Εργαστηριακή Άσκηση Ανάπτυξης Απλού Δυναμικού Ιστοχώρου σε PHP και MySQL

Δημιουργία ενός συστήματος για εγγραφή χρηστών και σύνδεση στο σύστημα

Μηνάς Δασυγένης, mdasyg@ieee.org | 2021, 2022

Σε αυτή την άσκηση θα δημιουργήσουμε ένα σύστημα για εγγραφή χρηστών και σύνδεση χρησιμοποιώντας τη γλώσσα προγραμματισμού PHP. Θα χρησιμοποιήσουμε μια τοπική εγκατάσταση webserver + βάσης δεδομένων από: https://www.apachefriends.org/download.html

- Κατεβάστε την έκδοση XAMPP για το λειτουργικό σας σύστημα και εγκαταστήστε την.
- Θα πρέπει να δημιουργήσουμε μια βάση δεδομένων για τη φύλαξη των στοιχείων.
- Μέσα στη βάση δεδομένων θα δημιουργήσουμε έναν πίνακα για τα στοιχεία των χρηστών.
- Θα πρέπει να εκτελούνται οι παρακάτω υπηρεσίες: Apache , Mysql
- Πατάμε **admin** στο **Mysql** στο Xampp Control Panel για να δημιουργήσουμε τη βάση δεδομένων.
    - Databases → Database Name → msc και πατάμε Create

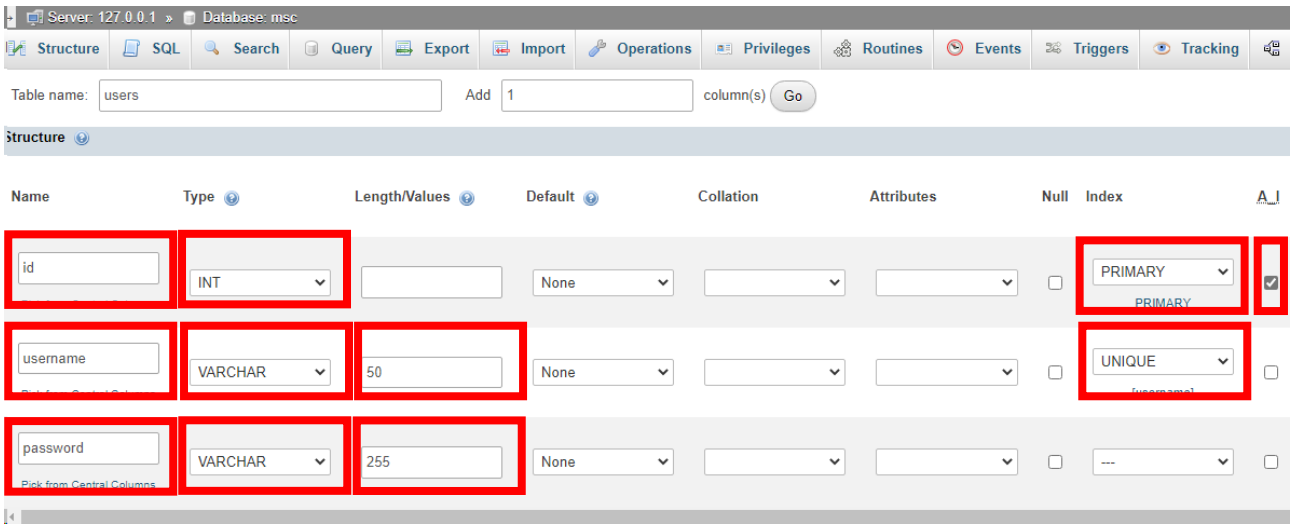
(προσοχή, όχι ελληνικούς χαρακτήρες, όχι κενά ή ειδικούς χαρακτήρες, μικροί λατινικοί)

- Από την αριστερή στήλη έχει επιλεχθεί η βάση δεδομένων.
- Θα πρέπει να δημιουργήσουμε ένα πίνακα.
    - Structure → Create Table → users (στο όνομα) με 3 στήλες. Επιλέξτε GO.

