|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Visualisering  V3.1 | | | |  |
| **Emne:** | PHP | **Opgave:** | Login funktionalitet |
| **Formål:** | * Jeg kan anvende formularer i PHP | | |
| * Jeg kan anvende phps superglobal arrays $\_POST og | | |
| $\_SESSION til at lade en bruger være logget ind | | |

**Opgave**

Du skal i denne opgave sørge for at brugeren bliver logget ind og fortsætter med at være logget ind, indtil de logger sig ud. Dette betyder, at dit login menupunkt skal ændres til et logud menupunkt i sidemenuen, når brugerne er logget ind. I topmenuen skal login ændres til Min konto. Når brugeren logger ud, skal menupunkterne ændres tilbage tilsvarende og du skal sørge for at afbryde sessionen og slette alle data, som har været opbevaret i sessions-variablen.

1. Du skal arbejde videre på dit CMS website
2. Du har i dine tidligere opgaver lavet en validering af felterne i din login formular, således at den skriver en fejlmeddelelse, hvis man ikke har indtastet noget i enten brugernavn eller password feltet. Hvis der var indtastet noget, flyttede du det over i din $\_SESSION variabel, så det ville blive ved med at være tilgængeligt. Dette skal du nu bruge til at sikre dig, at brugeren er logget ind, så der kan blive vist nogle andre menupunkter, så længe brugeren er logget ind:
   1. Du skal teste på, om der står noget i din $\_SESSION[”login-submit”] (her skal du selvfølgelig bruge det navn, som du har givet din submit-knap på din loginformular)
   2. Du kan teste på, om der står noget i en variabel ved at bruge funktionen isset()  
      if (isset($\_SESSION[”login-submit”] {  
       … så skal der i sidemenuen være et link til en logout side (som er beskrevet i punkt 4) eller til en ”Min konto” side (i topmenuen). Min konto siden har vi ikke lavet endnu, så du gør blot linket tomt med <a href=”#”>  
      } else {  
       … så skal der være et link til din login side  
      }
3. Du skal nu sørge for, at brugeren bliver dirigeret videre til forsiden, når login lykkes. Sådan som du har skrevet koden til validering af formularfelterne, så vil siden reloade sig selv hver gang, brugeren trykker på submit-knappen (det er det, der sker, når formularens action sættes lig med $\_SERVER[”PHP\_SELF”]; )
   1. Vi kan bruge den indbyggede PHP-funktion header(), som fx kan bruges til at indsætte en redirect i HTML-kodens <head>, som dirigerer brugeren videre til den side, som vi specificerer. Du skal indsætte header-funktionen, hvis både brugernavn og password er indtastet. Koden til header-funktionen ser således ud:  
      header(”Location: index.php”)  
      Location angiver den side, der skal redirectes til. Det betyder også, at hvis du fx har din index-fil liggende en mappe over din login-fil, så skal du henvise til forsiden med ../index.php
4. Log ud siden skal indeholde den samme sidemenu, topmenu og footer som de andre sider. Herudover skal den indeholde en html tekst, som fortæller, at nu er du blevet logget ud
   1. Php-koden i toppen af siden skal gøre følgende:
      1. Starte (eller fortsætte) en session
      2. Fjerne alle sessions-data med funktionen session\_unset()
      3. Slå sessionen ihjel med session\_destroy()
      4. Tømme $\_SESSION variablen ved at sætte den lig med et tomt array
5. Husk, at du skal have en session\_start() på alle de sider, hvor brugeren skal blive ved med at være logget ind