

Documentation για τα αρχεία Login, Main Page, Insert

(χωρίς σχολιασμό του CSS styling επειδή θα ακολουθήσουν αλλαγές)

Η παρούσα τεκμηρίωση παρέχει μια ολοκληρωμένη εξήγηση της λειτουργικότητας των PHP scripts login.php και main_page.php.

Καλύπτει κάθε γραμμή κώδικα, εξηγώντας πώς λειτουργεί το σύστημα για τον έλεγχο ταυτότητας χρήστη (user authentication), τη χρήση των συνεδριών (php sessions), και την εμφάνιση της κύριας σελίδας μετά τη σύνδεση.

login.php - User Login Page

Το script login.php διαχειρίζεται το user authentication με τον εξής τρόπο: συνδέεται σε μια βάση δεδομένων MySQL, επικυρώνει τα εισαγόμενα από τον χρήστη στοιχεία μέσω μιας Stored Procedure και ξεκινά μια συνεδρία(session) σε περίπτωση επιτυχούς σύνδεσης.

1. Έναρξη Συνεδρίας (Session)

```
<?php
```

```
session_start();
```

- Αυτή η γραμμή αρχικοποιεί ή συνεχίζει(resumes) μια συνεδρία. Η PHP χρησιμοποιεί συνεδρίες για να αποθηκεύει δεδομένα χρήστη σε πολλές σελίδες. Σε αυτό το script, χρησιμοποιείται για την παρακολούθηση της κατάστασης σύνδεσης του χρήστη.

2. Database Connection Parameters

```
$dsn = "mysql:host=localhost;port=3306;dbname=html"; // Βάση δεδομένων
$dbusername = "root"; // Όνομα χρήστη
$dbpassword = "556782340"; // Κωδικός
$error = ""; // Μεταβλητή για σφάλματα
$username = ""; // Μεταβλητή για το όνομα χρήστη
$password = ""; // Μεταβλητή για τον κωδικό
```

- **\$dsn**: Το Data Source Name (DSN) καθορίζει τις λεπτομέρειες σύνδεσης με τη βάση δεδομένων. Περιλαμβάνει τον host (localhost), τη θύρα (3306) και το όνομα της βάσης δεδομένων (html).
- **\$dbusername & \$dbpassword**: Αυτές οι μεταβλητές αποθηκεύουν τα credentials για τη σύνδεση με τη βάση δεδομένων.
- **\$error, \$username, \$password**: Μεταβλητές για την αποθήκευση των δεδομένων της φόρμας και των μηνυμάτων σφάλματος σε περίπτωση αποτυχίας της διαδικασία σύνδεσης.

3. PDO Connection και Error Handling

```
try {
    $pdo = new PDO($dsn, $dbusername, $dbpassword);
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
}
```

- **new PDO()**: Δημιουργεί μια σύνδεση με τη βάση δεδομένων χρησιμοποιώντας το καθορισμένο DSN, το όνομα χρήστη και τον κωδικό πρόσβασης. Εάν η σύνδεση αποτύχει, 'πετάει' Exception.
- **setAttribute()**: Αυτή η μέθοδος θέτει το ATTR_ERRMODE στην τιμή ERRMODE_EXCEPTION, που σημαίνει ότι σε περίπτωση σφάλματος θα προκαλέσει μήνυμα εξαίρεσης αντί να αποτύχει χωρίς καμία ειδοποίηση.

4. Η Μέθοδος POST Request

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {  
    $username = $_POST['username'];  
    $_SESSION['username'] = $username;  
    $password = $_POST['password'];
```

- **\$_SERVER["REQUEST_METHOD"] == "POST"**: Φροντίζει ώστε το block κώδικα να εκτελεστεί μόνο όταν η φόρμα έχει υποβληθεί μέσω της μεθόδου POST.
- **\$_POST['username'], \$_POST['password']**: Τα username και password συλλέγονται από τα input fields της φόρμας.
- **\$_SESSION['username']**: Αποθηκεύει το username στη συνεδρία, ώστε να διατηρηθεί η κατάσταση σύνδεσης του χρήστη (το όνομα) και μεταξύ άλλων σελίδων.

5. Εκτέλεση της Stored Procedure “login”

```
$stmt = $pdo->prepare("CALL login(?, ?, @ptype)");  
$stmt->bindParam(1, $username);  
$stmt->bindParam(2, $password);  
$stmt->execute();
```

- **\$pdo->prepare()**: Προετοιμάζει ένα SQL statement που καλεί την stored procedure login. Η διαδικασία παίρνει ως είσοδο τα username και password και επιστρέφει το είδος χρήστη (ptype).
- **bindParam()**: Συνδέει το username και το password με τα placeholders στο SQL query.
- **\$stmt->execute()**: Εκτελεί το SQL statement, καλώντας την stored procedure login.

6. Ανάκτηση του Είδους Χρήστη

