Student.cz

Zaprvé musíme říct, že pro ukládaní dat používáme databáze SQL, proto musíme připojit tam náš projekt .

$conn = new mysqli(  
 $db\_host,  
 $db\_user,  
 $db\_password,  
 $db\_db  
);

Také jsme zpracováváme chybně připojeni :

*/\*\*  
 \* Check for database connection errors.  
 \*/*if ($conn->connect\_error) {  
 */\*\*  
 \* Display the error number if the connection fails.  
 \* @var int  
 \*/* echo 'Errno: '.$conn->connect\_errno;  
  
 */\*\*  
 \* Display the error message if the connection fails.  
 \* @var string  
 \*/* echo '<br>';  
  
 */\*\*  
 \* Display the error message if the connection fails.  
 \* @var string  
 \*/* echo 'Error: '.$conn->connect\_error;  
  
 */\*\*  
 \* Exit the script if there's a database connection error.  
 \*/* exit();  
}

1.Začneme zpracováním registrační formy, přes POST metod dostáváme od uživatele data, a se začíná proces kontroly: Pro nás důležité aby data v databáze se neopakovala proto děláme kontrolu existuji-li uživatel s těmi samým data, pokud ano, tak vrátíme uživatele zpět aby ten formulář vyplnil znovu, všichni položky kromě hesel budou předvyplněny

$stmt = $conn->prepare("SELECT \* FROM users WHERE username = ? OR email = ?");  
$stmt->bind\_param("ss", $name, $email);  
$stmt->execute();  
$result = $stmt->get\_result();  
  
if ($result->num\_rows > 0) {  
 $\_SESSION['reg\_error'] = "User already exists";  
 $stmt->close();  
 header('Location: ../reg.php');  
 die();  
}

<label for="name">name</label>  
<input class="qwe" type="text" id="name" name="name" placeholder="name" value="<?php if (isset($\_SESSION['reg\_error'])) {  
 echo $\_SESSION['name'];  
} ?>"><br>  
<label for="mail">e-mail</label>  
<input class="qwe" type="email" id="email" name="email" placeholder="e-mail" required value="<?php if (isset($\_SESSION['reg\_error'])) {  
 echo $\_SESSION['email'];  
} ?>">

Jinak bude přeposlán na stánku autorizaci.

header('Location:../sign\_in.php');  
die();

2. Pro autorizaci zaprvé posíláme dotaz do naší databázi a kontrolujeme existuji-li uživatel, kontrola probíhá pomoci jména a hesla, kvůli tomu že všechny hesla uloženi v hešu potřebujeme využit:

if (password\_verify($pass\_to\_check, $row['password'])) {  
 */\*\*  
 \* If credentials are valid, set a user cookie and redirect to the index page.  
 \* @var int $id - User ID from the database.  
 \*/* $id = $row['id'];  
 setcookie('user', $id, time() + 2000, "/");  
 header('Location:../index.php');  
 die();

a pokud heslo a uživatelské jméno jsou zapsaný, tak nastavím cookies.

3.Pro vytvářeni témat diskuze také posíláme dotaz, v databáze

$sqlCheckTopic = "SELECT \* FROM topics WHERE topic = ?";  
  
*/\*\*  
 \* Prepare and bind parameters for the SQL query.  
 \*/*$stmtCheckTopic = $conn->prepare($sqlCheckTopic);  
$stmtCheckTopic->bind\_param('s', $topic);  
  
*/\*\*  
 \* Execute the prepared statement to check if the topic already exists.  
 \*/*$stmtCheckTopic->execute();

Pokud existuje témata s tím samým názvem:

if ($resultCheckTopic->num\_rows > 0) {  
 */\*\*  
 \* Set a session variable indicating that the topic already exists.  
 \*/* $\_SESSION['topic'] = "Topic already exists";  
  
 */\*\*  
 \* Redirect to the create\_topic.php page.  
 \*/* header('Location: ../create\_topic.php');  
  
 */\*\*  
 \* Terminate the script execution.  
 \*/* die();  
}

4.Pro přidáni správy uživatele používáme „foreign key“ každá spává v databáze hraní „id“ tématy pod kterou tato spává byla uložena

echo "<input type='hidden' name='topic\_id' value='$topic\_id'>";

posíláme id spolu se správou, ale skrytě

$stmt = $conn->prepare("INSERT INTO comments (username, topic\_id, message) VALUES (?, ?, ?)");  
$stmt->bind\_param('sis', $author, $topic\_id, $text);  
$stmt->execute();

A pak už ukládáme všechno.

