Projektuppgift

DT057G

Moment 5
Projektrapport

Albin Rönnkvist

Mittuniversitetet
MID SWEDEN UNIVERSITY

Campus Härnösand Universitetsbacken 1, SE-871 88. **Campus Sundsvall** Holmgatan 10, SE-851 70 Sundsvall.
Campus Östersund Kunskapens väg 8, SE-831 25 Östersund.
Phone: +46 (0)771 97 50 00, Fax: +46 (0)771 97 50 01.

MITTUNIVERSITETET
Avdelningen för informationssystem och -teknologi

Författare: Albin Rönnkvist, alrn1700@student.miun.se
Utbildningsprogram: Webbutveckling, 120 hp
Huvudområde: Datateknik
Termin, år: HT, 2017

Sammanfattning

I denna rapport kommer jag gå igenom förberedelserna och skapandet av min projektwebbplats i kursen webbutveckling 1. I projektet använder jag de kunskaper jag fått från kursen för att skapa en stilren och praktisk webbplats som fungerar på alla upplösningar och i alla webbläsare.

Innehållsförteckning

1 Introduktion / Inledning

Jag ska skapa en hemsida där musikproducenter kan ladda hem filer till eget musikskapande. Målgruppen kommer främst vara yngre män mellan 15-25 från många olika länder runtom i världen. Dess yrke/intresse är musikproducenter med olika erfarenhet och de kommer främst använda datorer när de besöker hemsidan.

1.1 Bakgrund och problemmotivering

Musikproduktion och intresset för musikskapande har ökat med hjälp av nya program som gör det lättare att skapa musik. Allt du behöver för att skapa en professionell låt idag är en dator med en DAW(Digital Audio Workstation). Den nya generationen av musikskapare jobbar i en DAW och laddar oftast ner diverse ljudfiler som de sedan implementerar i sina egna projekt. De flesta använder Youtube för att dela och ladda ner ljudfiler, det finns även andra hemsidor för musikdelning men ingen av dem är optimal och hanterar ofta endast en typ av ljudfil. Jag ska därför skapa en hemsida som riktar in sig på delning av musikfiler där alla projekt- och ljudfiler finns tillgängliga.

De som besöker sidan kommer vara musikproducenter eller folk med mindre erfarenhet som vill lära sig att göra musik/utveckla sitt musikskapande. Många av besökarna kommer framförallt vilja ha projektfiler som innehåller en färdig låt som de kan göra om/remixa eller mallar med färdig struktur som gör det lättare för nybörjare att skapa låtar. Besökaren vill även ha så kallade ”samples” vilket är ljudsnuttar som är urklippta ur en annan låt eller originellt inspelade ljud.

1.2 Övergripande syfte / Högnivåproblemformulering

Besökaren kommer alltså för att ladda ned en eller flera musikprojekt/ljudfiler. De som besöker vet vad de vill ladda ned och de flesta förstår hur allt fungerar. En kort beskrivning räcker, det kommer även finnas en undersida med beskrivningar och FAQ. Om besökaren fortfarande stöter på problem så ska det gå att maila företaget.

Besökarna vill veta vad de laddar ned så det kommer därför finnas en förhandsvisning med hjälp av video eller ljud. Alla filer kommer även ha en kort beskrivning med info om filstorlek och vilka program som behövs för att köra filerna.

Det är oftast samma personer som kommer besöka sidan och därför behövs den uppdateras ofta för att besökaren ska återkomma och kunna hitta något nytt. I framtiden ska även besökaren kunna ladda upp egna filer som andra besökare kan ladda ner. På så sätt hålls sidan bättre uppdaterad.

1.3 Avgränsningar

Hemsidan är gjord för musikproduktion men denna rapport kommer hålla sig inom ämnet webbutveckling och skapandet av webbplatsen.

1.4 Översikt

Kapitel 1 är en introduktion med problemmotivering och en förklaring av syftet med arbetet. ”Problemen” som ska lösas i arbetet och vad som ska skapas.

Kapitel 2 går igenom arbetets teori och ger läsaren den kunskap som behövs för att förstå rapporten. T.ex. alla tekniker, begrepp och standards som använts i arbetet.

Kapitel 3 beskriver vilka metoder jag kommer använda för lösa de olika momenten i uppgiften.

Kapitel 4 beskriver upplägget av arbetet och hur webbplatsen konstruerades.

Kapitel 5 går igenom resultatet av arbetet. Är problemen från introduktionen i kapitel 1 lösta och fungerar hemsidan som den ska? Kapitlet går även in på juridiken kring webbplatsen.

Kapitel 6 är en slutsats av arbetet. Det handlar om hur skapandet av webbplatsen gick och vad som är bra, dåligt eller kan utvecklas med webbplatsen. Jag går även igenom vad jag lärt mig.

