

PROCESSANALYS

Elina Samuelsson

2023-03-26

AWH201 - Avancerad webbutveckling med HTML och CSS

INTRODUKTION

Ramverk för front-end utveckling används för snabb och enkel framtagning av hemsidor. Eftersom ramverk ofta kommer med färdigskrivna CSS- och JavaScriptfiler behöver man nästan uteslutande koda i HTML. Förutom enkelheten har ramverk också fördelar som tillgång till färdigbyggda komponenter och ibland avancerade funktioner, och eftersom ramverket är kvalitetstestat i förväg uppkommer också väldigt få buggar (Bose, 2023).

Responsivitet är en av de viktiga funktioner som ramverk erbjuder. De flesta ramverk säger sig vara designade med mobile first i åtanke, men anpassar sig automatiskt till större skärmar.

Beroende på vilket ramverk man väljer kan dock inlärningskurvan vara lång, och varje gång man byter ramverk behöver man lära sig ett nytt syntax (Vaskin, 2019).

Harcourt (2021) påpekar att ramverkets fördelar är en av dess största nackdelar. Eftersom ramverket är skrivet för att inkludera kod för varje möjlig lösning hemsidan kan tänkas behöva, är filerna fyllda med kod som troligtvis aldrig kommer att användas. Problemet kan kringgås med hjälp av verktyg såsom PurgeCSS - som rensar CSS-filer på all oanvänd kod - i den mån att ramverket är kompatibelt (ibid), och vissa ramverk erbjuder modulär nedladdning där man enbart hämtar hem dem delar som är relevanta för projektet.

Ytterligare nackdelar är att hemsidor byggda med hjälp av ramverk ofta får ett liknande utseende. Mary Collins (2016) skriver uteslutande om Bootstrap i sin artikel och säger då att på grund av dess grid-system och färdigbyggda komponenter skapar en igenkännbar stil. Eftersom det finns ramverk som imiterar Bootstraps grid-layout anser jag att även dessa, åtminstone delvis, faller inom samma kategori.

VAL AV FRAMEWORK

Ramverk jag valt bort:

Skeleton

Skeleton är en minimal boilerplate för mindre projekt som inte behöver mer än en layout och positionering (Thakur, u. å.). Namngivningen bakom Skeleton är enkel att sätta sig in i och eftersom Skeleton är så minimal går det snabbt att lära sig.

För ett litet projekt som denna examinationsuppgift kan Skeleton vid första ögonkast verka vara det perfekta valet men eftersom Skeleton är så avskalat fanns rädslan att det inte skulle räcka till för de funktioner jag vill att min hemsida ska ha. Därför valde jag bort även Skeleton.

Foundation

Foundation är ett massivt system som skryter om att vara det mest avancerade alternativet i världen. Foundation erbjuder anpassning på högre grad än andra ramverk, vilket tillåter unika designer som bryter loss från tropen att hemsidor byggda på ramverk ofta ser likadana ut (Zahoor, u. å.).

Nackdelen med detta omfattande system är att det på grund av storleken och anpassningsbarheten är att Foundation blir svårt att lära sig.

Boilerplate

Boilerplate valde jag bort eftersom jag fann det svårt att hitta information om bortom deras egen hemsida. Enligt dem själva hjälper ramverket till att snabbt bygga robusta hemsidor och webbappar.

Dokumentationen är enligt mig bristfällig jämfört med ramverk som Bootstrap eller Foundation, och om man stöter på problem misstänker jag att det kan vara svårt att hitta hjälp online.

Bulma

Ett ramverk som jag funderade över men slutligen valde bort dels eftersom det inte nämns som ett alternativ i uppgiften, och dels eftersom det inte erbjuder fullt samma variation av funktionalitet som Bootstrap var Bulma; ett relativt nytt ramverk jämfört med många av de mer kända och populära ramverk som används idag (Juviler, 2022).

Utöver de gemensamma fördelarna som Bulma delar med alla ramverk kan de skryta med ett konsekvent syntax som är enkelt att lära sig. Bulma erbjuder också modularitet i form av separata .sass-filer som alla kan importeras individuellt för att hålla nere filstorleken (ibid).

Tailwind

Tailwind var ett system som också intresserade men valdes bort i förmån för Bootstrap. Förutom att Tailwind inte erbjuder förbyggda komponenter att klippa och klistra in i projektet har ramverket många fördelar (Shankar, 2022). Genom att inte erbjuda färdiga komponenter har man större kreativ frihet att skapa unika lösningar.

En annan fördel är att Tailwind förlitar sig starkt på JavaScript för i kompileringen ta bort all CSS som inte används (Thakur, u. å.). På så vis håller sig Tailwind lättviktigt.

En av de nackdelar jag ser med Tailwind är att man måste sätta sig in i namngivningssystemet för att effektivt kunna använda sig av ramverket. Detta i samband med avsaknaden av förbyggda komponenter är orsaken till att jag valde bort Tailwind.

