**✅ create\_bookings\_table.php – Ανάλυση γραμμή-γραμμή**

<?php

🔹 Ξεκινάει το PHP block. Ό,τι είναι ανάμεσα σε <?php ... ?> ερμηνεύεται ως PHP κώδικας.

include("db\_con.php");

🔹 Συμπεριλαμβάνει το αρχείο db\_con.php, το οποίο περιέχει πιθανότατα τις ρυθμίσεις σύνδεσης με τη βάση δεδομένων. Έτσι, δεν χρειάζεται να γράφουμε τον ίδιο κώδικα σύνδεσης ξανά.

$sql = "CREATE TABLE IF NOT EXISTS bookings (

🔹 Ξεκινάει η δημιουργία ενός πίνακα SQL με το όνομα bookings, **μόνο αν δεν υπάρχει ήδη** (IF NOT EXISTS).

id INT(6) UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

🔹 Ορίζει το πεδίο id ως ακέραιο 6 ψηφίων, χωρίς αρνητικές τιμές (UNSIGNED), το οποίο **αυξάνεται αυτόματα** με κάθε νέα καταχώριση και είναι το **πρωτεύον κλειδί**.

name VARCHAR(30) NOT NULL,

email VARCHAR(50),

🔹 name: Κείμενο έως 30 χαρακτήρες, **υποχρεωτικό**.
🔹 email: Κείμενο έως 50 χαρακτήρες, **προαιρετικό** (αφού δεν έχει NOT NULL).

lesson\_type VARCHAR(30),

date DATE,

time TIME

)";

🔹 lesson\_type: Είδος μαθήματος (π.χ. θεωρητικό/πρακτικό) – έως 30 χαρακτήρες.
🔹 date: ημερομηνία κράτησης.
🔹 time: ώρα κράτησης.
🔹 Το κλείσιμο της εντολής SQL με ), και τέλος το "; δηλώνει το τέλος της εντολής.

if ($conn->query($sql) === TRUE) {

🔹 Εκτελεί το SQL ερώτημα. Αν **εκτελεστεί με επιτυχία**, τότε:

echo "Table bookings created successfully";

🔹 Εμφανίζεται μήνυμα ότι ο πίνακας δημιουργήθηκε επιτυχώς.

} else {

echo "Error creating table: " . $conn->error;

}

🔹 Αν **υπάρξει σφάλμα**, εμφανίζεται σχετικό μήνυμα μαζί με το σφάλμα της βάσης.

$conn->close();

🔹 Κλείνει η σύνδεση με τη βάση δεδομένων.

**✅ create\_users\_table.php – Ανάλυση γραμμή-γραμμή**

<?php

🔹 Έναρξη PHP block. Οτιδήποτε ακολουθεί μέχρι το ?> ερμηνεύεται από τον PHP διερμηνευτή.

include("db\_con.php");

🔹 Εισάγεται ο εξωτερικός κώδικας από το αρχείο db\_con.php που περιέχει τη **σύνδεση με τη βάση δεδομένων** ($conn).

$sql = "CREATE TABLE IF NOT EXISTS users (

🔹 Δημιουργείται SQL εντολή για δημιουργία πίνακα με όνομα users, μόνο αν **δεν υπάρχει ήδη**.

id INT(6) UNSIGNED AUTO\_INCREMENT PRIMARY KEY,

🔹 Πρωτεύον κλειδί id: Αυτόματη αύξηση (AUTO\_INCREMENT), ακέραιος θετικός αριθμός έως 6 ψηφία, μοναδικός.

username VARCHAR(30) NOT NULL,

password VARCHAR(255) NOT NULL,

🔹 username: Το όνομα χρήστη (μέγιστο 30 χαρακτήρες), υποχρεωτικό.
🔹 password: Ο κωδικός χρήστη (μέχρι 255 χαρακτήρες – έτσι αποθηκεύονται τα hash των κωδικών με ασφάλεια), επίσης υποχρεωτικό.

role VARCHAR(20) DEFAULT 'user'

)";

🔹 role: Ο ρόλος του χρήστη (π.χ. user, admin). Προεπιλεγμένη τιμή: 'user'.

if ($conn->query($sql) === TRUE) {

🔹 Εκτελεί την εντολή SQL με τη μέθοδο query. Αν πετύχει:

echo "Table users created successfully";

🔹 Εμφανίζει μήνυμα επιτυχίας.

} else {

echo "Error creating table: " . $conn->error;

}

🔹 Αν υπάρξει σφάλμα κατά την εκτέλεση, το εμφανίζει μαζί με το μήνυμα λάθους της βάσης.

