppgaver. Microdata ble derfor en stor utfordring på grunn av manglende ressurser med relevant innhold og konkrete eksempler, men også på grunn av begrenset erfaring rundt temaet.

Videre var det andre problemområdet bruken av flex-grow i portfolio.html siden. Opprinnelig var en annen fremgangsmåte brukt, frem til en rutinesjekk av oppgavekravene avdekket at man burde bruke flex-grow i stedet. Derfor var neste steget å gjøre om oppførselen til artikkelkortene ved å bruke flex-grow. For å få til dette ble ChatGPT en viktig ressurs i å utarbeide en fremgangsmåte som ga riktig løsning i forhold til oppgavebeskrivelsen. Allikevel var dette en utfordring da det krevde mye frem og tilbake i media queriene og hyppige visninger av demovideoen for å sikre et resultat som var tilfredsstillende. I tillegg ga ChatGPT mye feil tilbakemelding, da oppførselen til artikkelkortene var

avansert for ChatGPT å forstå. Her ble det mye arbeid frem og tilbake og testing av hvilken kode som fungerte eller ikke. Allikevel ble sluttresultatet bra og utfordringene ble håndtert.

2.3 Sammendrag

Oppsummert var nettsideutviklingen en prosess som krevde jevnlig sjekker i oppgavebeskrivelsen for å sikre ønsket resultat, og en grundig gjennomgang før prosessen startet. Noen steder krevde oppgaven også egne tolkninger og antagelser, som for eksempel å velge mellom å utvikle i henhold til mobile-first eller desktop-first, og å finne ut av passende størrelser til skjermoppløsning i media queries. Ellers var oppgaveprosessen relativt problemfri, med hovedsakelig to problemområder som tok ekstra tid. Det var utfordringer med å finne relevante, passende kilder for microdata-bruk, og at portfolio-siden måtte gjøres på nytt - men begge ble løst til slutt til beste evne. I fremtidige prosjekter vil det være et mål å lære mer om flex og hvordan man kan anvende dette, samt å lære mer om korrekt bruk av microdata. Disse tiltakene vil bidra til økt responsivitet og SEO i fremtidige nettsideprosjekter, samt effektivisere utviklingsprosessen med mindre tid brukt på disse problemområdene.

3. Kilder

I tillegg til å være dokumentert her kan man også finne kilder for eksempelvis bilder, kode og fonts i readme filen i repositoriet eller inni CSS og HTML-filene.

3.1 Bilder

1. 400x400 (img-mappe)
Placeholder bilde hentet fra <https://placeholder.co/> og lagret fra <https://placeholder.co/400x400>
2. business (img-mappe)
Bilde av [6689062](#) fra [Pixabay](#) - <https://pixabay.com/photos/business-computer-mobile-smartphone-2846221/>
3. busy (img-mappe)
Bilde av [Lukas](#) fra [Pixabay](#) - <https://pixabay.com/photos/busy-home-desk-paper-laptop-view-1972166/>
4. computer (img-mappe)
Bilde av [Thorsten Frenzel](#) fra [Pixabay](#) - <https://pixabay.com/photos/computer-technology-laptop-company-3324200/>
5. laptop (img-mappe)
Bilde av [StockSnap](#) fra [Pixabay](#) - <https://pixabay.com/photos/analytics-charts-traffic-marketing-925379/>
6. work (img-mappe)
Bilde av [Veeka Skaya](#) fra [Pixabay](#) - <https://pixabay.com/photos/flat-lay-blogger-blog-laptop-2879273/>

7. workspace (img-mappe)

Bilde av [Free Photos](#) fra [Pixabay](#) - <https://pixabay.com/photos/workspace-coffee-laptop-macbook-1280538/>