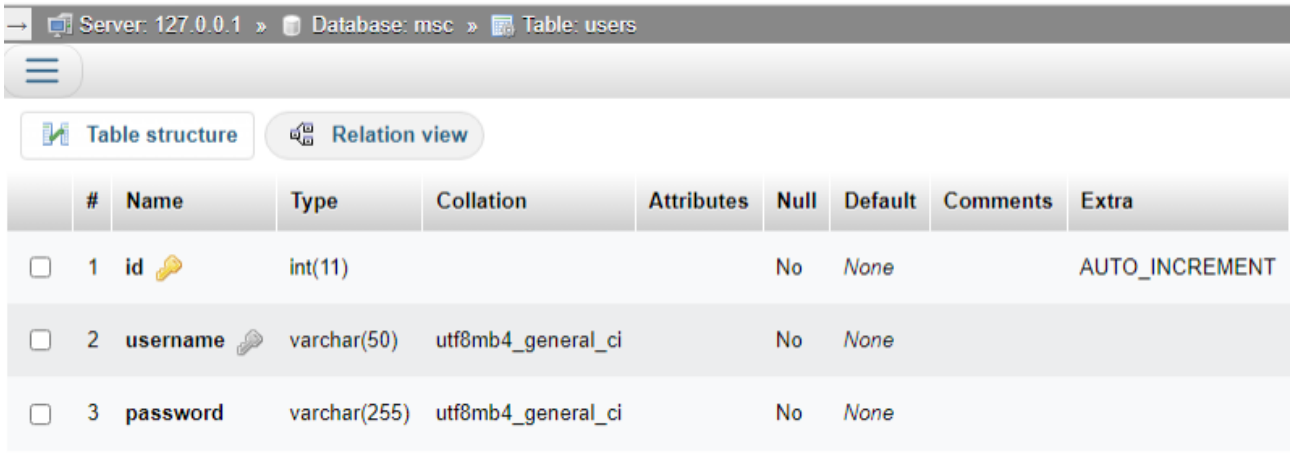
Τα πεδία που θα τροποποιήσουμε φέρουν τα ονόματα (Type, Length/Values, Null, Index, AI (auto increment)) (Δείτε την εικόνα μετά τον πίνακα)

id	INT NOT NULL PRIMARY KEY AUTO_INCREMENT
username	VARCHAR(50) NOT NULL UNIQUE
password	VARCHAR(255) NOT NULL

Η παρακάτω εικόνα δείχνει τις επιλογές που πρέπει να κάνετε, ώστε να δημιουργηθεί ο πίνακας στη βάση δεδομένων με τα σωστά πεδία.



Πατήστε SAVE και θα δημιουργηθεί η βάση δεδομένων. Μπορείτε να επιλέξετε το structure για να δείτε τη δομή που έχετε δημιουργήσει. Θα πρέπει να είναι παρόμοιο με την παρακάτω εικόνα:



Το πρώτο βήμα είναι να διαπιστώσουμε ότι ο διερμηνευτής κώδικα PHP λειτουργεί σωστά. Για να το διαπιστώσουμε θα δημιουργήσουμε το παρακάτω αρχείο στη γλώσσα προγραμματισμού PHP (για αυτό θα έχει κατάληξη .php) με όνομα info.php μέσα στο φάκελο **c:\xampp\htdocs** του webserver.

```
<?php
phpinfo();
?>
```

Το αρχείο θα πρέπει να το αποθηκεύσετε ως: "**c:\xampp\htdocs\info.php**"

Επισκεφτείτε τη σελίδα http://localhost/info.php .

Αν όλα έχουν γίνει σωστά, τότε θα σας εμφανίσει πληροφορίες σχετικά με την εγκατάσταση της PHP και τις βιβλιοθήκες που υποστηρίζει. Αν σας εμφανίσει αυτές τις πληροφορίες μπορείτε να συνεχίσετε παρακάτω.