$conn->close();

🔹 Κλείνει η σύνδεση με τη βάση δεδομένων.

**✅ db\_con.php – Ανάλυση γραμμή-γραμμή**

<?php

🔹 Έναρξη PHP block.

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "drivingschool";

🔹 Δημιουργούνται 4 μεταβλητές με τα στοιχεία σύνδεσης στη βάση δεδομένων:

* localhost: ο server της βάσης (τρέχει τοπικά).
* root: το προεπιλεγμένο username για τοπικό MySQL/MariaDB.
* Κενό password: για τον root, τοπικά συχνά δεν έχει οριστεί κωδικός.
* drivingschool: το όνομα της βάσης που θα χρησιμοποιηθεί.

$conn = new mysqli($servername, $username, $password, $dbname);

🔹 Δημιουργείται αντικείμενο mysqli για σύνδεση με βάση δεδομένων, χρησιμοποιώντας τις παραπάνω μεταβλητές.
Το αποτέλεσμα αποθηκεύεται στη μεταβλητή $conn.

if ($conn->connect\_error) {

🔹 Έλεγχος για αποτυχία σύνδεσης. Η ιδιότητα connect\_error περιέχει το σφάλμα (αν υπάρχει).

die("Connection failed: " . $conn->connect\_error);

🔹 Αν η σύνδεση αποτύχει, η εκτέλεση του script σταματά με die() και εμφανίζεται μήνυμα λάθους.

}

🔹 Κλείσιμο του if.

Αυτό το αρχείο είναι κρίσιμο γιατί χρησιμοποιείται σε **όλα τα PHP scripts** που χρειάζονται σύνδεση με τη βάση.

**✅ funcs.php – Ανάλυση γραμμή-γραμμή**

<?php

🔹 Έναρξη PHP block.

session\_start();

🔹 Ξεκινά η συνεδρία χρήστη (session). Αυτό είναι απαραίτητο για να διαχειριζόμαστε μεταβλητές συνεδρίας όπως $\_SESSION['username'] κ.λπ.

function check\_login() {

🔹 Ορισμός μιας συνάρτησης check\_login, η οποία θα ελέγχει αν ο χρήστης είναι **συνδεδεμένος**.

if (!isset($\_SESSION['username'])) {

🔹 Αν **δεν έχει οριστεί** μεταβλητή session username, δηλαδή ο χρήστης **δεν έχει κάνει login**...

header("Location: login.php");

🔹 …τότε γίνεται **ανακατεύθυνση** στη σελίδα login.

exit();

🔹 Και **τερματίζεται** το script ώστε να μην προχωρήσει περαιτέρω.

}

🔹 Τέλος της if.

}

🔹 Τέλος της συνάρτησης check\_login.

Αυτή η συνάρτηση προορίζεται για να την τοποθετούμε στην αρχή προστατευμένων σελίδων (π.χ. management.php) ώστε να διασφαλίζουμε ότι κάποιος που δεν είναι συνδεδεμένος **δεν μπορεί να τις δει**.

**✅ hello.php – Ανάλυση γραμμή-γραμμή**

<?php

🔹 Έναρξη PHP block.

session\_start();

🔹 Ξεκινά η συνεδρία. Αυτό επιτρέπει την πρόσβαση σε session μεταβλητές όπως το $\_SESSION['username'].

echo "Hello, " . $\_SESSION['username'] . "!";

🔹 Εμφανίζει μήνυμα καλωσορίσματος για τον συνδεδεμένο χρήστη.
Παράδειγμα: Αν η $\_SESSION['username'] είναι "Dimitra", τότε θα εμφανιστεί:

Hello, Dimitra!

⚠️ Αν κάποιος ανοίξει απευθείας αυτή τη σελίδα **χωρίς να έχει κάνει login**, θα προκληθεί **σφάλμα** επειδή η $\_SESSION['username'] δεν θα έχει οριστεί. Καλό θα ήταν να προστεθεί έλεγχος όπως στη check\_login().

**✅ index.php – Ανάλυση γραμμή-γραμμή**

<?php

session\_start();

?>

🔹 Ξεκινάει το PHP block και η **συνεδρία**. Αν έχει ήδη γίνει login, εδώ θα υπάρχουν ενεργές μεταβλητές όπως $\_SESSION['username'].

<!DOCTYPE html>

<html>

<head>

 <title>Welcome</title>

</head>

🔹 Ξεκινά το HTML έγγραφο. Τίτλος σελίδας: **Welcome**.

