**Skisser och Reflektion av:**

**Kunduppgift – en inloggningssida**

Jag började med att först läsa igenom uppgiften och skriva ner krav och önskemål i den på ett papper så att jag skulle få lite bättre överblick på uppgiften i huvudet. Därefter så tittade jag på exempelbilderna i uppgiften och började skissa en grovskiss över de tre fallen som hemsidan ska kunna vara i:A notebook with writing on it

Description automatically generated

*Skissblad 1: Fall 1 med Logga in diven och Fall 2 med välkomstdiv och medlemskapsdiv*

A close-up of a paper

Description automatically generated

*Skissblad 2: Fall 3 med ogiltiga inloggningsuppgifter diven*

**Tankar och utförande av min plan:**

**index.html:** I alla tre fallen så syns ”Välkommen till Fantasia Brukshundsklubb”, samt ”Förhållningsregler” med en lorem ipsum text. Dessa två är varsin div med sina element h1 och p i ”Välkommen…….” och h2 och p i ”Förhållningsregler”. Den första gav jag ”class name-header” och den andra gav jag class ”rules” och id ”rules” som används i style.css filen för att styla dem. Varför jag valt att lägga dessa divs och element i html:en är för att de kommer ej att bytas ut under ”körningens gång” utan vara kvar hela tiden på sidan. Så därför tänkte jag att då ka de vara i html:en. Jag länkade så klart också till style.css och scriptet till script.js.

**style.css:** I CSS-filen style.css så använder jag mig av Flexbox för att placera divar i column och space-between för att flytta isär dem på ett trevligt sätt (tycker jag). Jag sätter dem i centrum i sidled också med hjälp av align-items. Detta gjorde jag i body-elementet då det ej ska vara annat på denna sida. Jag satte även min bild (på en vuxen hund, med pinne i munnen och en valp som springer tillsammans) som bakgrundsbild över hela sidan i bodyn. Jag valde cover som background-size så att bilden ska täcka hela sidan.

För utseendet på de olika divsen och dess innehåll har jag:

* **elementtagg**(h1)
* **klasser**(name-header, rules, logInDiv, logInDivHide, welcomeDiv, welcomeDivHide, memberDiv, memberDivHide, invalidLogDiv, invalidLogDivHide och logInDiv-item)
* **id:n**(h2-rules, p-rules, p-header och red)

De divar som har mer än ett element inom sig använder flex-direction column såa att de placerar sig ovanför varandra på ett bra sätt. De som bara har ett element fixar jag placeringen med margin och padding när det behövdes för att inte bli för tight till närmsta element eller till kanten.

De klasserna som heter ”Hide” i slutet är till för att sätta ”display” till ”none” så att den diven blir gömd när den inte ska visas.

Id:t som heter ”red” är till för att göra så att färgen på den felaktiga inloggningstexten blir röd så att det blir tydligare för de som ser färgen. För de som är färgblinda så märker de att det blir en ny ruta med texten i som talar om att det är fel.

Jag valde vit bakgrundsfärg i divarna för att det ska bli hög kontrast mellan texten och bakgrunden i divarna så inte bakgrundsbilden gör texten mer svårläslig.

**script.js:** I skriptfilen ligger ju den större delen i denna uppgiften. Här har jag delat in koden med hjälp av /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Titel på del i koden\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/ i början av varje ”kodavsnitt” och med /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/ i slutet av varje ”kodavsnitt” för att lättare kunna hitta snabbt i koden och så att jag eller någon annan lättare ska kunna tolka och läsa koden om man ska jobba vidare på den i framtiden. Jag har även kommenterat på andra ställen i koden.