Ramverket jag valde:

Bootstrap

Bootstrap är det ramverk som mitt val föll på i slutändan, till stor del eftersom det är ett populärt ramverk där det finns många resurser att ta lärdom, eller hjälp, från även utanför deras egen dokumentation (Thakur, u. å.). Jag har även kollat närmare på Bootstrap tidigare och kände att det "försprång" det gav mig var hjälpsamt.

Utöver de gemensamma för- och nackdelarna har Bootstrap dessutom ett stort bibliotek av färdigbyggda komponenter att ta inspiration från, eller kopiera (ibid).

Så bra som jag hoppades att Bootstrap skulle vara baserat på mina tidigare erfarenheter har detta djupdyk visat att det finns många problem med ramverket. Jag önskar att jag bortsett från de fördelar som Bootstrap har och kollat mer på nackdelarna för att göra ett mer utbildat beslut i vilket ramverk jag skulle valt.

Bootstrap är väldigt svårt att skriva över om man behöver göra ändringar, eftersom det använder sig extensivt av !important (Pattakos, 2023). Flera gånger under projektet har jag haft svårigheter att göra ändringar i komponenter eller inte fått specifika element att alls fungera. Tid som jag hade kunnat spendera på att utveckla min kod har istället fått spenderas på troubleshooting eller att tänka om min layout helt.

Som nämnt ovan har Bootstrap också ett eget ID, och dem sidor som använder sig av Bootstrap ser ofta likadana ut.

REFLEKTION

Mina tankar kring ramverk generellt är positiva trots de svårigheter jag haft under detta projekt. Med rätt kunskap inom rätt ramverk kan man skapa vackra hemsidor på kort tid. Risken finns att dessa hemsidor saknar kreativitet och liknar många andra sidor på internet. Jag ser dock inte detta som en nackdel i alla situationer, utan det beror på vad projektet kräver. Om hemsidans layout och funktioner är bekanta innebär det dessutom att hemsidans besökare naturligt kommer att kunna navigera sig runt på sidan enklare.

Det är också betryggande att veta att den kod man förlitar sig på håller hög kvalitet och har ett supportsystem att förlita sig på.

Att inte behöva tänka på responsivitet är också en massiv fördel som ramverk för med sig, är något jag uppskattar. Att hitta rätt brytpunkter och att sedan anpassa koden utefter detta är komplicerat och utan eftertanke riskerar man att skriva onödigt många regler som jobbar mot varandra.

Ytterligare en fördel jag ser är chansen att utveckla min egen kod genom att lära mig från ramverken jag väljer att använda mig av. Talangfulla kodare jobbar hårt för att ramverken ska

fungera så bra, och så effektivt som möjligt och det finns mycket att lära.

Jag kommer absolut att använda mig av ramverk även framöver då jag anser att fördelarna överväger nackdelarna.

REFERENSER

Bose, S. (2023). *Top 5 CSS Frameworks for Developers and Designers*. Browserstack. Tillgänglig: <https://www.browserstack.com/guide/top-css-frameworks>. [2023-03-26].

Collins, M. (2016). *Web Design Trends: Why Do All Websites Look The Same?* Friday. Tillgänglig: <https://www.friday.ie/blog/why-do-all-websites-look-the-same/>. [2023-03-26].

Harcourt, H. (2021). *PurgeCSS: Remove unused CSS code*. LogRocket. Tillgänglig: <https://blog.logrocket.com/purgecss-remove-unused-css-code/>. [2023-03-26].

Juviler, J. (2022). *The Bulma CSS Framework: What It Is and How To Get Started*. Hubspot. Tillgänglig: <https://blog.hubspot.com/website/bulma-css>. [2023-03-26].

Pattakos, A. (2023). *9 Best CSS Frameworks in 2023*. aThemes. Tillgänglig: <https://athemes.com/collections/best-css-frameworks/>. [2023-03-26].

Shankar, A. (2022). *5 Best Frameworks for Front-end Developers*. Atatus. Tillgänglig: <https://www.atatus.com/blog/5-best-css-frameworks/>. [2023-03-26].

Thakur, A. (u. å.). *14 Best CSS Frameworks/Libraries for Front-End Developers*. Geekflare. Tillgänglig: <https://geekflare.com/best-css-frameworks/>. [2023-03-26].

Vaskin, U. (2019). *Pros and Cons of Using a CSS Framework*. Medium. Tillgänglig: <https://medium.com/@ustinvaskin/pros-and-cons-of-using-a-css-framework-21ea722b42e>. [2023-03-26].

Zahoor, I. (u. å.). *What is the difference between Bootstrap and Foundation?* Educative. Tillgänglig: <https://www.educative.io/answers/what-is-the-difference-between-bootstrap-and-foundation>. [2023-03-26].