1.5 Författarens bidrag

Jag har skapat allt själv förutom ”hamburgermenyn” till mindre enheter. Jag tog CSS-kod och Javascript-kod för att ha en mall som jag sedan modifierade. koden hämtades från W3Schools.com
https://www.w3schools.com/howto/howto_js_topnav_responsive.asp

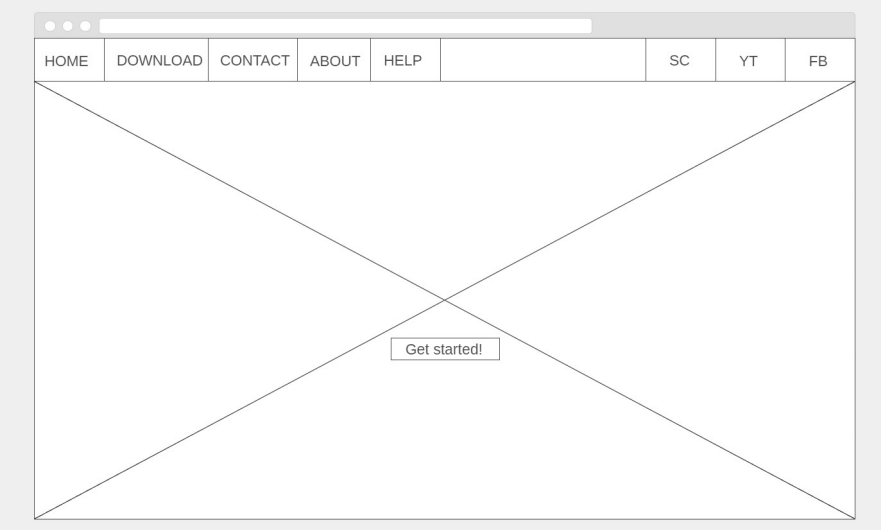
2 Teori / Bakgrundsmaterial

Jag kommer använda mig av HTML, CSS och Javascript för att skapa denna webbplats. Jag använder HTML för att skapa allt innehåll och sedan CSS för att göra en layout och designa innehållet. I HTML-koden skapar jag element för varje sak jag vill lägga till på webbplatsen. I CSS ger jag elementen attribut och bestämmer var de ska placeras. I Javascript ger jag elementen funktioner.

Jag kommer att skapa en responsiv webbplats med liquid layout. Det innebär att det kommer finnas en enstaka webbplats som kommer ändra layout beroende på skärmstorlek. Jag kommer alltså inte skapa en separat mobilversion och pc-version för hemsidan. Jag kommer att använda Media Queries så att webbläsaren ska kunna läsa in en passande layout beroende på besökarens skärmstorlek. Med hjälp av Media Queries sätter jag maxbredd och eller minimumbredd för skärmen med pixlar som måttenhet. I de olika Media Queries skapar jag sedan olika layouter som passar skärmstorlekarna. När besökaren går in på min webbplats så ska webbläsaren läsa av bredden på besökarens skärm och sedan läsa in layouten som ligger inom samma pixelbredd som bestäms av Media Queries.

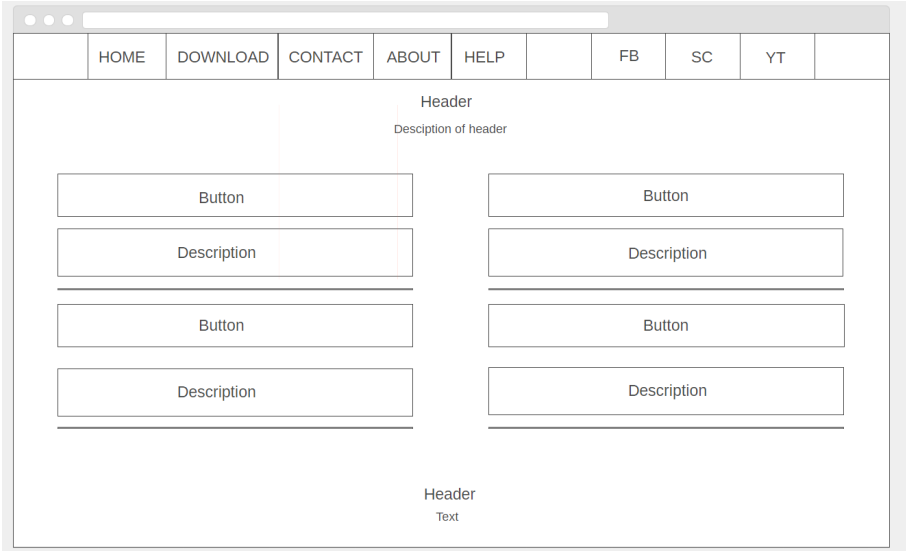
Jag kommer att validera koden för att se om den är korrekt och optimalt skriven. Först laddar jag upp filerna i ett valideringsprogram och sedan kommer programmet köra igenom koden för att leta efter fel eller kod som inte är optimal. Om programmet hittar några fel så går jag in och ändrar i koden och sedan validerar jag den igen tills koden valideras korrekt utan några fel. Jag kommer eventuellt ignorera varningar eftersom de inte alltid är nödvändiga att ändra på, sidan fungerar felfritt även med varningar i valideringen.

