

# Inlämningsuppgift

## Innehållsförteckning

1. Inledning och Kodstruktur
2. SCSS-struktur
3. Sökmotoroptimering (SEO)
4. Felsökning av kod
5. Agil arbetsmetod
6. AI-verktyg och tjänster

## Inledning och Kodstruktur

Vi har strukturerat upp koden i indexfilen i root-mappen. SCSS-filerna finns i en mapp med en partials-mapp, medan vi även har mappar för bilder och JavaScript i root. Genom att använda GitHub kan vi enkelt följa alla commits, uppdateringar och arbetsprocessen.

Som ett tillägg till uppgiften har vi skapat detta dokument för att komplettera vissa betygskriterier som kanske inte helt uppfylls av den inlämnade uppgiften. Jag vill påvisa några av mina kunskaper inom detta område.

Vi har också uppdaterat Figma-filen med komponenter för att stödja designen av wireframes. Vi har strukturerat och designat dem med en metod som jag kallar Medium-fi. Detta innebär att de inte är tillräckligt detaljerade för att klassas som high-fi wireframes, men de är fortfarande av bättre kvalitet än de mer

grundläggande low-fi wireframes, som ofta enbart innehåller kryss för bilder och liknande.

## SCSS-struktur

Vi har strukturerat vår SCSS-kod genom att dela upp den i flera partial-filer. Vi använder en grundläggande uppdelning där vi har en fil för grundläggande stilmallar som täcker både normal- och mobilversioner, en fil specifikt för tablets, och en annan för desktops. Utöver dessa har vi separata filer för variabler, mixins och functions.

När vi började strukturera våra variabler insåg vi att det går att skapa en variabel för nästan varje egenskap i alla selectors. Detta liknar hur ramverk som Tailwind fungerar, där nästan varje pixel representeras av en klass, vilket skapar ett stort antal klasser. Samtidigt erbjuder ramverk som Bootstrap en mer komponentbaserad struktur, med färre men mer återanvändbara klasser. Efter att ha reflekterat över detta ansåg vi att det inte alltid är nödvändigt att skapa en variabel för varje enskild property. Däremot vissa egenskaper, som `display: none;`, är mer logiska att skriva variabel, att skapa en variabel som `$display-none;`. Det finns då tillfällen variabler kan underlätta, som i fallet med olika visningslägen, exempelvis `$deviceShow-block;` och `$deviceShow-flex;`, där `display` ändras beroende på enhetsstorlek. Å andra sidan finns det properties som `position: absolute;` som alltid kommer att vara samma inom en viss selector, vilket gör det onödigt att skapa en variabel.

SCSS ger större kontroll över hur man bygger upp sitt projekt. Man kan skapa sitt eget ramverk med specifika komponenter anpassade efter ett visst projekt, eller designa ett generellt ramverk för återanvändning i flera projekt. Det viktigaste är att hitta en balans mellan när det är värt att använda variabler och när det är enklare att skriva egenskaper direkt i koden.

## Sökmotoroptimering

Vi har insett att Google är den största sökmotorn och därmed den viktigaste att anpassa sig till när det gäller SEO. Samtidigt verkar det vara enklare att optimera för mindre sökmotorer, som Bing, eftersom Googles kriterier för SEO kan vara mer komplexa att uppfylla.

Oavsett vilken sökmotor man riktar sig mot är det grundläggande att varje hemsida har en favicon och en korrekt titel. En välformulerad meta-beskrivning är också viktig, då den ger andra sökmotorer möjligheten att använda den beskrivning man själv vill ska representera sidan. Google kan välja att antingen använda denna beskrivning eller kombinera innehåll från sidan för att skapa en egen sammanfattning av sidans innehåll.

För att förbättra SEO bör man använda ord och fraser som är relevanta för ämnet och som både lockar och beskriver sidans innehåll väl. Om sidan till exempel handlar om mobilbanktjänster kan en kort och tydlig beskrivning av vad besökaren kan förvänta sig vara mycket värdefull, oavsett om det är en nedladdningssida, informationssida eller en sida med öppettider. En välplanerad struktur är nyckeln till bra SEO.

Google erbjuder en omfattande guide som täcker allt från listor och stjärnbetyg till fördelarna med så kallade "Rich Results", vilket kan vara särskilt användbart för receptsidor, produktförsäljning och liknande. Dessa förbättringar kan göra det lättare för sökmotorer att presentera relevant och lockande information direkt i sökresultaten.

