PHP-based platform that allows student sign-ups/logins, and for admins to manage courses, video lectures, and PDF notes—including CRUD operations (create, read, update, delete).

See following straightforward architecture, point out key security considerations, and provide sample PHP code for each feature.

PHP-based e-learning system from the previous message, follow this step-by-step guide. You'll need to:

⍰ 1. Requirements

Make sure you have the following installed:-

Tool	Purpose
PHP	Runs the backend code
MySQL	Stores data (users, courses, lectures)
Apache	Web server
phpMyAdmin (optional)	Manage MySQL via browser

Simplest Setup: Install XAMPP

XAMPP includes Apache, MySQL, and PHP in one package.

⍰ 2. Set Up the Database

1. Start Apache and MySQL via XAMPP control panel.
2. Open http://localhost/phpmyadmin in your browser.
3. Create a database, e.g., elearning.

---

1. Architecture & Design

a. Database Schema (simplified):-

Run SQL to create tables:

```
CREATE TABLE users (
  id INT AUTO_INCREMENT PRIMARY KEY,
  username VARCHAR(50) UNIQUE NOT NULL,
  password_hash VARCHAR(255) NOT NULL,
  role ENUM('student','admin') NOT NULL
);

CREATE TABLE courses (
  id INT AUTO_INCREMENT PRIMARY KEY,
  title VARCHAR(255) NOT NULL,
  description TEXT
);

CREATE TABLE lectures (
  id INT AUTO_INCREMENT PRIMARY KEY,
  course_id INT NOT NULL,
  title VARCHAR(255) NOT NULL,
  video_path VARCHAR(255),
  pdf_path VARCHAR(255),
  FOREIGN KEY (course_id) REFERENCES courses(id) ON DELETE CASCADE
);
```

b. Flow Summary

- Registration/Login: Students can register and log in; admins must be created directly in the DB or by an existing admin.

- Admin: Can Create/Read/Update/Delete courses and lectures (including file uploads for videos and PDFs).

- Student: Can log in and view course listings and lecture content.

3. Sample PHP Code (Skeleton)

Below is a functional—but minimal—example. You'll need to expand it with proper HTML forms, error handling, routing, and styling.

---

db.php — Database Connection

```php
<?php

session_start();

$dsn = 'mysql:host=localhost;dbname=elearning;charset=utf8mb4';

$pdo = new PDO($dsn, 'root', '', [

    PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,

]);

?>
```

auth.php   Registration & Login code:-

```php
<?php
require 'db.php';

function register($username, $password) {
    global $pdo;
    $hash = password_hash($password, PASSWORD_DEFAULT);
    $stmt = $pdo->prepare("INSERT INTO users (username, password_hash, role) VALUES (?, ?, 'student')");
    return $stmt->execute([$username, $hash]);
}

function login($username, $password) {
    global $pdo;
    $stmt = $pdo->prepare("SELECT id, password_hash, role FROM users WHERE username = ?");
    $stmt->execute([$username]);
    if ($user = $stmt->fetch(PDO::FETCH_ASSOC)) {
        if (password_verify($password, $user['password_hash'])) {
            session_regenerate_id(true);
            $_SESSION['user_id'] = $user['id'];
            $_SESSION['role'] = $user['role'];
            return true;
        }
    }
    return false;
}
```

```
// Usage example:
// if (isset($_POST['register'])) { register($_POST['username'], $_POST['password']); }
// if (isset($_POST['login'])) { login(...)? redirect accordingly }
?>
```

Student register.php file code :-

```
<?php
require 'auth.php'; // the code you shared
$message = '';

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    // Get the message returned by the register function
    $message = register($_POST['username'], $_POST['password']);
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Student Registration</title>
  <style>
    <?php include 'style.css'; ?>
  </style>
</head>
<body>
```

```
  <div class="form-container">
    <h2>Register</h2>
    <?php if ($message): ?><p class="message"><?= $message ?></p><?php endif; ?>
    <form method="POST">
      <input type="text" name="username" placeholder="Username" required>
      <input type="password" name="password" placeholder="Password" required>
      <button type="submit">Register</button>
    </form>
    <p>Already have an account? <a href="login.php">Login</a></p>
  </div>
</body>
</html>
```

admin_register.php file code:-

```
<?php
require 'auth.php';



$message = '';
```

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

    // Manually register with admin role
    global $pdo;
    $hash = password_hash($password, PASSWORD_DEFAULT);
    $stmt = $pdo->prepare("INSERT INTO users (username, password_hash, role) VALUES (?, ?, 'admin')");
    try {
        $stmt->execute([$username, $hash]);
        $message = "New admin created successfully.";
    } catch (PDOException $e) {
        $message = "Error: Username might already exist.";
    }
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Create New Admin</title>
  <style>
    <?php include 'style.css'; ?>
  </style>
</head>
```

```
<body>
  <div class="form-container">
    <h2>Create Admin</h2>
    <?php if ($message): ?><p class="message"><?= $message ?></p><?php endif; ?>
    <form method="POST">
      <input type="text" name="username" placeholder="Admin Username" required>
      <input type="password" name="password" placeholder="Password" required>
      <button type="submit">Create Admin</button>
    </form>
    <p><a href="admin_courses.php">Back to Admin Panel</a></p>
  </div>
</body>
</html>
```

admin_login.php file:-

```
<?php
require 'auth.php';
$message = '';