2.1 Illustrationer



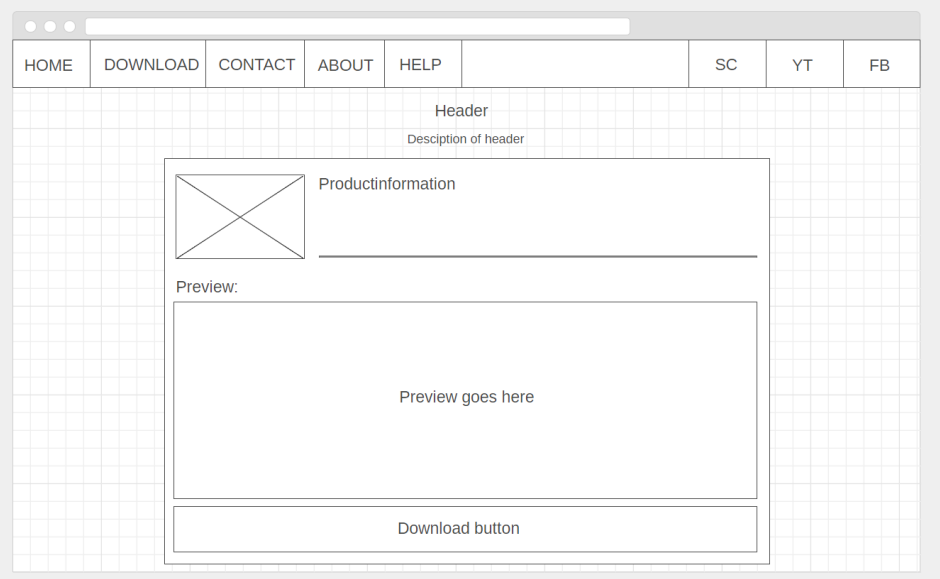
Figur 1:

Wireframe för startsektionen på stora skärmar.



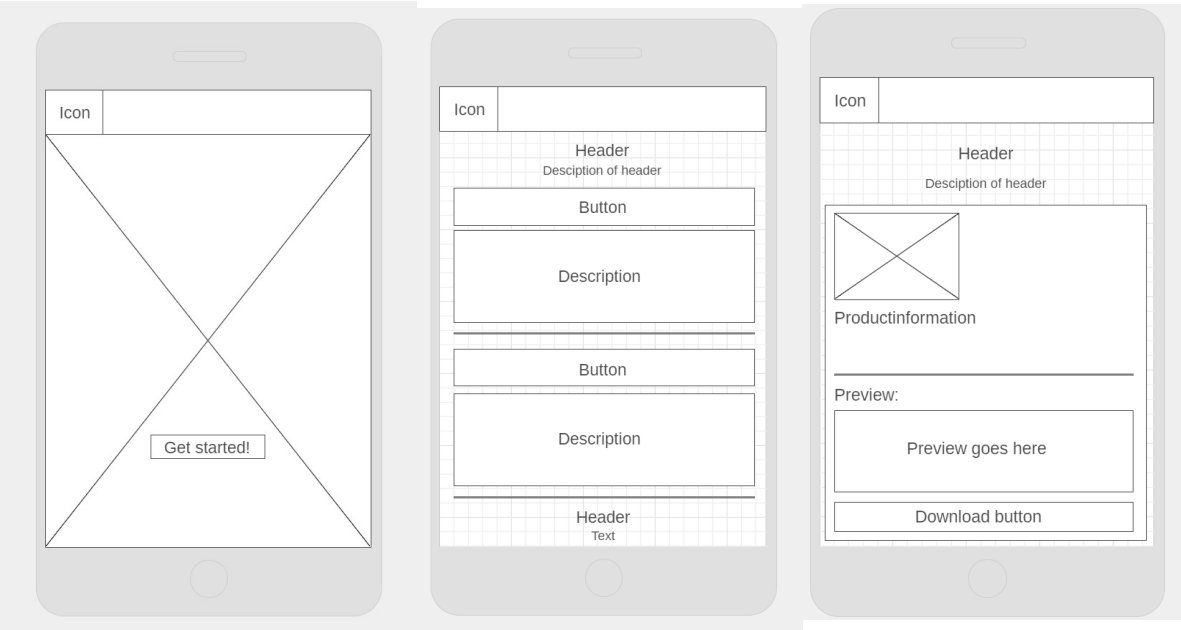
Figur 2:

Wireframe för ”Download”-sektion på stora skärmar.



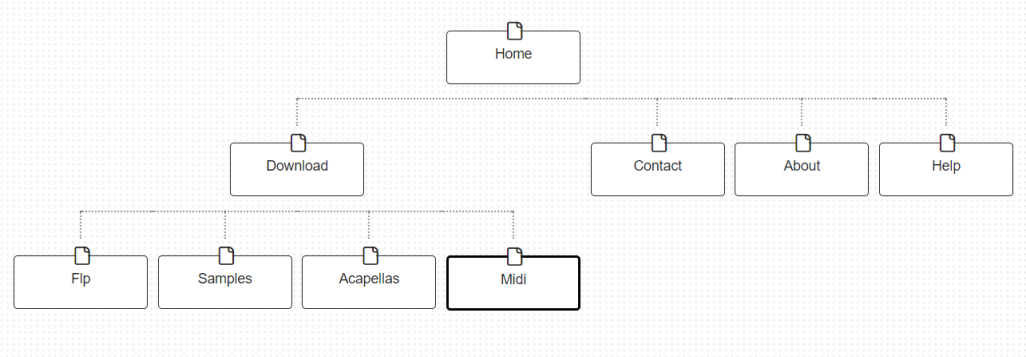
Figur 3:

Wireframe för topplista på startsidan på stora skärmar.