Το επόμενο βήμα είναι να δημιουργήσουμε ένα αρχείο .php για να συνδέσουμε τον ιστοχώρο μας στη βάση. Δημιουργήστε το παρακάτω αρχείο με όνομα database.php. Οι τιμές για την IP του Server,

το όνομα σύνδεσης στη βάση δεδομένων, τον κωδικό για τη βάση δεδομένων και το όνομα της βάσης δεδομένων, δίνονται από το διαχειριστή της βάσης δεδομένων. Για το xampp είναι τα παρακάτω:

```
<?php
define('DB_SERVER', '127.0.0.1');
define('DB_USERNAME','root');
define('DB_PASSWORD','');
define('DB_NAME','msc');
$link=mysqli_connect(DB_SERVER,DB_USERNAME,DB_PASSWORD,DB_NAME);
if ($link == false)
{
die('<font color=red> ERROR: Could not connect: ' . mysqli_connect_error() . '</font>') ;
}
else
{echo("<font color=green> OK connected to database</font>");}
?>
```

- Αποθηκεύστε το αρχείο ως "**c:\xampp\htdocs\database.php**"
- Επισκεφτείτε τη σελίδα http://localhost/database.php Αν έχετε υλοποιήσει σωστά τη σύνδεση δε θα σας εμφανίσει κανένα μήνυμα λάθους. Θα σας εμφανίσει το μήνυμα "Ok connected to database". ➔ Αν δείτε αυτό το μήνυμα μπορείτε να συνεχίσετε στη δημιουργία της φόρμας. Αν σας εμφανίσει warning ή error, τότε επιλύστε το πριν συνεχίσετε.

Δημιουργία της φόρμας εγγραφής χρηστών

Η επόμενη σελίδα .php είναι η register.php που παρουσιάζει μια φόρμα για την εγγραφή των χρηστών. Η φόρμα εισόδου δεδομένων ονομάζεται τεχνικά και 'frontend (εμπρόσθιο τμήμα)' γιατί αυτό βλέπει ο χρήστης, ενώ η σελίδα επεξεργασίας δεδομένων ονομάζεται 'backend (οπίσθιο τμήμα)'.

Σε αυτό το παράδειγμα έχουμε μια σελίδα που έχει και τις 2 λειτουργικότητες. Αν ο χρήστης δεν έχει υποβάλλει κάτι, του εμφανίζεται η φόρμα εισόδου δεδομένων. Αν ο χρήστης έχει υποβάλλει κάτι θα γίνει η επεξεργασία των δεδομένων.

** Η σελίδα αυτή δεν έχει κώδικα για την επεξεργασία των δεδομένων ακόμη **

Τοποθετήστε τον παρακάτω κώδικα στη σελίδα register.php.

```
<html>
<body>
<?php
if(isset($_POST['username'])    &&    isset($_POST['password'])    &&
isset($_POST['confirm_password']))
{    echo "Processing..."; }
else
{
echo <<< EOT
<!-- form -->
<h1> register new user</h1>
<form method="post">
Name:
<input type="text" name="username"><br>
Password:
<input type="text" name="password"><br>
Confirm Password:
<input type="text" name="confirm_password"><br>
<input type="submit">
<input type="reset">
</form>
EOT;
}
?>
</body>
</html>
```

Επισκεφτείτε τη σελίδα http://localhost/register.php . Αν έχετε υλοποιήσει σωστά τη σύνδεση δε θα σας εμφανίσει κανένα μήνυμα λάθους. Επιβεβαιώστε ότι εμφανίζεται η φόρμα. Επιβεβαιώστε ότι αν υποβάλλετε κάτι θα αναφερθεί μόνο το μήνυμα **"Processing...".**

Για να εμπλουτίσουμε αυτή τη σελίδα θα κάνουμε τα παρακάτω:

(α) τα πεδία της φόρμας που έχουν τον κωδικό θα πρέπει να είναι type=”password”

(β) θα πρέπει να μπει κώδικας για τη σύνδεση στη βάση δεδομένων

(γ) θα πρέπει να μπει κώδικας που επεξεργάζεται την είσοδο και εισάγει το χρήστη στη βάση δεδομένων.