<body>

 <h1>Καλώς ήρθες στη Σχολή Οδηγών</h1>

🔹 Κύριος τίτλος στη σελίδα. Εμφανίζεται ανεξαρτήτως του αν έχει κάνει login ο χρήστης.

<?php

if (isset($\_SESSION['username'])) {

🔹 Αν έχει γίνει login (υπάρχει μεταβλητή $\_SESSION['username']), τότε:

echo "<p>Συνδεδεμένος ως: " . $\_SESSION['username'] . "</p>";

echo '<p><a href="logout.php">Logout</a></p>';

🔹 Εμφανίζεται:

* Το όνομα του συνδεδεμένου χρήστη
* Ένας σύνδεσμος για αποσύνδεση (logout)

} else {

echo '<p><a href="login.php">Login</a></p>';

}

?>

🔹 Αν **δεν έχει γίνει login**, τότε εμφανίζεται σύνδεσμος για login.

</body>

</html>

🔹 Κλείσιμο της HTML δομής.

Ουσιαστικά, αυτή η σελίδα λειτουργεί σαν **κεντρική σελίδα** που δείχνει αν κάποιος είναι συνδεδεμένος ή όχι, και παρέχει το ανάλογο link (login ή logout).

**✅ login.php – Ανάλυση γραμμή-γραμμή**

<?php

session\_start();

?>

🔹 Έναρξη PHP block και ενεργοποίηση της συνεδρίας (χρησιμοποιείται για το login αργότερα).

<!DOCTYPE html>

<html>

<head>

 <title>Login</title>

</head>

<body>

 <h2>Σύνδεση Χρήστη</h2>

🔹 Ένα απλό HTML interface για login, με τίτλο "Σύνδεση Χρήστη".

 <form method="post" action="login\_a.php">

 Όνομα χρήστη: <input type="text" name="username"><br>

 Κωδικός: <input type="password" name="password"><br>

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

 </form>

</body>

</html>

🔹 Φόρμα σύνδεσης:

* method="post": τα δεδομένα αποστέλλονται μέσω POST.
* action="login\_a.php": όταν πατηθεί το κουμπί, τα δεδομένα στέλνονται στο login\_a.php για έλεγχο.
* username: πεδίο για το όνομα χρήστη.
* password: πεδίο για τον κωδικό.
* submit: κουμπί "Login".

⚠️ Εδώ **δεν γίνεται ακόμα έλεγχος** των στοιχείων. Όλα τα δεδομένα απλώς στέλνονται στο login\_a.php.

**✅ login\_a.php – Ανάλυση γραμμή-γραμμή**

<?php

session\_start();

🔹 Έναρξη PHP και ενεργοποίηση συνεδρίας – απαραίτητο για να ορίσουμε session μεταβλητές μετά το login.

include("db\_con.php");

🔹 Εισάγεται ο κώδικας σύνδεσης με τη βάση δεδομένων από το db\_con.php.

$username = $\_POST['username'];

$password = $\_POST['password'];

🔹 Παίρνει τα δεδομένα που υποβλήθηκαν από τη φόρμα του login.php (μέσω POST) και τα αποθηκεύει σε μεταβλητές:

* username: όνομα χρήστη
* password: κωδικός χρήστη

$sql = "SELECT \* FROM users WHERE username='$username'";

🔹 Δημιουργεί SQL ερώτημα για να βρει στη βάση τον χρήστη με το συγκεκριμένο username.

⚠️ **ΠΡΟΣΟΧΗ**: Η χρήση απευθείας του $username στην SQL εντολή χωρίς έλεγχο ή χρήση prepared statements κάνει την εφαρμογή ευάλωτη σε **SQL Injection**.

$result = $conn->query($sql);

🔹 Εκτελεί το ερώτημα και αποθηκεύει το αποτέλεσμα στη μεταβλητή $result.

if ($result->num\_rows > 0) {

🔹 Αν βρέθηκε τουλάχιστον ένας χρήστης με αυτό το username…

$row = $result->fetch\_assoc();

🔹 Παίρνει τα δεδομένα του χρήστη από τη βάση σε μορφή πίνακα ($row).

if (password\_verify($password, $row['password'])) {

🔹 Ελέγχει αν ο κωδικός που πληκτρολογήθηκε **ταιριάζει** με τον hash που είναι αποθηκευμένος στη βάση.
Η password\_verify είναι ασφαλής μέθοδος για έλεγχο hashed passwords.