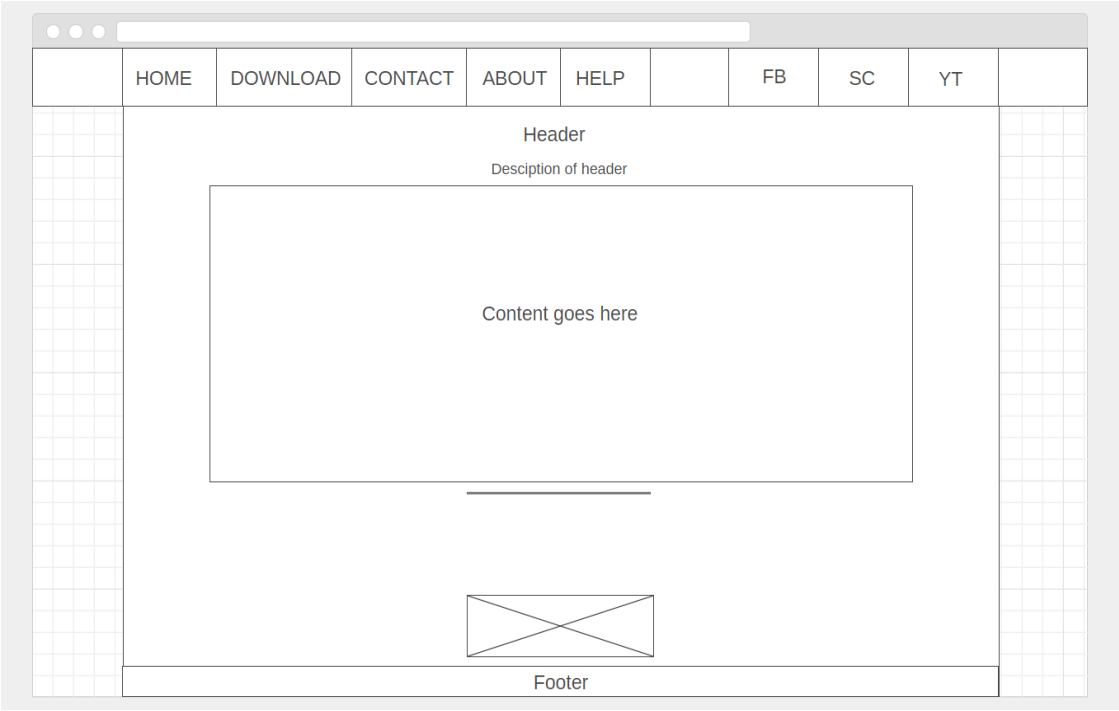
Figur 4:

Wireframe för startsida på mellanstor och liten skärm. Längst till vänster är Wireframe för startsektionen, mitten är för ”Download”-sektionen och till höger är topplistan.



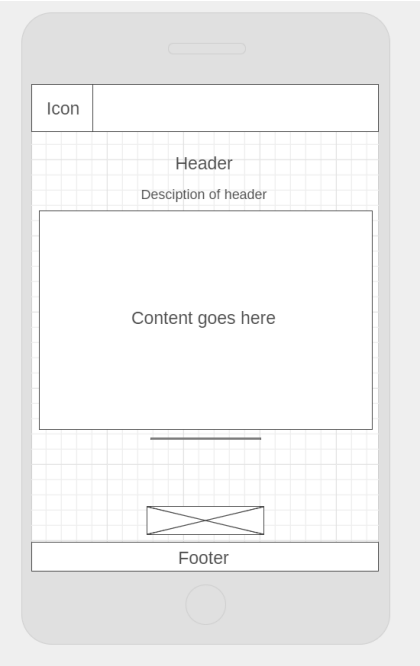
Figur 5:

Sitemap för alla sidor på webbplatsen.



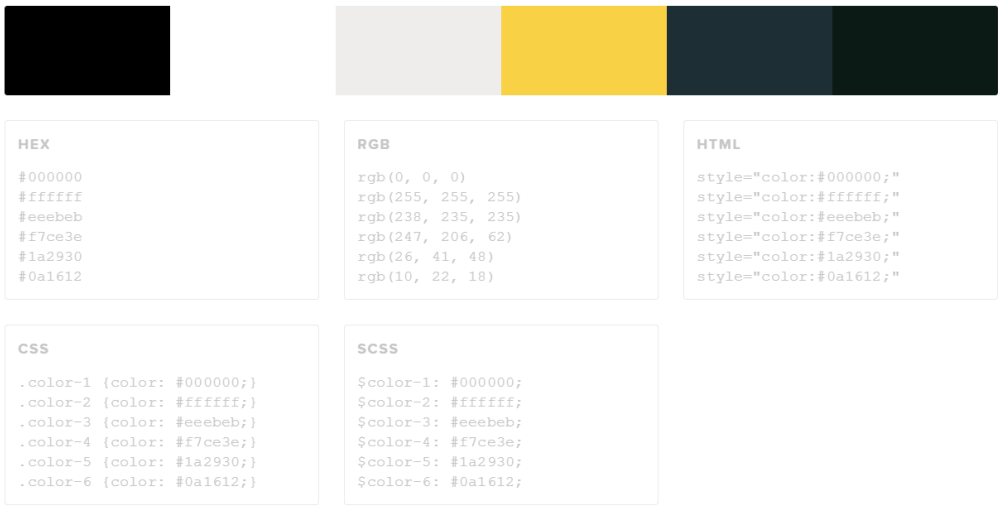
Figur 6:

