**Step By Step Guide –school\_management (Php and MySQL for deployment to WAMP Server)**

What we discussed a few hour ago

Hey Jagdev! Earlier today, we were exploring your school management system project—specifically the one for Government High School, Chananke. I noticed the page mentions it’s still under construction and credited to your email, so it seems like you’re actively building or refining the platform.

If you were working on a particular feature—like the Adminer login, managing student/teacher records, or integrating it with your SPT-QuizBridge project—I’d be happy to pick up right where we left off. Want to dive back into one of those tasks or show me any updates you’ve made since?

Last time you have given me this code : <?php session\_start(); if ($\_SERVER['REQUEST\_METHOD'] === 'POST') { if ($\_POST['username'] === 'jagdevsinghdosanjh' && $\_POST['password'] === 'your\_password\_here') { $\_SESSION['is\_admin'] = true; header('Location: generate\_tables.php'); exit(); } else { echo "Invalid credentials."; } } ?> <form method="POST"> Username: <input type="text" name="username" required /><br/> Password: <input type="password" name="password" required /><br/> <input type="submit" value="Login as Admin" /> </form> as project\_admin.php

Thanks for the reminder, Jagdev! Yes, that project\_admin.php script sets up a basic admin login using PHP sessions with a hardcoded username and password.

If you're now planning to refine it—maybe add hashed passwords, session timeouts, or even move the credentials to a secure file or database—I can help expand or secure it further. Or if you’re ready to integrate it with the rest of your school management system, like redirecting it to your existing generate\_tables.php workflow and beyond, we can tighten up that whole flow.

Want to make it more secure or connect it to your student progress features next?

Now I need the rights for project admin to create the tables for students users and teacher users. The students and teacher need these tables as credential matching to login to their logged in session pages. Based on their role in a db impose some restrictions on to them as student users and teacher users, Provide the rights to them that the students and teachers are to do after login.

Great step forward, Jagdev! Here's a secure and modular approach to extend your project\_admin.php logic into a full-featured role-based access system using PHP and MySQL:

**🧱 1. Database Design: Suggested Tables**

You’ll need three essential tables:

sql

CREATE TABLE users (

id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(50) UNIQUE NOT NULL,

password\_hash VARCHAR(255) NOT NULL,

role ENUM('student', 'teacher') NOT NULL

);

CREATE TABLE students (

user\_id INT PRIMARY KEY,

name VARCHAR(100),

class VARCHAR(20),

FOREIGN KEY (user\_id) REFERENCES users(id)

);

CREATE TABLE teachers (

user\_id INT PRIMARY KEY,

name VARCHAR(100),

subject VARCHAR(100),

FOREIGN KEY (user\_id) REFERENCES users(id)

);

Your **admin interface** will create users, assign roles, and populate the appropriate secondary table.

**🔐 2. Admin Rights (project\_admin.php)**

Once logged in, allow the admin to:

* Create users with roles.
* Securely hash passwords using password\_hash($\_POST['password'], PASSWORD\_DEFAULT).
* Populate student/teacher details.

Want me to write a sample PHP form and handler for creating users based on role?

**🧭 3. Login Script (shared for students & teachers)**

After authentication, check the user’s role and redirect:

php

$\_SESSION['user\_id'] = $user['id'];

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

if ($user['role'] === 'student') {

header("Location: student\_dashboard.php");

} elseif ($user['role'] === 'teacher') {

header("Location: teacher\_dashboard.php");

}

**🎯 4. Role-Based Restrictions**

On each dashboard (student\_dashboard.php, teacher\_dashboard.php), start with a check:

php

session\_start();

if ($\_SESSION['role'] !== 'student') {

header("Location: unauthorized.php");

exit();

}

**Students can:**

* View assigned classes.
* Check marks/progress.
* View announcements.

**Teachers can:**

* Upload marks.
* View assigned classes/students.
* Post announcements.

Want to see the full working PHP scripts for creating users or login logic? Or maybe a basic UI template for both dashboards?