1. **Inloggningsuppgifter:** Först deklarerar jag de enligt uppgiften båda variablerna ”namn” och ”lösenord” som konstanta variabler. Jag funderade här på om jag istället skulle använda ”name” och ”password” för att passa övrig kod, men jag behöll dess två på svenska då det stod så i uppgiften.
2. **Variabel för sista diven……:** Jag gör en konstantvariabel till rulesDiv så att jag lätt når den sista diven när jag senare ska placera de som skapas på rätt plats i dokumentet.
3. **Inloggningsdiven:** Sedan kommer ett kodavsnitt där jag skapar logInDiv:en och dess element som jag sätter dit med appendChild() till logInDiv:en. Jag sätter också attribut till både div:en och elementen som t.ex. klasser för sstyle.css filens skull, placeholder för att ha text i inputrutan när den är ”tom” så användaren vet vad den ska fylla i. Så en div med en h2, två inputrutor och en knapp skapas här och läggs till div:en kort och gott.
4. **Välkomstdiven:** Precis som i Inloggningsdiv-koden men här är det ju bara en paragraf som ska in i denna div som skapas.
5. **Medlemskapsdiven:** Samma som för Inloggningsdiv-koden men här är det en h2, en p och en knapp som skapas för att säga hej till användaren med namn, hantera inlogg och kunna logga ut.
6. **Ogiltig inloggningsdiv:** Precis som med Välkomstdiven fast en annan text inuti denna så klart som meddelar för användaren att uppgifterna vid inloggningen är felaktiga.
7. **Eventlyssnare för inloggningsknappen:** Här läggs det till en eventlyssnare som väntar på om användaren trycker på inloggningsknappen. Om det görs sås körs ett villkor med if och else if. Ifen har även en if inuti sig. Den yttersta if:en kollar om användarnamnet är rätt jämtemot konstanten ”namn” och om lösenordet är rätt jämtemot konstanten ”lösenord”. Om det är så så körs funktionen logInDisplay() som gömmer inloggningssidan(diven) och tar fram välkomstdiven och medlemskapsdiven. När funktionen har körts så sparar vi i localStorage att vi är inloggad med nyckeln ”isLoggedIn” med värdet som användarens användarnamn är ifrån inputen(kan ju vara andra användare i framtiden). Innan denna funktionen är slut så kör vi en if som kollar om vi har visat ogiltig inloggningsdiv för då vill vi ju dölja den så den inte är kvar ifall vi loggar ut senare. I else if om namn och lösenord (eller någon av dem) är fel så kollar vi om klassen ”invalidLogDivHide” finns på invalidLogDiv:en. Om den gör det så tar vi bort den för att visa felmeddelandet för användaren.
8. **Eventlyssnare för utloggningsknappen:** Samma förklaring almänt som förra knappen men när användaren trycker på denna knappen så göms välkomstdiven och medlemsdiven. Inloggningsdiven blir synlig igen, vi rensar localStorage och tömmer de 2 inputrutorna i inloggningsdiven så att det ser ut som från början igen.
9. **Function logInDisplay:** Redan nämnt vad denna gör ovan.
10. **If:en i slutet av koden:** Denna if kollar om det är så att vi är inloggade eller inte. Är vi det så körs funktionen logInDisplay annars händer inget övrigt så vi kommer till inloggningssidan som vanligt.

**Lärdomar och vad kan förbättras som jag vet om**

Med de krav som är satta i uppgiften så tror jag att jag har fått med allt som behövs så som jag har tolkat det. Jag har lärt mig en massa spännande i DOM och i Javascript. Roligt att kunna förändra sidan under ”körning”. Saker som kan optimeras kan nog vara att ännu mer se över CSS-filen för jag tror att några klasser kanske kan ”bakas ihop” som t.ex. att gömma kanske skulle kunna vara samma klass för alla. Om jag hinner och vill prioritera så kanske jag ska fixa ännu mer med t.ex. färger och placering av rutor och knappar för at ännu mer förtydliga saker. Det jag är nöjde med i min script.js fil är att jag delade in den med hjälp av /\*\*\*\*\*/ så att det lättare går att hitta vilken del av koden som hör till vad. Detta tror jag kommer bli smidigare om man går tillbaka vid ett senare tillfälle. Jag är nöjd med arbetet i helhet. Antingen förbättrar jag den mer eller kommer jag kolla på Udemy filmerna mellan CSS och Javascript avsnitten.