Wireframe för undersidor på stor skärm.



Figur 7:

Wireframe för undersidor på mellanstor och liten skärm.



Figur 8:

Färgpalett med alla färger på webbplatsen.

3 Metod

Jag kommer att koda i kod editorn Visual Studio Code. All HTML-kod, CSS-kod och Javascript-kod kommer alltså att skapas i Visual Studio Code. Jag kommer sedan använda HTML-validator för att validera koden för att se om den är korrekt skriven.

Jag kommer använda FTP-klienten FileZilla för att ladda upp webbplatsens filer till en server. Webbplatsen filer är: HTML-filerna, CSS-filen, Javascript-filen, bilderna och ljudfilerna.

Jag kommer använda Photoshop för att redigera bilder samt skapa wireframes och en sitemap.

4 Konstruktion / Lösningsalternativ

Jag började med att skissa en startsida för stora skärmar. Tanken jag hade när jag började var att jag ville ha en heltäckande bakgrundsbild med namnet på hemsidan och en slogan samt en ”Kom igång”-knapp som skickar besökaren nedåt i hemsidan till första sektionen. Jag ville skapa en meny för alla ”huvud”-undersidor och en submeny för alla undersidor som ligger under samma kategori i en av ”huvud”-undersidorna. I menyn skulle det även finnas länkar till sociala medier och menyn skulle vara positionerad högst upp på hemsidan och följa med när besökaren skrollar. (Se Figur 1).

På huvudsidan tänkte jag att det även skulle finnas någon typ av innehåll så att besökaren får en liten introduktion över vad hemsidan erbjuder och hur de ska navigera sig för att hitta det den söker. Jag skapade då en sektion för alla undersidor för ”Download” så att besökaren lätt kan navigera sig till det den vill åt.(se figur 2). Jag skapade även en till sektion med en topplista på mest nedladdade filer så att besökaren får några exempel på vad hemsidan erbjuder. Jag visar då de bästa produkterna för att skapa intresse så att besökaren vill utforska mer. (Se figur 3).

När skissen för stora skärmar var klar så började jag skissa på layouten för mellanstora skärmar(surfplatta) och små skärmar(mobiler). Jag tänkte att layouten kunde vara ungefär likadan fast bredden på elementen och placeringen av texten på vissa ställen skulle vara annorlunda. Från början hade jag tänkt att endast mobillayouten skulle ha en ”hamburger”-meny och att surfplattelayouten skulle ha samma meny som PC-layouten. Men jag stötte på problem i kodandet och PC-menyn blev för stor på surfplattan så alla menylänkar låg på fel ställen vilket skapade en icke tillfredsställande visuell upplevelse. Eftersom jag inte hade så gott om tid så bestämde jag mig för att använda en hamburgermeny även för surfplattelayouten vilket inte är optimalt men det fungerar och ser helt okej ut. (Se figur 4).

Efter det bestämde jag vilka undersidor jag skulle ha och hur de skulle se ut. Jag skapade först en sitemap med alla sidor på webbplatsen. (Se figur 5). Sedan gjorde jag en wireframe med lite annorlunda design än huvudsidan. Designen på innehållet skulle vara lika som på startsidan men containern där allt innehåll ligger skulle vara smalare och menylänkarna skulle starta och sluta på samma bredd som containern. (Se figur 6). Jag skapade även en wireframe för mellanstora och små skärmar som i princip har samma design som startsidan. (Se figur 7).

De exakta färgerna på webbplatsen var inte särskilt genomtänkta från början men jag visste ungefär hur jag ville använda färger. Jag ville ha en mörk meny med ljus text. Jag ville att innehållet skulle ha mörk text på en ljus bakgrund för att skapa hög kontrast. Jag ville inte ha en vit bakgrund eftersom många av

besökarna sitter i ett mörklagt rum och det blir ansträngande för ögonen. Jag ville även ha en unik färg som står ut och ger hemsidan något som besökaren känner igen. Facebook har blått och Youtube har rött t.ex. Den unika färgen skulle finnas med i loggan och i många HTML-element. Jag improviserade färgerna under kodandet men jag skapade en färgpalett i efterhand. (Se figur 8).

