Jag bestämde mig för att använda mig av mallen, för att jag tycker det är en rolig utmaning att följa någon annans design. Det första jag gjorde var att dela in allt jag såg i 3 sektioner (header, main och footer), och sedan delade jag in allt inuti dessa sektioner, till ännu mindre sektioner (header-nav, header-hero o.s.v.).

Här stötte jag på mitt första problem då jag ville namnge de 3 största sektionerna till ”<section>” och inte ”<div>”, men jag fick en varning vid kontroll av html (w3, n.d.), för att en <section> alltid ska ha en heading (Mozilla, n.d.), vilket jag inte hade planerat att ha i min “main” där jag bara ville ha <p> element som skötte text-innehållet. Jag ändrade <section> till <div> för att lösa problemet, det verkar enligt Mozilla inte göra någon skillnad.

Tanken med en namngivandet av “hero” <div> var att ha bakgrundsbilden där, men jag valde senare att istället placera bakgrundsbilden i “header” för att jag tyckte det var snyggt om bakgrundsbilden även syntes under “nav”. Jag gav därför “nav” en halv-transparent bakgrundsfärg så att “logo” och “links” blev mer lättlästa med den underliggande bakgrundsbilden som annars hade påverkat tydligheten.

Jag gav “main” en “class” med container i namnet, för i denna ”container” skulle de tre “korten” vara. Dessa ”korten” fick också klass-namn med “container” i namnet för att de isin tur innehöll olika element såsom text och bild. “Korten” fick även olika id, eftersom jag såg att det var tänkt att de skulle bete sig annorlunda gentemot varandravid ändrad skärmstorlek. Jag ville därför kunna kontrollera dem individuellt. Med “margin-left/right” kunde jag sedan flytta bild-elementet vid ändring av skärmstorlek.

Jag valde att ha en “break-point” för ändrad skärm-storlek vid “max-width: 700px”, dels för att vi använde oss av detta i en tidigare uppgift om jag minns rätt, och för att det faktiskt passade bra för denna sidan. Det finns, (Mozilla, n.d.), för många olika skärmar för att göra en perfekt anpassning, så det bästa att göra är att anpassa bryt-punkten till där sidan börjar att ge vika, och i det här fallet så tittade jag på när länkarnas <div> närmade sig loggans <div>.

Jag gav en google-font till länkarna. Det var den roligaste men kanske inte den bästa lösningen. Jag kontrollerade att menyn inte förändrade sidans “break-point” för mycket om fonten inte kunde nås p.g.a. ingen internetåtkomst.

Bilderna är (Pixabay, n.d.) gratis och royalty-fria, och temat var bara något jag valde för att göra det lite roligare. Det viktiga var att ha olika bilder så att man kan se hur “korten” omplacerar sig vid olika skärmstorlekar.

Till hela sidan gav jag, i CSS-filen längst upp, en “margin” och “padding” med värdet 0, för att jag längre ner ska ha absolut kontroll över detta, Jag började med detta för längesedan för att jag hade problem med vissa element som hade antingen margin eller padding inbyggt som “default”, och att jag nollställde detta genom att ge allt ett värde av 0, men jag kan inte minnas vad/vilka, och nu när jag söker efter svar, så hittar jag inget, så det kanske är onödigt.

“Body” fick “width” 100% för att allt innehåll ska sträcka sig horisontellt, men däremot så fick “height” “auto” för att innehållet ska kunna vara flexibelt lodrätt då mängden innehåll förändras vid ändrad skärmstorlek.

“Nav” fick en flex-box för att det är (Mozilla, n.a.) ett smidigt sätt att hantera de “items” (logo, meny), som är i den. “Justify-content” användes för spridning längs x-axeln, och “space-between” för att föra loggan och menyn till respektive sidor. Dessa justerade jag sedan ytterligare med “margin”. “Nav” fick “rgba” för att kontrollera opacity så att bakgrundsbilden fortfarande syns igenom.

Länk-menyn justerades med “margin”i sin <div>, och även för att öka mellanrummet mellan länkarna. “Text-decoration” blev “none” för att ta bort “underline”.

“Header” fick en “background-image” och en “background-position: center center;” för att hålla den centrerad. Den fick även en ordentlig höjd på 400px. Eftersom jag inte ville ha samma höjd på headern till processanalys-sidan, så gav jag headern i den html-koden ett eget id, så att jag härnäst i CSS-dokumentet kunde ge den en höjd på 75px vilket var lagom för att “nav” skulle få plats. Bakgrundsbilden kunde här få vara kvar eftersom det inte är något tydligt motiv som blir konstigt visuellt om man inte ser mer av bilden. När den är kvar bidrar den med färger som knyter ihop sidan med huvudsidan där hela bilden syns.

Likadant gjorde jag med andra delar av processanalys-sidan, att jag gav egna id till olika delar av sidan så att jag t.ex. kunde ge en container i dess “main” en speciell margin m.m. som jag inte ville ge till index.html.

Loggan blev ett h1-element, med backgrundsfärg liknande den i mallen, ändrad font-storlek pch “text-align: center” för att centrera texten så den inte var vänster-centrerad.

“Hero-texten” placerades med hjälp av “margin”, och text-align användes för att göra den lokalt centrerad.

“p” fick en “font-family” med vanliga fonter.

“.cardContainer” flyttades uppåt med margin, så att kortet med allt innehåll hamnar ovanpå bakgrundsbilden, som enligt mallen.

“.cardsContainer” som innehåller de 3 “korten, gavs “justify-content: space-evenly” för att de tre korten skulle fördela sig längs x-axeln med luft på båda sidor av alla kort.

Bilderna till “korten” gavs “border-radius: 50%” för att göra dem runda, och de fick en tjock vit “border”.

“.footer” fick också “space-evenly” flex-box för att sprida ut dem med luft på båda sidorna.

Med “@media” ändras många element, bland annat hero-texten som bland annat centreras på skärmen i sidled. Headerns höjd minskas då den annars tar upp för stor plats på en mindre skärm. “Nav” fick “display: block” för att logo och meny ska kunna placera sig vertikalt, och likadant med”.footer”. “h1“ fick även “display:inline-block” som centrerar den och även respekterar width och height. “.cardsContainer” fick “flex-direction:column;” för att kunna lägga sig som en kolumn, och här fick även ett kort med rätt id, “order: -1”, så att det kortet får den första platsen i kolumn-flex containern.

Vid “@media print” så får header och footer “display:none”, då den som skriver ut fömodligen vill skriva ut det som finns i “main”.

W3schools (n.d.). Nu Html Checker.  Tillgänglig: https://validator.w3.org/nu/ [2023-03-20].

Pixabay (n.d.). Stunning free images & royalty free stock.  Tillgänglig: https://pixabay.com/ [2023-03-20].

Mozilla (n.d.). <section>: The Generic Section element.  Tillgänglig: https://developer.mozilla.org/en-US/docs/Web/HTML/Element/section [2023-03-20].

Mozilla (n.d.). <section>: The Generic Section element.  Tillgänglig: https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS\_layout/Media\_queries [2023-03-20].

Mozilla (n.d.). <section>: The Generic Section element.  Tillgänglig: https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS\_layout/Flexbox [2023-03-20].