Ο νέος κώδικας register.php είναι ο παρακάτω (αντικαταστήστε το άλλο αρχείο):

```
<html>
<body>
<?php
if(isset($_POST['username']) && isset($_POST['password']) && isset($_POST['confirm_password']))
{
     require("database.php");
     echo "Processing...<br>";
     $username=$_POST['username'];
     $password=$_POST['password'];
     $confirm_password=$_POST['confirm_password'];

     if($password != $confirm_password)
     {die("ERROR: Password do not match.<br>");}
     echo "Passwords match..continuing....<br>";
    $encrypted_password=password_hash($password,PASSWORD_DEFAULT);
     $sql="INSERT INTO users (username,password) values ('$username','$encrypted_password')";
     $result=mysqli_query($link,$sql);
     if ($result==TRUE)
     {echo "User added successfully.<br>";}
     else
     {echo "ERROR: user was not added: ".mysqli_error($link);}
}
else
{

echo <<< EOT
<!-- form -->
<h1> register new user</h1>
```

```
<form method="post">
Name:
<input type="text" name="username"><br>
Password:
<input type="password" name="password"><br>
Confirm Password:
<input type="password" name="confirm_password"><br>
<input type="submit">
<input type="reset">
</form>
EOT;
}
?>
</body>
</html>
```

Επισκεφτείτε τη σελίδα http://localhost/register.php . Αν έχετε υλοποιήσει σωστά τη σύνδεση δε θα σας εμφανίσει κανένα μήνυμα λάθους. Επιβεβαιώστε ότι εμφανίζεται η φόρμα. Επιβεβαιώστε ότι αν υποβάλλετε κάτι θα αναφερθεί το μήνυμα **"Processing..." .** τ και δείτε από το διαχειριστικό περιβάλλον της βάσης δεδομένων αν έχουν εισαχθεί κάποια δεδομένα ή όχι στον πίνακα msc.

➔Αν έχουν εισαχθεί δεδομένα μπορούμε να πάμε στο επόμενο βήμα.

Η συγκεκριμένη φόρμα έχει 2 προβλήματα που θα διορθώσουμε στην επόμενη έκδοση, η οποία έχει βελτιώσεις ως εξής:

(α) Η είσοδος από το χρήστη πρέπει πάντα να φιλτράρεται, ώστε να αποφεύγονται κακόβουλοι χρήστες που εισάγουν προβληματικό κώδικα. Ως εκ τούτου πριν τις αναθέσουμε σε μεταβλητές θα καλέσουμε τις trim() και filter_var().

(β) Αν υπάρχει ίδιο όνομα χρήστη κατά το insert στη βάση δεδομένων θα έχουμε πρόβλημα επειδή το username έχει το χαρακτηριστικό UNIQUE στη mysql. Ως εκ τούτου απαιτείται πρώτα να κάνουμε ένα ερώτημα στη βάση δεδομένων και να δούμε αν υπάρχει. Αν υπάρχει ενημερώνουμε το χρήστη, διαφορετικά συνεχίζουμε ως έχει.

Ο νέος μας κώδικας register.php που αντικαθιστά τον προηγούμενο είναι ο παρακάτω:

```
<html>
<body>
<?php
if(isset($_POST['username']) && isset($_POST['password']) && isset($_POST['confirm_password']))
{
     require("database.php");
     echo "Processing...<br>";
     $username=filter_var(trim($_POST['username']),FILTER_SANITIZE_STRING);
     $password=filter_var(trim($_POST['password']),FILTER_SANITIZE_STRING);
     $confirm_password=filter_var(trim($_POST['confirm_password']),FILTER_SANITIZE_STRING);

     if($password != $confirm_password)
     {die("ERROR: Password do not match.<br>");}
     echo "Passwords match..continuing....<br>";

    $encrypted_password=password_hash($password,PASSWORD_DEFAULT);

     //check if user exists
     $sql="SELECT * from users where username ='$username'";
     $result=mysqli_query($link,$sql);
     if ($result!=TRUE)
     {
     echo "Error in SQL query: ".mysqli_error($link);
     }

     $number_of_results=mysqli_num_rows($result);
     if ($number_of_results>0) { die("User already exists. Select another username");}
     $sql="INSERT INTO users (username,password) values ('$username','$encrypted_password')";
```

```
        $result=mysqli_query($link,$sql);
        if ($result==TRUE)
        { echo "User added successfully.<br>";}
        else
        {echo "ERROR: user was not added: ".mysqli_error($link);}
}
else
{
echo <<< EOT
<!-- form -->
<h1> register new user</h1>
<form method="post">
Name:
<input type="text" name="username"><br>
Password:
<input type="password" name="password"><br>
Confirm Password:
<input type="password" name="confirm_password"><br>
<input type="submit">
<input type="reset">
</form>
EOT;
}
?>
</body>
</html>
```

