|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Visualisering  V3.1 | | | |  |
| **Emne:** | PHP og database | **Opgave:** | MySQLi og SQL |
| **Formål:** | * Jeg kan anvende MySQLi til at forbinde til en database | | |
| * Jeg kan anvende SQL sætninger | | |
| * Jeg kan anvende formularer sammen med databaser | | |

**Opgave: Opret tabel til brugere og kontrollér password**

1. Du skal først oprette en ny database og tabel ved at følge nedenstående punkter
   1. På cPanel forsiden skal du vælge MySQL® Databases og her oprette en ny database (du vælger selv, hvad du vil kalde den) - du skal bruge dette databasenavn, når du i din PHP kode skal oprette forbindelse til databasen
   2. Du skal også oprette en ny bruger med password (disse to skal du også bruge, når du skal forbinde til databasen)
   3. Til sidst skal du tilføje din bruger til databasen og give den alle rettigheder
   4. Gå tilbage til forsiden i din cPanel
2. Du skal nu have oprettet en tabel i databasen, som hedder users. Det gør du ved at gå ind i PhpMyAdmin via cPanel forsiden
   1. I tabellen users skal der være tre felter:
      1. ID: Datatype: INT, Længde/værdi angives ikke, Indeks: PRIMARY (når du vælger primary kommer der en ny dialogboks op – her klikker du blot på Udfør), A\_I afkrydses (A\_I står for auto increment, så den tæller automatisk op, når der kommer en ny bruger i tabellen)
      2. Username – Datatype: VARCHAR, Længde/værdi: 50, Indeks: UNIQUE (når du vælger unique kommer der en ny dialogboks op – her klikker du blot på Udfør)
      3. Password – Datatype: VARCHAR, Længde/værdi: 1024
   2. Nulværdier er ikke tilladt i nogle af felterne
   3. Klik dig ind på din database i menuen til venstre (træstrukturen)
   4. Klik dig herefter ind på din users tabel
   5. Opret et par brugere i tabellen ved at klikke på Indsæt i menulinjen øverst
   6. Indtast brugernavn og adgangskode i den øverste formular (du skal ikke indtaste noget i ID, for den genereres automatisk af databasen)
   7. Klik på Udfør
   8. I næste vindue går du blot tilbage til Indsæt eller til Vis (hvis du trykker Udfør igen, så forsøger databasen at indsætte samme bruger to gange, og du vil få en fejl, fordi der ikke må være to brugere i tabellen med samme brugernavn)
3. Opret forbindelse til databasen i php og test, om brugeren har indtastet et korrekt password
   1. I din login-formular skal du oprette forbindelsen til databasen
   2. Husk at teste på, om forbindelsen bliver oprettet korrekt ellers skal du udskrive databasens fejlmeddelelse om forbindelsen (connect\_error)
   3. Alt der herefter har med databasen at gøre, skal du lave i den if-sætning, hvor du har testet, om både brugernavn og adgangskode er udfyldt
   4. Start med at oprette variablen $username. Den bruger vi midlertidigt til at lægge værdien ind i, som brugeren har indtastet i formularfeltet. Dette gør vi, fordi vi skal have tilføjet et citationstegn (’) både før og efter brugernavnet, så vi kan bruge det i vores SQL sætning. Brug dot concatenatoren (punktum) til at tilføje et enkelt citationstegn både før og efter det brugernavn, som ligger i din $\_POST variabel
   5. Herefter skal vi have hentet den rigtige bruger ud af databasen – dvs den bruger, hvor brugernavnet er det samme, som det brugeren lige har indtastet i formularen. Du kan hente en bestemt post ud af en tabel ved at bruge SQL keywordet WHERE på denne måde:  
      $result = $db->query(”SELECT \* FROM users WHERE Username = $username”);  
      ($db er den databaseforbindelse, som du har oprettet tidligere, da du oprettede et MySQLi objekt).  
      Query-sætningen her returnerer et objekt af typen MySQLi result, som lægges ind i variablen $result.  
      Querien henter alle felter (\*) fra tabellen users hvor feltet Username er lig med det, som vi lige har lagt ind i variablen $username i punkt d.
   6. Vi skal nu have lagt vores MySQLi\_result objekt over i et associative array (i opgaven her er $row dit associative array - man bruger ofte variabelnavnet "row" til at lægge databasedata i, da de også ligger i en række, når man fx kigger på tabellen i phpmyadmin). Når vi har lagt de hentede data over i $row arrayet, kan vi begynde at arbejde med data’ene fra databasen på samme måde, som vi ellers arbejder med arrays. For at lægge dataene over i arrayet $row bruger vi funktionen fetch\_assoc() på følgende måde:  
      $row = $result->fetch\_assoc();
   7. Du skal nu teste på, om det der ligger i databasetabellens felt ”Password” er det samme, som der blev tastet ind i formularens password-felt. Din kode er naturligvis afhængigt af, hvad du har kaldt dit password-felt i formularen, men hos mig ser koden således ud:  
      if ($row[”Password”] == $\_POST[”login-password”])
   8. Hvis de to passwords er ens, skal du udføre den kode, som du lavede i de forrige opgaver: Sætte $\_SESSIONs værdi fra submit-knappen lig med $\_POSTs værdi fra submit knappen og herefter redirecte til index siden
   9. Hvis de to passwords ikke er ens, skal du skrive ”Du har indtastet en forkert adgangskode!”
   10. EKSTRA: Hvis du har tid og overskud kan du se, om du kan lave følgende nice-to-have for brugerne: Hvis man indtaster et forkert password, slettes det brugernavn, man har indtastet fra feltet username. Se, om du kan få brugernavnet til at blive stående, når siden reloades ved tryk på submit-knappen