3.2 Andre kilder

Font og ikoner

1. Fontawesome 4 ikoner er hentet fra Fontawesome sin side for ikonpakke versjon 4 - <https://fontawesome.com/v4/icons/>. Disse ikonene ble implementert med hjelp av en ikon-tutorial av w3schools - <https://www.w3schools.com/icons/>.
2. Favikon ble laget ved å overføre bildefilen til en ico-fil gjennom favicon.io - <https://favicon.io/>. Videre ble favikonet lagt inn ved hjelp av en tutorial fra w3schools - https://www.w3schools.com/html/html_favicon.asp.
3. Inter – font hentet fra Google Fonts. <https://fonts.google.com/specimen/Inter>

Kode

1. Microdata kode i html-koden til alle sider er lagt til med hjelp av Google Structured Data Markup Helper - <https://www.google.com/webmasters/markup-helper/>
 - a. Google. (u.å). *Kodehjelp for strukturerte data*. <https://www.google.com/webmasters/markup-helper/>.
2. Microdata kode i about.html er manuelt hentet fra Schema.org ved å bytte ut koden lagt inn fra Google Structured Data Markup Helper. Altså ble koden fra Markup Helper brukt, men noen steder ble itemtype LocalBusiness byttet ut med Organization og itemprop fra Organization ble lagt til i stedet - <https://schema.org/Organization>.
 - a. Schema. (2023, 17. oktober). *Welcome to Schema.org*. Schema.org. <https://schema.org/>.
 - b. Schema. (2023, 17. oktober). *Organization*. <https://schema.org/Organization>.
3. Microdata i flere/alle av html-sidene med WebPage Schema er lagt inn ut ifra denne eksempelkoden fra GitHub, publisert av MilanAryal - <https://gist.github.com/MilanAryal/ee861d7a065cc05868d9>.
 - a. MilanArval. (u.å). *An example of how to mark up a HTML5 webpage using the schema.org schemas and microdata*. GitHub. <https://gist.github.com/MilanAryal/ee861d7a065cc05868d9>
4. Microdata har blitt lagt inn med hjelp av Schema sine nettsider. Typer som har blitt brukt er Article, ItemList, Organization, PostalAddress, ReadAction, SiteNavigationElement, WebPage og WPFooter
 - a. Schema. (2023, 17. oktober). *Welcome to Schema.org*. Schema.org. <https://schema.org/>.

- b. Schema. (2023, 17. oktober). *Article*. Schema.org. <https://schema.org/Article>
 - c. Schema. (2023, 17. oktober). *ItemList*. Schema.org. <https://schema.org/ItemList>
 - d. Schema. (2023, 17. oktober). *Organization*. Schema.org. <https://schema.org/Organization>
 - e. Schema. (2023, 17. oktober). *PostalAddress*. Schema.org. <https://schema.org/PostalAddress>
 - f. Schema. (2023, 17. oktober). *ReadAction*. Schema.org. <https://schema.org/ReadAction>
 - g. Schema. (2023, 17. oktober). *SiteNavigationElement*. Schema.org. <https://schema.org/SiteNavigationElement>
 - h. Schema. (2023, 17. oktober). *WebPage*. Schema.org. <https://schema.org/WebPage>
 - i. Schema. (2023, 17. oktober). *WPFooter*. Schema.org. <https://schema.org/WPFooter>
5. Microdata har blitt lagt inn med hjelp av ChatGPT på steder kommentert I HTML-koden. Hovedsakelig ble ChatGPT brukt for å finne ut av hva slags microdata som kan legges på en a href link som tilhører en article. Først startet jeg ved å vise den HTML-koden for en article som hadde en tilhørende "les mer"-knapp/link, og deretter spurte jeg den hva slags microdata jeg burde bruke på den linken/Knappen som tilhører article-koden. Da svarte den med å legge inn `itemprop="potentialAction"` og `itemscope itemtype="http://schema.org/ReadAction"` på a-taggen, og skriften inni a-taggen ble wrappet I en span-tag med `itemprop=name`.
- a. OpenAI. (2023). ChatGPT (Januar 2022 versjon) [Stor språkmodell]. <https://chat.openai.com/>
6. Flex-grow løsning på Portfolio.html er utarbeidet med hjelp av ChatGPT. Først startet jeg ved å spørre den om å vise meg hvordan man bruker flex-grow, og deretter spurte jeg den hvordan jeg kan få artikkelkortene til å vokse og minske seg i en liste. Deretter sendte jeg html koden og css koden til portfolio siden, og beskrev hva slags oppførsel jeg ønsket på artikkelkortene i tråd med oppgavebeskrivelsen. Dette ble fort avansert for den, så derfor måtte jeg modifisere det i stor grad for å få det funksjonelt. Koden som har blitt utarbeidet med ChatGPT har en kommentar i CSS-arket.
- a. OpenAI. (2023). ChatGPT (Januar 2022 versjon) [Stor språkmodell]. <https://chat.openai.com/>
7. Løsning for å få lavere opacity/overlay på bakgrunnsbilde/video i alle html-sidene er hentet fra en video av YouTube-skaperen Kevin Powell.
- a. Kevin Powell. (2022, 18. august). *Lower the opacity of a background-image with CSS* [Video]. YouTube. <https://www.youtube.com/watch?v=IRPguPbovvo>
8. Løsning for kartet på about.html er utarbeidet med hjelp av ChatGPT. Først startet jeg ved å fortelle hva slags oppførsel jeg ønsket at kartet skulle ha. Da skrev jeg at jeg har et google map iframe som jeg vil skal ta opp 100% vidde av skjermen. Dette kom den med en god