```
$stmt->closeCursor();  
  
$result = $pdo->query("SELECT @ptype AS ptype");  
  
$row = $result->fetch(PDO::FETCH_ASSOC);
```

- **closeCursor()**: αναγκαία εντολή για να προετοιμάσω διαφορετικό query.
- **\$pdo->query()**: Εκτελεί ένα query για να ανακτήσει την τιμή της επιστρεφόμενης (από την stored procedure) παραμέτρου ptype. Στη μεταβλητή \$result επιστρέφεται το result set που σχετίζεται με το PDOStatement αντικείμενο (δες PHP documentation).
- **fetch(PDO::FETCH_ASSOC)**: Αποθηκεύει τα δεδομένα του result set στη μεταβλητή \$row υπό τη μορφή ενός associative array (δες documentation).

7. Αυθεντικοποίηση χρήστη

```
if ($row['ptype'] == 'USER' || $row['ptype'] == 'ADMIN') {  
    $_SESSION['user_type'] = $row['ptype'];  
    header("Location: main_page.php");  
    exit();  
} else {  
    $error = "Wrong combination, please try again.";  
}
```

- Εάν το είδος χρήστη είναι 'USER' ή 'ADMIN', ο χρήστης ανακατευθύνεται στην κύρια σελίδα(main_page.php) και το είδος χρήστη αποθηκεύεται στο τρέχον session.
- Σε κάθε άλλη περίπτωση δεν είναι επιτυχής η σύνδεση του χρήστη με τα συγκεκριμένα credentials και επιδεικνύεται μήνυμα λάθους.

8. Error Handling

```
} catch (PDOException $e) {  
    echo "Connection failed: " . $e->getMessage();  
}
```

- Εάν κατά τη διάρκεια της σύνδεσης ή της εκτέλεσης των queries συμβεί Exception, αυτό το block κώδικα επιδεικνύει ένα μήνυμα λάθους σχετικό με το εκάστοτε exception.

HTML Δομή

Η φόρμα HTML επιτρέπει στους χρήστες να εισάγουν το όνομα χρήστη και τον κωδικό πρόσβασης. Μετά την υποβολή, τα δεδομένα αποστέλλονται στο login.php script για επικύρωση.

Φόρμα Login

```
<form action="" method="post">

    <label for="username">Username:</label>

    <input type="text" id="username" name="username" value="<?php echo
htmlspecialchars($username); ?>" required>

    <label for="password">Password:</label>

    <input type="password" id="password" name="password" value="<?php echo
htmlspecialchars($password); ?>" required>

    <div class="button-group">

        <input type="submit" value="Login">

        <a href="insert.php" class="signup-btn">Sign Up</a>

    </div>

</form>
```

- Η φόρμα διαθέτει input fields για τα username και password και buttons για την υποβολή της φόρμας(login) και εγγραφής(sign up) για δημιουργία καινούργιου λογαριασμού.

main_page.php - Main Page After Login

Το script main_page.php εμφανίζει μια σελίδα καλωσορίσματος για τον συνδεδεμένο χρήστη. Περιλαμβάνει ένα μενού πλοήγησης με επιλογές για επεξεργασία πληροφοριών χρήστη, διαγραφή του λογαριασμού και αποσύνδεση.

1. Έναρξη Συνεδρίας και Έλεγχος Login

```
session_start();

if (!isset($_SESSION['username'])) {
    header("Location: login.php");
    exit();
}
```

- **session_start():** έναρξη/συνέχιση συνεδρίας
- **if (!isset(\$_SESSION['username'])):** Ελέγχει αν ο χρήστης είναι συνδεδεμένος επαληθεύοντας την ύπαρξη της μεταβλητής συνεδρίας username. Εάν ο χρήστης δεν είναι συνδεδεμένος, ανακατευθύνεται στη σελίδα login.php.

2. Σύνδεση με τη Βάση Δεδομένων

```
$dsn = "mysql:host=localhost;port=3306;dbname=html";
$dbusername = "root";
$dbpassword = "556782340";
```

```
$username = $_SESSION['username'];
```

3. Ανάκτηση όλων των Δεδομένων Χρηστών

```
try {  
    $pdo = new PDO($dsn, $dbusername, $dbpassword);  
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);  
  