Προσοχή: Η προηγούμενη έκδοση παρέχει ελάχιστη ασφάλεια και ΔΕΝ πρέπει να την ανεβάσετε σε ένα διακομιστή διαδικτύου. Σε περίπτωση που θέλετε μια φόρμα επεξεργασίας δεδομένων φόρμας με μεγαλύτερη ασφάλεια, τότε θα πρέπει να χρησιμοποιήσετε PDO.

Επισκεφτείτε τη σελίδα http://localhost/register.php . Αν έχετε υλοποιήσει σωστά τη σύνδεση δε θα σας εμφανίσει κανένα μήνυμα λάθους και μπορείτε να δοκιμάσετε τη φόρμα εγγραφής, και στη συνέχεια να δείτε τα δεδομένα που αποθηκεύονται στη βάση δεδομένων με το phpmyadmin.

➔ Αν δε θα σας εμφανίσει κανένα μήνυμα λάθους μπορείτε να συνεχίσετε στη δημιουργία της φόρμας εισόδου στο σύστημα.

Με παρόμοιο τρόπο με το αρχείο register.php θα δημιουργήσουμε το login.php το οποίο θα παρουσιάζει τη φόρμα (αν δεν έχουν υποβληθεί στοιχεία) ή αν έχουν υποβληθεί στοιχεία θα ψάχνει στη βάση δεδομένων να βρει το συγκεκριμένο username και μετά θα συγκρίνει με την κατάλληλη συνάρτηση επικύρωσης κωδικού, αν ο κωδικός που έδωσε ο χρήστης είναι ο ίδιος με αυτόν που έχει ο εγγεγραμμένος χρήστης.

ΠΡΟΣΟΧΗ: Δε μπορούμε να συγκρίνουμε τον hashed κωδικό μόνοι μας ως απλά αλφαριθμητικά (π.χ. if($hashed_password==$user_password) επειδή ακόμη και για τον ίδιο κωδικό δημιουργούνται διαφορετικά hashed strings. Απαιτείται ειδική συνάρτηση για αυτό.

Στη συνάρτηση login θέλουμε να διατηρήσουμε την πληροφορία ότι ο συγκεκριμένος χρήστης είναι ταυτοποιημένος. Για να το κάνουμε αυτό, χωρίς να χρειάζεται συνεχώς να στέλνουμε το username και το password στο HTML που είναι protocol stateless (=χωρίς εύκολη διατήρηση κατάστασης) η PHP υποστηρίζει τα sessions. Ενεργοποιούμε λοιπόν το session και αποθηκεύουμε την πληροφορία σύνδεσης μέσα σε αυτή την ειδική δομή που την χρησιμοποιούμε ως πίνακα $_SESSION.

Τοποθετήστε τον παρακάτω κώδικα στο αρχείο login.php.

```
<?php
session_start();
?>
<html>
<body>
<?php
if(isset($_POST['username']) && isset($_POST['password']))
{
     require('database.php');
     $username=filter_var(trim($_POST['username']),FILTER_SANITIZE
_STRING);
     $password=filter_var(trim($_POST['password']),FILTER_SANITIZE
_STRING);
     $encrypted_password=password_hash($password,PASSWORD_DEFAULT)
;
     echo "<br>Το όνομα που δώσατε είναι <i>".$username;
     echo "</i> και το επίθετο που δώσατε είναι:
".$password.'<br>';
```

```
     $sql="SELECT * FROM users where username='$username'";
     $result=mysqli_query($link,$sql);
     if ($result==FALSE)
     {echo "ERROR: Fail in searching database: ".
mysqli_error($link);
     }
     $number_of_results=mysqli_num_rows($result);
     if ($number_of_results>0)
          {
          $row = mysqli_fetch_row($result);
          $database_password=$row[2];
          $id=$row[0];
          $rc=password_verify($password,$database_password);
          if ($rc==TRUE)
               {
               echo "User Logged In<br>";
               $_SESSION['loggedin']=true;
               $_SESSION['id']=$id;
               $_SESSION['username']=$username;
               echo "<a href=privatepage.php> Visit Private
Page</a>";     }
               else
               {echo "ERROR: Invalid Password<br>";}
          }
          else
          {echo "ERROR: Invalid Username<br>";}
}
else
{
     echo '
     <h2>Login</h2>
     <form method="post">
     Username:
```

```
      <input type="text" name="username" value="" size=20>
      <br>
      Password:
      <input type="password" name="password" value="" size=20>
      <br>
      <input type="reset">
      <input type="submit">';
}
?>
</body>
</html>
```

