|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 indsætte og slette data i databaser | | |

**Opgave: Opret og slet bruger**

1. I forrige opgave arbejdede vi på en midlertidig tabel, som kun indeholdt brugernavn og adgangskoder på vores brugere. Du skal nu lave en ny tabel, som skal være vores egentlige brugertabel med alle de informationer, som vi vil gemme om vores brugere. Denne tabel skal også danne grundlaget for de formularfelter, som du skal have på din ”opret bruger” side.  
   Tabellen skal indeholde følgende felter:
   1. ID Type: INT  
       Længde/værdi angives ikke  
       Indeks: PRIMARY  
       Auto increment: ja
   2. Username Type: VARCHAR  
       Længde/værdi: 50  
       Index: UNIQUE
   3. Password Type: VARCHAR  
       Lægde/værdi: 1024
   4. Firstname Type: VARCHAR  
       Længde/værdi: 50  
       Nulværdi: ja
   5. Lastname Type: VARCHAR  
       Længde/værdi: 50  
       Nulværdi: ja
   6. Address Type: VARCHAR  
       Længde/værdi: 100  
       Nulværdi: ja
   7. Postcode Type: SMALLINT  
       Nulværdi: ja
   8. Country Type: VARCHAR  
       Længde/værdi: 50  
       Nulværdi: ja
   9. Email Type: VARCHAR  
       Længde/værdi: 50
   10. Website Type: VARCHAR  
        Længde/værdi: 100  
        Nulværdi: ja
2. Inden du kan lægge de indtastede data over i databasen, skal du først teste på, om brugeren har trykket på submit-knappen, dvs. al din efterfølgende php-kode skal kun udføres, hvis der er udført en ”POST” metode. Dette kan du teste med følgende kode:  
   if ($\_SERVER[”REQUEST\_METHOD”] == ”POST”)
3. Hvis brugeren har trykket på submit-knappen skal du teste på, om brugernavnet allerede findes i tabellen
   1. Start med at hente data fra tabellen hvor feltet username er lig med det input, som brugeren har indtastet i formularfeltet username
   2. Herefter kan du teste på, om der er kommet nogle resultater retur (altså at brugernavnet allerede findes i tabellen)  
      TIP: Du kan fx bruge propertyen num\_rows og test, om den er lig med 0. Hvis num\_rows er nul, er der ikke blevet returneret nogle rækker fra tabellen, og det betyder, at brugernavnet ikke eksisterer
4. Hvis brugernavnet ikke findes i tabellen, skal du teste på, om brugeren har indtastet samme password to gange
   1. Husk at gøre brugeren opmærksom på det, hvis de to passwords ikke matcher
   2. EKSTRAOPGAVE: Som god brugervenlighed skal du sørge for, at alle data, som brugeren lige har indtastet i formularen bliver stående i formularen, hvis brugeren får en password match fejl – bortset fra i felterne ”password” og ”gentag password”, som tømmes, så de er klar til at prøve igen
5. Opret den nye bruger i tabellen ved at [indsætte værdierne fra din formular i databasen](https://www.w3schools.com/sql/sql_insert.asp) (bortset fra gentag password feltet). Linket fører til w3schools hvor du kan læse om INSERT INTO - SQL
   1. Husk at de værdier du indsætter med din SQL sætning skal være omgivet af citationstegn og at feltnavnene i tabellen ikke skal være omgivet af citationstegn
   2. Test i phpmyadmin om du fik indsat den nye bruger i tabellen

**SLET BRUGER**

Du skal nu bruge din viden om, hvordan du arbejder med databasen til at brugeren kan slette sig selv igen. Dette skal brugeren have mulighed for under siden ”Min konto”.

Du vil i denne opgave ikke få en trin-for-trin vejledning til, hvordan du skal gøre det.

SQL sætningen til at slette data er:

DELETE FROM users WHERE username = [kriterie]

Husk at teste på, om brugeren allerede eksisterer i databasen, og husk også at tage højde for, hvis der sker fejl undervejs – enten i databaseforbindelsen, hentning af data til test og sletning af data.