Labb 2

E-tjänster och webbprogrammering 7.5 hp VT-14

Introduktion

I denna laboration kommer vi att arbeta med `HTML`, `CSS`, `PHP`, `Javascript` och `MySQL`. Målet med denna laboration är att bygga en applikation med inloggningsfunktionalitet. Samt användarregistrering. Med fördel utgår du ifrån den applikation du byggde i labb 1 (kom ihåg att säkerhetskopiera), men vill du börja ifrån "scratch" går det förstås lika bra.

För att lösa denna labb behöver vi oundvikligen bekanta oss med `HTTP POST` och `HTTP GET`. Såväl som koncepten `hash`:ning och `salt`:ning.

Föreslagen struktur

Denna laboration består av 6 faser. Dessa är inte separata uppgifter, utan föreslagna steg för att bygga ovan nämnd applikation. Denna laboration kräver ingen inlämning. Men för att få en uppfattning om hur ett färdigt projekt efter denna labb skulle kunna se ut – beakta nedan filstruktur.

- `labb2-fornamn-efternamn` (mapp)
  - `index.php`
  - `login.php`
  - `register.php`
  - `assets` (mapp)
    - `img` (mapp med bilder)
    - `js` (mapp med javscriptfiler)
    - `css` (mapp med stylesheets)

Hur många filer ni har i varje mapp beror förstås på vilken strategi ni väljer att lösa problemet med. Läs vidare i instruktionerna för klarare förståelse.

Uppgifter

Nedan följer uppgifterna som resulterar i inlämningen ovan.

Uppgift 1 Designa databasen

Låt oss börja med att designa databasen. Målet är att (minst) lagra följande information om en användare: e-post, password och salt. Läs på om du inte vet vad salt är.

1. Skapa en användartabell i databasen (enl. ovan spec.).
2. Tabellen måste ha en primärnyckel (`PK`) som automatiskt inkrementeras.

Uppgift 2 Skapa en registreringssidan

Målet med detta steg är att skapa en sida som innehåller ett formulär. Detta formulär ska innehålla alla fält som användaren behöver för att kunna registrera sig. Alltså skapa en användare.

1. Börja med att skapa `registration.php`
2. Fyll sidan med ett `HTML`-formulär innehållandes de komponenter användaren behöver för att fylla i relevant data.
3. Skriv en `PHP`-funktion som du kan använda för att randomisera fram den sträng som ska användas som salt till lösenordet.
4. När användaren sedan klickar på registreringsknappen, se till att skicka datat till en sida som sparar informationen i databasen.
5. Kom ihåg att lösenordet ska krypteras tillsammans med saltet.
6. Användaren måste på något sätt meddelas om huruvida registreringen lyckades eller inte.

Uppgift 3 Skapa inloggningssidan

Nu ska vi skapa en inloggningssida för samma applikation.

1. Börja med att skapa filen `login.php`.
2. Fyll sidan med ett `HTML`-formulär och de komponenter användaren behöver för att fylla i relevant data.

3. Hämta saltet från databasen.
4. Kryptera salt + lösenord och matcha mot det lösenord som finns i databasen.
5. Skapa en `session` för användaren.
6. Användaren måste på något sätt meddelas om huruvida inloggningen lyckades eller inte.

Uppgift 4 Skydda övriga sidor

Nu måste vi se till att användare som inte är inloggade inte kommer åt de sidor som kräver inloggning. Det räcker inte med att vi bara inte länkar till de sidor användaren kommer åt. Om hen skriver in adressen till en skyddad sida kan denna komma åt sidan om vi inte faktiskt kollar om denna är inloggad.

Om en användare försöker nå en sida trots att den inte är inloggad behöver vi välja emellan att antingen (A) ge användaren ett meddelande om att de måste logga in, eller (B) förflytta användaren (`redirect`) till inloggningssidan automatiskt. Kom ihåg att detta gäller för alla sidor som måste skyddas.

Ni behöver även skapa en knapp eller länk där användaren kan logga ut. Alltså "radera" användarens `session`.

Uppgift 5 Smycka sidan

Se nu till att sidan håller ett enhetligt intryck i sin design. Oseriösa förslag godtas ej.

Uppgift 6 Förslag på viktiga uttökningar

- Se till att det inte går att skapa användare med samma e-post som någon annan.
- Kontrollera att användaren anger ett "starkt" lösenord.