När jag var klar med skissandet så började jag att koda startsidan för stora skärmar. Jag började med att göra en ”container” där allt innehåll skulle ligga. Sedan skapade jag en header där menyn skulle ligga. Den skulle följa med när jag skrollar och det gjorde jag med ”position: fixed”. I headern skapade jag en simpel ul-meny med en bredd på 100% så att den täcker hela skärmbredden. Jag ville att li-elementen skulle ligga vågrätt och det gjorde jag med ”display: inline-table”. Li-elementen för undersidorna skulle ligga till vänster och det bestämde jag med ”float: left”. Li-elementen för sociala medier skulle ligga längst till höger i menyn och det bestämde jag med ”float: right” som jag skrev som intern CSS i elementen. Sedan gjorde jag en submeny för alla sidor som ligger under ”Download”. Jag skapade en simpel drop-down-meny som fäller ut sig själv när besökaren placerar muspekaren över li-elementet ”Download”. När grunden för menyn var klar så designade jag den lite mer och gjorde den snyggare.

När menyn var helt färdig så började jag med startsektionen där det skulle vara en heltäckande bild och en knapp som skickar besökaren vidare nedåt i sidan till nästa sektion. Jag laddade ned en bakgrundsbild med upplösningen 1920x1080. Sedan beskar jag den så att den skulle passa för surfplatta och mobil så att bilden inte blev hoptryckt och liten. Jag lade bilderna i CSS-koden som bakgrundsbilder med olika id för de tre olika upplösningarna. Sedan skapade jag 3 div:ar och lade in alla olika bild-id i varsin div. I media query bestämmer jag sen vilken av de 3 div:arna som ska visas, vilka som ska döljas och höjden/bredden på bilden men det kommer vi till senare i rapporten. När bilden är på plats så skapar jag en knapp som tar besökaren nedåt i sidan till nästa sektion. Jag skapar ett a-element som skickar besökaren till ett annat a-element. Det andra a-elementet har ett id som det första a-elementet hänvisar till och det placeras ovanför sektionen där jag vill att besökaren ska hamna. För att få det första a-elementet att se ut som en knapp så ger jag det en klass där jag sätter en ”border”, alltså en ram runt länktexten. För att lägga a-elementet på eller framför bilden så använder jag ”position: absolute” vilket lägger den utanför normal flow så att den kan placeras var som helst på skärmen.

När startsektionen var klar så började jag koda innehållet på startsidan. Först skulle det vara en downloadsektion för alla undersidor under ”Download”. Då gjorde jag helt enkelt 4 articles, en för varje undersida. Jag skapade 2 klasser, en för 2 articles och en för de 2 andra. Varje article innehåller ett a-element som skickar besökaren till en undersida. Dessa a-element har också en klass som får de att se ut som en knapp, likt den i startsektionen. Under a-elementen finns sedan en kort beskrivning av de olika undersidorna och en ”border-bottom” som visar var varje article slutar.

När downloadsektionen var klar så började jag med sektionen för topplistan. Topplistan har även samma design som kommer användas för undersidorna under ”Download”. Jag ville ha en ram runt varje produkt så jag skapade en klass med en ”border”. Inuti ramen ligger allt innehåll och innehållet består av: en bild, en rubrik, en underrubrik med filtyp/filstorlek, en kort text med beskrivning, en förhandsvisning och en nedladdningsknapp. På stora skärmar ville jag att rubrikerna och texten skulle ligga till höger om bilden så jag satte ”float: left” på bilden. På små skärmar skulle texten ligga nedanför bilden så jag satte ”float: none” i media queryn för små skärmar men återigen så kommer vi till det senare i rapporten. Till förhandsvisningen använde jag både <audio>-taggen och <iframe>-taggen. I <audio>-taggen lägger jag till en <source>-tagg med en mp3-fil eller ogg-fil. <iframe>-taggen använder jag till att visa inbäddade länkar från externa webbplatser. De länkarna jag bäddat in kommer från Youtube och Soundcloud. Nedladdningsknappen är gjord på samma sätt som knapparna i downloadsektionen och skickar besökaren till en extern webbplats där filerna kan laddas ned.

Sist skapade jag en simpel footer med namn och email-adress med en mailto-länk.

När jag var klar med startsidan så skapade jag media queries i CSS för stora skärmar(PC), mellanstora skärmar(surfplattor) och små skärmar(mobiler). För små skärmar satte jag en maxbredd på 500 pixlar, för mellanstora skärmar satte jag en maxbredd på 800 pixlar och en minimumbredd på 500 pixlar. På stora skärmar satte jag en minimumbredd på 801 pixlar. I kodandet fick jag återigen problem med menyn och att menylänkarna var för stora så att de lade sig utanför själva menyn. Problemet uppstod i en övergång mellan stora och mellanstora skärmar så jag var tvungen att skapa en till media query för den övergången. Den låg mellan pixelbredden 801 och 984. Då var jag även tvungen att ändra på minimumbredden för stora skärmar till 985 pixlar. Sist men inte minst gjorde jag även media query för 4k-upplösning där jag satte minimumbredden till 2560px.