Αν και σε ένα πραγματικό site δε θα υπάρχει διαφοροποίηση στο μήνυμα είτε έδωσε λάθος username είτε λάθος κωδικό, σε αυτό το παράδειγμα το κάνουμε για να καταλάβουμε τη διαφορά.

Παρατηρήστε ότι αρχικοποιούμε το session στις πρώτες γραμμές του αρχείου, πριν γίνει οτιδήποτε άλλο, όπως ορίζει η συνάρτηση session_start();

Παρατηρήστε ότι αν κάνει login αποθηκεύεται στο $_SESSION το όνομα του χρήστη όπως και η πληροφορία ότι έχει κάνει login με επιτυχία.

Επισκεφτείτε τη σελίδα http://localhost/login.php . Αν έχετε υλοποιήσει σωστά τη σύνδεση δε θα σας εμφανίσει κανένα μήνυμα λάθους και μπορείτε να κάνετε login. Δοκιμάστε με λάθος username, λάθος κωδικό ή σωστά συμπληρωμένα και τα 2 πεδία.

Αν δεν εμφανίζει κανένα μήνυμα λάθους τότε μπορείτε να συνεχίσετε στο επόμενο βήμα με τη δημιουργία μιας σελίδας που θα είναι προσβάσιμη μόνο σε χρήστες που έχουν κάνει login. Τοποθετήστε τον παρακάτω κώδικα στο αρχείο privatepage.php.

```
<?php
session_start();

if (!isset($_SESSION['loggedin'])) { header("Location: login.php");}
if ($_SESSION['loggedin'] != true ) { header("Location: login.php"); }

echo "Welcome user: $_SESSION[username]<br>";
echo "<a href=logout.php> logout</a>";
?>
```

Πηγαίνετε στη διεύθυνση http://localhost/privatepage.php πριν κάνετε σύνδεση και ενώ έχετε κάνει σύνδεση, και δείτε τη διαφορετική λειτουργικότητα.

Το επόμενο είναι να δημιουργήσουμε τη σελίδα logout.php που θα μας αποσυνδέει από το σύστημα.

```
<?php
session_unset();
session_destroy();
header('Location: login.php');
?>
```

Επισκεφτείτε τη σελίδα http://localhost/logout.php . Αν έχετε υλοποιήσει σωστά τη σελίδα δε θα σας εμφανίσει κανένα μήνυμα λάθους και θα σας ανακατευθύνει στην αρχική σελίδα σύνδεσης.

Σε αυτό το σημείο έχετε δημιουργήσει μια σελίδα εγγραφής χρηστών και τη σελίδα σύνδεσης και αποσύνδεσης χρηστών, τα πιο βασικά στοιχεία σε ένα δυναμικό ιστοχώρο. Συγχαρητήρια!

Μελλοντικά Επόμενα βήματα:

- Χρήση styles για την καλύτερη εμφάνιση. Για παράδειγμα δείτε τη σελίδα https://www.tutorialrepublic.com/php-tutorial/php-mysql-login-system.php πάνω στην οποία βασίστηκε αυτό το εργαστήριο, όπου έχει επιπρόσθετο κώδικα για τη στυλιστική εμφάνιση (styles).
- Ανάπτυξη υπόλοιπων σελίδων (π.χ. ενημέρωσης στοιχείων, διαγραφής χρηστών, αποστολής email).
- Χρήση Cookies για να θυμάται ο browser το username/password.