løsning for. Videre opplevde jeg en utfordring med denne koden når skjermstørrelsen ble 1600px og større, altså når html-siden har nådd sin maksvidde. Chatgpt fant en løsning på dette, som var å resette margin på kartet i en media query som starter på 1600px. Koden som har blitt utarbeidet med ChatGPT har en kommentar i CSS-arket.

- a. OpenAI. (2023). ChatGPT (Januar 2022 versjon) [Stor språkmodell].

<https://chat.openai.com/>

9. Løsning for back-to-top knappens oppførsel ved skjermvidde 1600px og over er utarbeidet med hjelp av ChatGPT. Her sendte jeg HTML-koden og CSS-koden for knappen og spurte den hvordan jeg kan fikse problemet til knappen, der den går utenfor html-innholdet når html-siden har nådd sin maksbredde på 1600px. Deretter spurte jeg om hvordan jeg kan gjøre at knappen har plass til litt padding. Da kom den frem til at jeg kan legge til kode for right ved bruk av calc under en media query som starter fra 1600px og over. Koden som har blitt utarbeidet med ChatGPT har en kommentar i CSS-arket.

- a. OpenAI. (2023). ChatGPT (Januar 2022 versjon) [Stor språkmodell].

<https://chat.openai.com/>

10. Kode for e-post form i about.html er hentet fra en artikkel i Getform og modifisert til å passe nettsiden.

- a. YÜCEL, M. (2020). *How to Create an HTML Form That Sends You an Email*.

Getform. <https://blog.getform.io/how-to-create-an-html-form-that-sends-you-an-email/>

Annet

1. Eksamensoppgave ITF10511. (2023, 15. november). Eksamensoppgave i Webutvikling.

Hentet fra <https://hiof.instructure.com/courses/7597/assignments/36561>

2. HTML-kode er validert gjennom W3C Markup Validation Service - <https://validator.w3.org/>

- a. W3C. (u.å). *Markup Validation Service*. <https://validator.w3.org/>.

3. HTML-kode er validert gjennom Schemas verktøy for validering av microdata -

<https://validator.schema.org/>

- a. Schema. (2023, 17. oktober). *Markup Validator*. <https://validator.schema.org/>.

4. HTML-kode er validert gjennom Googles test for rike resultater (Rich Results Test) -

<https://search.google.com/test/rich-results>

- a. Google. (u.å). *Støtter siden din rike resultater?* <https://search.google.com/test/rich-results>.