När jag var klar med alla media queries så började jag koda layouten för surfplatta och mobil. I media query satte jag en maxbredd på containern för de olika skärmarna. Maxbredden på containern ska vara samma som maxbredden på media queryn för att innehållet hålla sig inom storleken på skärmen. Annars måste man scrolla till höger och vänster för att se allt innehåll och det blir varken stilrent eller praktiskt. Jag måste göra likadant med bilden i startsektionen så att den inte blir bredare än skärmens bredd. Jag måste även sätta en minimumhöjd på bilden i startsektionen så att den alltid täcker en skärmhöjd oavsett om man förstorar eller förminskar skärmen. Då kommer bilden bli mindre på bredden men inte på höjden vilket tyvärr gör så att den dras ihop lite. Det är inte jättesnyggt vid alla övergångar men det funkar eftersom jag har 3 olika bilder med rätt upplösning för rätt skärmstorlek. Jag var tvungen att göra på detta sätt för att knappen i startsektionen skulle ligga på rätt ställe. Annars hade bilden blivit mindre på höjden vilket gör att sektionen under kommer fram och då hade knappen hamnat framför den sektionen istället. En

till sak jag måste göra med bilderna är att visa rätt bild för rätt skärm. För att dölja de två bilderna som inte har rätt storlek använder jag ”display: none” i media query för deras olika id.

I downloadsektionen kommer alla articles automatiskt lägga sig till under varandra med hjälp av ”float: left” som skrevs tidigare i koden. Jag sätter höjden på alla articles till 100% så att de tar slut där innehållet tar slut så att de hamnar direkt under varandra utan något onödigt stort avstånd.

I sektionen för topplistan ökar jag bredden på articles så att de täcker större delen av skärmen. Jag lägger även texten under bilderna med hjälp av ”float: none” i klassen för bilderna.

När layouten för innehållet var klart så var det dags att skapa en meny. Jag bestämde mig för en så kallad ”hamburgermeny”. Jag skapade en div med a-element som menylänkar. Första menylänken som är hamburgerloggan har en egen klass och endast den klassen visas i menyn med hjälp av ”display: block”. De resterande a-elementen har en annan klass där de ej visas, med hjälp av ”display: none”. Jag skapar sedan ytterligare en klass där alla menylänkar visas med hjälp av ”display: block”. Vid klick på hamburgerloggan körs en funktion i Javascript som lägger till klassen där alla menylänkar visas och på så sätt skrivs de resterande menylänkarna ut. Vid klick på hamburgerloggan igen så fälls menylänkarna in och döljs. Javascriptfunktionen växlar mellan att lägga till och ta bort klassen som visar alla menylänkar. Jag visste inte hur jag skulle bära mig åt för att skapa denna funktion så jag tog hjälp av en annan hemsida med en genomgång om hamburgermenyer[1]. Jag kopierade koden och och modifierade den så att den passar min hemsida. Sedan gick jag igenom koden bit för bit och försökte förstå vad den gjorde tills jag till slut förstod den helt.

Undersidornas layout är samma som på startsidan för mobil och surfplatta men för PC och 4k-upplösning bestämde jag mig för en lite annorlunda layout (Se Figur 6). Jag skapade en ny container och minskade bredden från 100% till 80% vilket jag tyckte såg mycket bättre ut. Jag skapade även en ny meny där jag flyttade första li-elementet och sista li-elementet 10% inåt mot mitten så att de låg i bredd med den nya containern. För att välja det första li-elementet använde jag ”li:first child” och för att flytta det använde jag ”margin-left: 10%”. För att välja det sista li-elementet som nu råkar vara det tredje sista i HTML-koden så jag använde jag koden ”li:nth-last-child(3)”[2]. Jag väljer alltså det tredje sista li-elementet och för att flytta det använde jag ”margin-right: 10%”.

5 Resultat

Resultatet av arbetet blev någorlunda bra med tanke på hur lite tid det fanns och hur mycket som ska in på hemsidan för att den ska ha någon riktigt funktion. Jag har gjort layouten och skapat allt innehåll som jag hade tänkt. Det enda som saknas om jag utgår ifrån inledningen är mängder av filer som ska kunna gå att laddas ned. Just nu har jag bara skapat exempel på hur varje fil och förhandsvisning kommer se ut. För att fylla på sidan så måste jag leta rätt på musikproducenter som delar sina filer och sedan fråga om lov att visa de på min hemsida. Just nu har jag inte frågat någon om lov för något av det jag använder på hemsidan eftersom jag inte har något som är någon annans. Bilderna jag använder är från en hemsida som heter Unsplash där bilderna får användas på detta sätt: ”Unsplash grants you an irrevocable, nonexclusive copyright license to download, copy, modify, distribute, perform, and use photos from Unsplash for free, including for commercial purposes, without permission from or attributing the photographer or Unsplash, but this license does not include the right to compile photos from Unsplash to replicate a similar or competing service.”[3].

Om jag laddar upp någon fil eller en förhandsvisning i mp3-format utan att fråga personen om lov så faller det under upphovsrättslagen. Även om jag frågat personen om lov så kan den personen använt en fil i sitt projekt eller liknande som inte personen äger rättigheterna till och då kan jag inte ge ladda upp de filerna på min hemsida eftersom de faller under upphovsrättslagen. Just nu är det inga filer som jag eller den jag bett om lov inte äger rättigheterna till men vid alla filuppladdningarna måste jag se till så att inga filer faller under upphovsrättslagen.