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

$\_SESSION['role'] = $row['role'];

🔹 Αν ο κωδικός είναι σωστός:

* Δημιουργούνται session μεταβλητές username και role (π.χ. admin, user) για να αναγνωρίζεται ο χρήστης στην υπόλοιπη εφαρμογή.

header("Location: index.php");

exit();

🔹 Ανακατεύθυνση στην αρχική σελίδα μετά από επιτυχημένο login.

} else {

echo "Λάθος κωδικός.";

}

🔹 Αν ο κωδικός **δεν είναι σωστός**, εμφανίζεται σχετικό μήνυμα.

} else {

echo "Ο χρήστης δεν βρέθηκε.";

}

🔹 Αν δεν βρέθηκε καν χρήστης με αυτό το username, εμφανίζεται άλλο μήνυμα λάθους.

$conn->close();

🔹 Κλείνει η σύνδεση με τη βάση.

**✅ logout.php – Ανάλυση γραμμή-γραμμή**

<?php

session\_start();

🔹 Έναρξη PHP και ενεργοποίηση συνεδρίας. Χρειάζεται για να μπορούμε να τη **διαγράψουμε**.

session\_unset();

🔹 Αφαιρεί **όλες τις μεταβλητές** από τη συνεδρία, π.χ. $\_SESSION['username'], $\_SESSION['role'] κ.λπ.

session\_destroy();

🔹 Καταστρέφει τελείως τη συνεδρία. Ο χρήστης θεωρείται πλέον **αποσυνδεδεμένος**.

header("Location: index.php");

🔹 Μετά την αποσύνδεση, γίνεται **ανακατεύθυνση** στην αρχική σελίδα (index.php).

exit();

🔹 Τερματισμός του script. Βεβαιώνει ότι δεν εκτελείται τίποτα παρακάτω.

Αυτό το αρχείο είναι το πιο σύντομο και απλό: απλώς **κάνει logout** και επιστρέφει τον χρήστη στην αρχική σελίδα.

**✅ management.php – Ανάλυση γραμμή-γραμμή**

<?php

session\_start();

include("funcs.php");

check\_login();

🔹 Ξεκινά το PHP block και:

* Εκκινεί συνεδρία,
* Περιλαμβάνει το funcs.php, και
* Εκτελεί τη check\_login() για να **ελέγξει αν ο χρήστης είναι συνδεδεμένος**. Αν όχι, θα τον ανακατευθύνει στο login.php.

include("db\_con.php");

🔹 Περιλαμβάνει και το αρχείο σύνδεσης με τη βάση.

$sql = "SELECT \* FROM bookings";

$result = $conn->query($sql);

🔹 Κάνει query στη βάση για να πάρει **όλες τις κρατήσεις** από τον πίνακα bookings, και αποθηκεύει τα αποτελέσματα στη μεταβλητή $result.

echo "<h2>Διαχείριση Κρατήσεων</h2>";

🔹 Εμφανίζει τίτλο σε HTML.

if ($result->num\_rows > 0) {

 echo "<table border='1'><tr><th>ID</th><th>Όνομα</th><th>Email</th><th>Τύπος Μαθήματος</th><th>Ημερομηνία</th><th>Ώρα</th></tr>";

🔹 Αν υπάρχουν εγγραφές:

* Δημιουργεί έναν HTML πίνακα
* Ορίζει τις επικεφαλίδες στηλών: ID, Όνομα, Email, Τύπος Μαθήματος, Ημερομηνία, Ώρα.

while($row = $result->fetch\_assoc()) {

 echo "<tr><td>".$row["id"]."</td><td>".$row["name"]."</td><td>".$row["email"]."</td><td>".$row["lesson\_type"]."</td><td>".$row["date"]."</td><td>".$row["time"]."</td></tr>";

}

🔹 Για κάθε εγγραφή/κρατήση:

* Δημιουργεί μια νέα γραμμή στον πίνακα
* Συμπληρώνει τα πεδία με τα δεδομένα από τη βάση (id, name, email, lesson\_type, date, time)

echo "</table>";

} else {

 echo "Δεν υπάρχουν κρατήσεις.";

}

🔹 Αν υπάρχουν κρατήσεις → κλείνει τον πίνακα.
🔹 Αν **δεν υπάρχουν** → εμφανίζει το μήνυμα “Δεν υπάρχουν κρατήσεις.”

$conn->close();

?>

🔹 Κλείνει η σύνδεση με τη βάση και το PHP block.

Αυτό το αρχείο εμφανίζει **όλες τις κρατήσεις** σε πίνακα, **μόνο αν ο χρήστης είναι συνδεδεμένος**. Είναι βασική σελίδα "admin" για διαχείριση ή εποπτεία.