Create files with complete codes as per these requirements

Absolutely, Jagdev! Let's build a secure, role-based login system for your school management platform. I’ll walk you through a complete setup using PHP and MySQL, organized into modular files:

### 📁 ****File 1:**** db.php – Database Connection

php

<?php

$host = 'localhost';

$db = 'school\_management';

$user = 'root';

$pass = ''; // Set your MySQL root password

try {

$pdo = new PDO("mysql:host=$host;dbname=$db", $user, $pass);

$pdo->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

} catch (PDOException $e) {

die("Connection failed: " . $e->getMessage());

}

?>

### 📁 ****File 2:**** project\_admin.php – Admin Login

php

<?php

session\_start();

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

if ($\_POST['username'] === 'jagdevsinghdosanjh' && $\_POST['password'] === 'your\_password\_here') {

$\_SESSION['admin\_logged\_in'] = true;

header('Location: create\_user.php');

exit();

} else {

$error = "Invalid credentials.";

}

}

?>

<form method="POST">

<h2>Admin Login</h2>

Username: <input type="text" name="username" required /><br/>

Password: <input type="password" name="password" required /><br/>

<input type="submit" value="Login as Admin" />

<?php if (!empty($error)) echo "<p>$error</p>"; ?>

</form>

### 📁 ****File 3:**** create\_user.php – Create Student/Teacher Users

php

<?php

session\_start();

if (!isset($\_SESSION['admin\_logged\_in'])) {

header('Location: project\_admin.php');

exit();

}

require 'db.php';

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

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

$password\_hash = password\_hash($\_POST['password'], PASSWORD\_DEFAULT);

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

$stmt = $pdo->prepare("INSERT INTO users (username, password\_hash, role) VALUES (?, ?, ?)");

$stmt->execute([$username, $password\_hash, $role]);

$user\_id = $pdo->lastInsertId();

if ($role === 'student') {

$stmt = $pdo->prepare("INSERT INTO students (user\_id, name, class) VALUES (?, ?, ?)");

$stmt->execute([$user\_id, $\_POST['name'], $\_POST['class']]);

} elseif ($role === 'teacher') {

$stmt = $pdo->prepare("INSERT INTO teachers (user\_id, name, subject) VALUES (?, ?, ?)");

$stmt->execute([$user\_id, $\_POST['name'], $\_POST['subject']]);

}

echo "User created successfully!";

}

?>

<form method="POST">

<h2>Create New User</h2>

Username: <input type="text" name="username" required /><br/>

Password: <input type="password" name="password" required /><br/>

Role:

<select name="role" required onchange="toggleFields(this.value)">

<option value="">Select role</option>

<option value="student">Student</option>

<option value="teacher">Teacher</option>

</select><br/>

<div id="student\_fields" style="display:none">

Name: <input type="text" name="name" /><br/>

Class: <input type="text" name="class" /><br/>

</div>

<div id="teacher\_fields" style="display:none">

Name: <input type="text" name="name" /><br/>

Subject: <input type="text" name="subject" /><br/>

</div>

<input type="submit" value="Create User" />

</form>

<script>

function toggleFields(role) {

document.getElementById('student\_fields').style.display = role === 'student' ? 'block' : 'none';

document.getElementById('teacher\_fields').style.display = role === 'teacher' ? 'block' : 'none';

}

</script>

### 📁 ****File 4:**** login.php – Student/Teacher Login

php

<?php

session\_start();

require 'db.php';

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

$stmt = $pdo->prepare("SELECT \* FROM users WHERE username = ?");

$stmt->execute([$\_POST['username']]);

$user = $stmt->fetch();

if ($user && password\_verify($\_POST['password'], $user['password\_hash'])) {

$\_SESSION['user\_id'] = $user['id'];

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

header("Location: {$user['role']}\_dashboard.php");

exit();

} else {

$error = "Invalid login.";

}

}

?>

<form method="POST">

<h2>User Login</h2>

Username: <input type="text" name="username" required /><br/>

Password: <input type="password" name="password" required /><br/>

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

<?php if (!empty($error)) echo "<p>$error</p>"; ?>

</form>

### 📁 ****File 5:**** student\_dashboard.php

php

<?php

session\_start();

if ($\_SESSION['role'] !== 'student') {

header('Location: unauthorized.php');

exit();

}

echo "<h2>Welcome, Student!</h2>";

echo "You can view your progress, subjects, and announcements here.";

?>

### 📁 ****File 6:**** teacher\_dashboard.php

php

<?php

session\_start();

if ($\_SESSION['role'] !== 'teacher') {

header('Location: unauthorized.php');

exit();

}

echo "<h2>Welcome, Teacher!</h2>";

echo "You can manage student marks, post announcements, and view class lists.";

?>

### 📁 ****File 7:**** unauthorized.php

php

<h2>Access Denied</h2>

<p>You do not have permission to access this page.</p>

<a href="login.php">Return to Login</a>

If you’d like, I can help you set up the database creation script too or integrate this cleanly into your existing SPT-QuizBridge architecture. Just say the word