Just nu innehåller webbplatsen korrekt information förutom i alla nedladdningsexempel. Alla exempel är bara påhittade. Filstorleken, filtypen och titeln stämmer inte på alla ställen och nedladdningsknappen har ingen hänvisning på alla ställen. Topplistan innehåller inte heller de flest nedladdade filerna. Det är bara en exempellista.

För att testa så att hemsidan fungerar som den ska så har jag testat den på alla upplösningar i ”inspektera-läget” i Chrome. Jag testade allt från 4k-upplösning med 2560 i pixelbredd till Mobile Small med 320 i pixelbredd. Jag har testat den i de fem största webbläsarna: Google Chrome, Internet Explorer, Microsoft Edge, Mozilla Firefox och Opera. Jag har även testat den på mobilen(iPhone 6). Jag hade velat testa den på en iPad men hade inte tillgång till en.

6 Slutsatser / Analys / Diskussion

Det största problemet jag stötte på var menyn för små skärmar. Det blev svårt när man skulle blanda in Javascript och jag var tvungen att ta hjälp från en annan hemsida för att göra den. Även fast jag jag lyckades göra den så blev den inte optimal eftersom jag inte fick med alla undersidor i menyn. Jag försökte göra en drop-down-meny för undersidan ”Download” som i menyn för stora skärmar men jag lyckades inte med det. Nu måste mobilanvändarna först klicka på menyn, sedan på ”Download” som går till den sidan och sen välja en undersida för ”Download”. Min tanke var att när besökaren klickar på ”Download” i menyn så fälls det ut 4 undersidor under ”Download”-knappen och när man klickar på ”Download” igen så fälls de in. Jag skulle kunna göra en simpel select-meny också men jag tycker inte de ser bra ut som huvudmenyer.

Ett till problem jag stötte på som jag inte har lyckats lösa än är att förklaringsbilden(explanation image) inte laddas in på den publicerade webbplatsen. Den laddas in när jag testkör webbplatsen lokalt men inte annars.

En sak jag kunde gjort med menyn på stora skärmen är att ha en logga på hem-knappen istället för text. Jag skulle även kunna lägga till en logga ovanför meny som sedan fälls in vid skroll nedåt. Jag skulle även kunna ha ett bildspel istället för en bild på startsidan där jag visar upp vad hemsidan erbjuder i bilder.

I topplistan/nedladdningslistan ville jag ha en funktion där varje enskild förhandsvisning endast laddas in vid ett klick på en knapp eller text[4]. Nu laddas alla youtubeklipp och soundcloudklipp in samtidigt och det tar lång tid för sidan att laddas in om det är väldigt många. Jag ville också ha ett kontaktformulär som skickar det som skrivs till min mail men som jag förstod det så behövdes det PHP för att kunna åstadkomma det vilket jag inte har någon erfarenhet av.

Jag skulle kunna göra tydligare sektioner så att man ser var de startar och slutar. Textfärgen i sektionerna är lite tråkiga eftersom det blir mycket grått på grått. Jag skulle använt en annan färg på huvudrubrikerna och eventuellt underrubrikerna. En till sak som jag inte blev helt nöjd med är footern. Jag skulle ha gjort den större och lagt in mer information i den.

För att utveckla hemsidan ännu mer så har jag främst tänkt att det ska finnas en funktion för uppladdning där man kan skapa en profil och ladda upp egna filer. Det är även viktigt att lägga till en sökfunktion där man kan söka mer specifikt efter vad man vill ha. Topplistan ska även ha en funktion där den filen med flest nedladdningar hamnar på första plats. För att åstadkomma alla dessa saker så måste jag utveckla min kunskap i mer än bara HTML och CSS. Det är även en sak jag lärt mig av denna uppgift, att det krävs mer än HTML och CSS för att

skapa en komplett och väl fungerande hemsida. Beroende på vilken typ av hemsida man gör såklart.

Utöver det så har jag lärt mig massvis av saker i både HTML och CSS. Jag har främst lärt mig att utveckla för mobiler/surfplattor och för olika skärmupplösningar. Jag har även lärt mig lite bildredigering med wireframes, loggor och bakgrundsbilder etc.

Källförteckning

[1] W3Schools, ”How to – Responsive Top Navigation
https://www.w3schools.com/howto/howto_js_topnav_responsive.asp
Hämtad 2017-11-05

[2] Css-tricks, ”:nth-last-child”,
https://css-tricks.com/almanac/selectors/n/nth-last-child/
Publicerad 2013-04-03. Hämtad 2017-11-05.

[3] Unsplash, ”Terms & conditions”,
https://unsplash.com/terms
Hämtad 2017-11-05

[4] Envato tuts, ”How To Lazy Load Embedded Youtube Videos”,
https://webdesign.tutsplus.com/tutorials/how-to-lazy-load-embedded-youtube-videos--cms-26743
Publicerad 2016-06-30. Hämtad 2017-11-05

Bilaga A: Dokumentation av egenutvecklad programkod

Exempel på underrubrik