    $stmt = $pdo->query("SELECT * FROM users");  
    $users = $stmt->fetchAll(PDO::FETCH_ASSOC);  
} catch (PDOException $e) {  
    echo "Connection failed: " . $e->getMessage();  
    exit;  
}
```

- Αυτό το μπλοκ κώδικα συνδέεται με τη βάση δεδομένων, ρυθμίζει τη λειτουργία σφαλμάτων σε εξαιρέσεις και ανακτά όλες τις γραμμές από τον πίνακα χρηστών χρησιμοποιώντας ένα απλό SQL query. Το αποτέλεσμα αποθηκεύεται ως associative array στην μεταβλητή **\$users**.

HTML Δομή

1. Μενού Πλοήγησης

```
<header>  
    <nav class="nav-menu">  
        <a href="main_page.php">Home</a>
```



```

<div class="dropdown">
    <a href="javascript:void(0)">Account</a>
    <div class="dropdown-content">
        <a href="edit_info.php">Edit Info</a>
        <a href="javascript:void(0)"
onclick="confirmDelete()">Delete Account</a>
    </div>
</div>
<a href="logout.php">Logout</a>
</nav>
</header>

```

- Η μπάρα πλοήγησης περιλαμβάνει σύνδεσμο προς την αρχική σελίδα, ενέργειες σχετικές με τον λογαριασμό σε μορφή αναδυόμενου μενού (επεξεργασία πληροφοριών, διαγραφή λογαριασμού) και μια επιλογή αποσύνδεσης.

2. JavaScript για Επιβεβαίωση Διαγραφής Λογαριασμού

```

<script>
function confirmDelete() {
    if (confirm("Are you sure?")) {
        window.location.href = "delete_account.php";
    }
}
</script>

```

- Παρέχει ένα διάλογο επιβεβαίωσης πριν επιτρέψει στον χρήστη να διαγράψει τον λογαριασμό του. Εάν ο χρήστης επιβεβαιώσει, ανακατευθύνεται στη σελίδα `delete_account.php`.
-

`insert.php`

Το αρχείο `insert.php` διαχειρίζεται την εγγραφή χρηστών στην εφαρμογή. Παρέχει μια HTML φόρμα για είσοδο δεδομένων και αλληλεπιδρά με τη βάση δεδομένων για την αποθήκευση νέων πληροφοριών χρηστών, διασφαλίζοντας παράλληλα ότι τηρούνται οι περιορισμοί για το όνομα χρήστη και το email.

1. Sign-Up Φόρμα

```
<form action="insert.php" method="POST">
    <input type="text" name="username" placeholder="Username" required>
    <input type="password" name="password" placeholder="Password" required>
    <input type="text" name="firstname" placeholder="First Name" required>
    <input type="text" name="lastname" placeholder="Last Name" required>
    <input type="email" name="email" placeholder="Email" required>
    <input type="submit" value="Sign Up">
</form>
```

- ✓ **Η φόρμα χρησιμοποιεί τη μέθοδο POST** για να υποβάλει δεδομένα, τα οποία αποστέλλονται πίσω στο ίδιο script insert.php για επεξεργασία.

Ανάκτηση Δεδομένων Φόρμας:

```
$username = $_POST['username'];
$password = $_POST['password'];
$firstname = $_POST['firstname'];
$lastname = $_POST['lastname'];
$email = $_POST['email'];
```

Έλεγχος αν υπάρχουν ήδη τα username και email:

- Κλήση δύο stored procedures:
 - **name_taken:** Τσεκάρει εάν το username υπάρχει ήδη.

- **email_taken:** Τσεκάρει εάν το email χρησιμοποιείται ήδη.
- Και αν υπάρχουν ήδη, επιδεικνύονται μηνύματα λάθους για την κάθε περίπτωση (\$username_error, \$email_error).

2. Εγγραφή Χρήστη

Εάν ΚΑΙ το όνομα χρήστη ΚΑΙ το email είναι διαθέσιμα(δεν υπάρχουν ήδη), το script εισάγει τον νέο χρήστη στη βάση δεδομένων:

```
$stmt = $pdo->prepare("INSERT INTO users (username, password, firstname, lastname, email) VALUES (?, ?, ?, ?, ?)");
```

```
$stmt->execute([$username, $password, $firstname, $lastname, $email]);
```

- Μετά την επιτυχή εγγραφή, εκτελεί μια JavaScript συνάρτηση **showPopup()** για να εμφανίσει ένα μήνυμα επιτυχίας και να ανακατευθύνει τον χρήστη στη σελίδα σύνδεσης (login.php).