En strategi som kan vara effektiv är att inledningsvis optimera för Bing. När sidan får mer trafik och relevans ökar, tenderar Google att börja ge sidan mer uppmärksamhet. Det spekuleras i att en av de stora parametrarna för Google är just hur relevant innehållet uppfattas baserat på sidans trafik.

## Felsökning av kod

Vid felsökning av kod använder jag kontinuerligt webbläsarens utvecklingsverktyg (DevTools). Dessa är särskilt användbara för att identifiera problem relaterade till specificitet, vilket gör det lätt att avgöra om en stilregel har tillämpats eller om den är överstruken av en annan regel. Dessutom hjälper verktygen mig att testa och ändra värden direkt i webbläsaren innan jag implementerar ändringarna permanent i koden. Genom att kunna se elementens padding och margin när jag hovrar med musen, får jag en djupare förståelse för hur de är placerade på sidan, hur flexbox-systemet hanterar sina objekt, samt hur grid-strukturen fungerar.

Under HTML- och CSS-kursen har vi också använt andra felsökningsmetoder, såsom terminalutgången för SCSS, för att spåra eventuella problem vid koduppdateringar. När fel uppstår,

som vid syntaxproblem eller när funktionalitet upphör att fungera, har terminalen varit ovärderlig för att identifiera vad som gått fel. Ett konkret exempel var när `rem()`-funktionen slutade fungera vid ny uppdatering och jag behövde skapa en ny lösning för att hantera storlekar i projektet.

## Agil arbetsmetod

I vårt projekt har det agila arbetssättet främst inneburit att följa de videolektioner som tillhandahållits i skolan och arbeta utifrån den struktur och information som varit tillgänglig. Vår metod har varit att arbeta uppifrån och ned, sektion för sektion, och göra varje sektion responsiv innan vi implementerade dark mode.

Det agila arbetssättet verkar tillämpas mer effektivt i samarbete mellan olika roller, som designers, frontend-utvecklare (både statisk och dynamisk), och backend-utvecklare. Det handlar om att planera, bygga komponenter och successivt finslipa projektet. När det gäller HTML/CSS känns den agila metoden mindre tillämplig, förutom att teammedlemmar kanske kan få ansvar för specifika sidor i den statiska front-end-fasen.

Men det som går är skapa alla element och innehåll, sedan styla dem och göra dem responsiva. Efter detta anpassar vi för tillgänglighet och sökmotoroptimering (SEO). När dessa grundläggande funktioner är på plats, kan vi implementera inställningar som dark mode eller andra funktioner som sidan kan behöva. Ett viktigt steg är att säkerställa att varje del av koden följer teamets struktur för tillgänglighet innan den avslutas.

När alla dessa delar är på plats anser jag att arbetet för Frontend Static kan betraktas som färdigt.

## AI verktyg och tjänster

Vi använder AI-verktyg som ChatGPT, Gemini och tjänster som Builder.io regelbundet i vårt arbete. Dessa textbaserade chatbotar, med avancerad informationshantering och grammatisk struktur, fungerar som ett effektivt bollplank som hjälper oss att repetera uppgifter och skapa en mer strömlinjeformad arbetsprocess.

Personligen har jag använt ChatGPT flitigt för att skriva detta dokument. Genom att ständigt ställa frågor om repetitiva uppgifter, be om förklaringar och få hjälp med information, har jag kunnat effektivisera mitt arbete. Många av lösningarna i min kod är resultat av svar från ChatGPT eller Gemini. Trots deras förmåga att snabbt erbjuda lösningar, är det viktigt att vara medveten om att den kod som genereras inte alltid är 100% korrekt. Det krävs kontinuerlig interaktion och förtydliganden för att uppnå det önskade resultatet, och det är inte möjligt att skapa ett fullskaligt projekt enbart med hjälp av dessa verktyg.

Däremot är AI-verktyg en ovärderlig tillgång för att arbeta parallellt med andra metoder och tjänster som kraftfulla sökmotorer och assistenter. Gemini kan till exempel kopplas till e-post, Google Docs och andra verktyg, vilket underlättar dokumentation och skrivprocessen.

Ett konkret exempel är hur jag använde ChatGPT för att automatisera processen att omvandla alla egenskaper i alla selektorer till variabler i min SCSS-fil, ett arbete som har blivit betydligt mer effektivt tack vare AI.