5. Editace uživatelských dat také existuji tady

echo "<form action='edit\_user\_data.php' method='post'>";  
echo "<button type='submit' name='edit\_username'>Edit username</button>";  
echo "<button type='submit' name='edit\_email'>Edit Email</button>";  
echo "<button type='submit' name='edit\_password'>Edit Password</button>";  
echo "</form>";

Po stlačeni jednoho z třech tlačítek, můžeme editovat vybraná data, tlačítko převádí nás na další formulář, takhle vypadá formulář pro změnu jména

if (isset($\_POST['edit\_username'])) {  
  
 echo "<form action='src/edit.php' method='post'>";  
 echo "<label for='new\_username'>New name</label>";  
 echo "<input class='qwe' type='text' name='new\_username' value=";  
 if (isset($\_SESSION['change\_error'])) {  
 echo htmlspecialchars($\_SESSION['new\_username']);  
 }  
 echo ">";  
  
 echo "<label for='password'>Enter password</label>";  
 echo "<input class='qwe' type='password' name ='password'>";  
 echo "<label for='repass'>Password again</label>";  
 echo "<input class='qwe' type='password' name ='repass'>";  
 echo "<input class='submit' type='submit' value='submit'>";  
 echo "</form>";

změna uživatelských dat funguje skoro stejně jak i registrace, data bude kontrolována na existence duplikátu:

while ($row = $result->fetch\_assoc()) {  
 if (strtolower($row['username']) == strtolower($new\_username)) {  
 $\_SESSION['change\_error'] = 'name already exists';  
 header('Location:../user\_account.php');  
 die();  
 }  
}

6. Všichni obrázku ukládáme v souboru které se nahází v našem projektu, v databáze ukládáme jen jméno tohoto obrázku: (\_\_DIR\_\_ vrací Absolutní cestu, což znamená z kořenového adresáře)

if (isset($\_FILES['user\_pic'])) {  
 */\*\*  
 \* Move the uploaded file to the 'images' directory.  
 \* @var string $file\_name - Name of the uploaded file.  
 \*/* move\_uploaded\_file($\_FILES['user\_pic']['tmp\_name'], *\_\_DIR\_\_* . "/../images/" . $\_FILES['user\_pic']['name']);  
 $file\_name = $\_FILES['user\_pic']['name'];

Takhle zobrazíme odrazky:

while ($row = $result->fetch\_assoc()) {  
 $src = $row['userpic'];  
 echo "<ul class='user\_data' >";  
 if ($row['userpic'] == '')  
 {  
 echo "<img class='userpic' src='images/unknown\_user.png'>";  
 }  
 elseif (!file\_exists("images/$src"))  
 {  
 echo "<img class='userpic' src='images/unknown\_user.png'>";  
 }  
 else {  
 echo "<img class='userpic' src='images/$src' alt='car'>";  
 }

7. Tento skript zajišťuje vyhledávání na fóru pomocí AJAX, což umožňuje uživatelům okamžitě vidět přibližné výsledky. Při zadávání textu do pole pro hledání skript odesílá asynchronní požadavek na server, který zpracovává hledaný dotaz.

xhr.open('POST', '../src/search.php', true);  
 xhr.setRequestHeader('Content-type', 'application/x-www-form-urlencoded');  
  
  
 xhr.send('query=' + encodeURIComponent(query));  
} else {  
  
 *document*.getElementById('search-results').innerHTML = '';

PHP-скript přijímá hledaný dotaz od JavaScriptu, provádí hledání v databázi a vrací výsledky. Pokud je zadán hledací dotaz, sestavuje se SQL dotaz s použitím zástupných znaků pro hledání podobných témat. Po provedení dotazu následuje cyklus přes výsledky a témata se zobrazují jako odkazy na příslušné stránky.

if (isset($\_POST['query'])) {  
 */\*\*  
 \* Build the search query with a wildcard pattern.  
 \* @var string $query - Search query with wildcard patterns.  
 \*/* $query = '%' . $\_POST['query'] . '%';  
  
 */\*\*  
 \* Prepare a SQL statement to select topics matching the search query.  
 \* @var string $sql - SQL query with a parameterized search condition.  
 \* @var mysqli\_stmt $stmt - Prepared statement to execute the query.  
 \*/* $sql = "SELECT \* FROM topics WHERE topic LIKE ?";  
 $stmt = $conn->prepare($sql);  
  
 */\*\*  
 \* Bind the search query parameter to the prepared statement.  
 \*/* $stmt->bind\_param('s', $query);