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $success = login($_POST['username'], $_POST['password']);
    if ($success && $_SESSION['role'] === 'admin') {
        header('Location: admin_courses.php');
        exit;
    } else {
        $message = "Access denied or incorrect credentials.";
    }
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Admin Login</title>
  <style>
    <?php include 'style.css'; ?>
  </style>
</head>
<body>
  <div class="form-container">
    <h2>Admin Login</h2>
    <?php if ($message): ?><p class="message"><?= $message ?></p><?php endif; ?>
    <form method="POST">
      <input type="text" name="username" placeholder="Admin Username" required>
      <input type="password" name="password" placeholder="Password" required>
      <button type="submit">Login</button>
    </form>
    <p><a href="login.php">Back to student login</a></p>
  </div>
</body>
</html>
```

admin_courses.php — Admin: CRUD Courses

```
<?php
require 'db.php';
if ($_SESSION['role'] !== 'admin') { die('Access denied'); }

// Initialize form values
$edit_mode = false;
$form_id = '';
$form_title = '';
$form_description = '';

// Handle Create or Update
```

```
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $id = $_POST['id'] ?? null;
    $title = $_POST['title'];
    $desc = $_POST['description'];

    if ($id) {
        // Update existing
        $stmt = $pdo->prepare("UPDATE courses SET title=?, description=? WHERE id=?");
        $stmt->execute([$title, $desc, $id]);
    } else {
        // Create new
        $stmt = $pdo->prepare("INSERT INTO courses (title, description) VALUES (?, ?)");
        $stmt->execute([$title, $desc]);
    }
    header('Location: admin_courses.php');
    exit;
}

// Handle Delete
if (isset($_GET['delete_id'])) {
    $stmt = $pdo->prepare("DELETE FROM courses WHERE id=?");
    $stmt->execute([$_GET['delete_id']]);
    header('Location: admin_courses.php');
    exit;
}
```

```
// Handle Edit

if (isset($_GET['edit_id'])) {

    $stmt = $pdo->prepare("SELECT * FROM courses WHERE id=?");

    $stmt->execute([$_GET['edit_id']]);

    $course = $stmt->fetch(PDO::FETCH_ASSOC);

    if ($course) {

        $edit_mode = true;

        $form_id = $course['id'];

        $form_title = $course['title'];

        $form_description = $course['description'];

    }

}


// List courses

$courses = $pdo->query("SELECT * FROM courses")->fetchAll(PDO::FETCH_ASSOC);

?>


<!DOCTYPE html>

<html>

<head>

  <title>Manage Courses</title>

  <style>

    body { font-family: Arial; padding: 20px; background: #f9f9f9; }

    form, ul { margin-top: 20px; }
    input, textarea { width: 100%; padding: 8px; margin: 6px 0; }
    button { padding: 10px 15px; background: #007bff; color: white; border: none; cursor: pointer; }
    button:hover { background: #0056b3; }
    .course-item { margin: 10px 0; }
    a { margin-left: 10px; color: #007bff; text-decoration: none; }
    a:hover { text-decoration: underline; }
  </style>
</head>
<body>

<h2><?= $edit_mode ? 'Edit Course' : 'Add New Course' ?></h2>

<form method="post">
  <input type="hidden" name="id" value="<?= htmlspecialchars($form_id) ?>">
  <input name="title" placeholder="Course Title" value="<?= htmlspecialchars($form_title) ?>" required>
  <textarea name="description" placeholder="Course Description" required><?= htmlspecialchars($form_description) ?></textarea>
  <button type="submit"><?= $edit_mode ? 'Update' : 'Create' ?> Course</button>
  <?php if ($edit_mode): ?>
    <a href="admin_courses.php">Cancel Edit</a>
  <?php endif; ?>
</form>

<h2>All Courses</h2>
<ul>
```

```
<?php foreach ($courses as $c): ?>
  <li class="course-item">
    <strong><?= htmlspecialchars($c['title']) ?></strong><br>
    <?= nl2br(htmlspecialchars($c['description'])) ?><br>
    <a href="?edit_id=<?= $c['id'] ?>">✏□ Edit</a>
    <a href="?delete_id=<?= $c['id'] ?>" onclick="return confirm('Delete this course?')">□ Delete</a>
  </li>
<?php endforeach; ?>
</ul>

</body>
</html>
```

admin_lectures.php — Admin: CRUD Lectures with File Uploads

```
<?php
require 'db.php';
if ($_SESSION['role'] !== 'admin') die('Access denied');

// --- Handle Create or Update ---
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
```

```
$lecture_id = $_POST['lecture_id'] ?? null;

$course_id = $_POST['course_id'];

$title = $_POST['title'];


// Get existing lecture if updating

$videoPath = $pdfPath = null;

if ($lecture_id) {

    $stmt = $pdo->prepare("SELECT * FROM lectures WHERE id = ?");

    $stmt->execute([$lecture_id]);

    $existing = $stmt->fetch(PDO::FETCH_ASSOC);

    $videoPath = $existing['video_path'];

    $pdfPath = $existing['pdf_path'];

}


// Handle new file uploads

if (!empty($_FILES['video']['tmp_name'])) {

    $vname = uniqid() . '_' . basename($_FILES['video']['name']);

    move_uploaded_file($_FILES['video']['tmp_name'], 'uploads/videos/' . $vname);

    $videoPath = 'uploads/videos/' . $vname;

}

if (!empty($_FILES['pdf']['tmp_name'])) {

    $pname = uniqid() . '_' . basename($_FILES['pdf']['name']);

    move_uploaded_file($_FILES['pdf']['tmp_name'], 'uploads/pdfs/' . $pname);

    $pdfPath = 'uploads/pdfs/' . $pname;

}
```

```
    if ($lecture_id) {
        // UPDATE
        $stmt = $pdo->prepare("UPDATE lectures SET course_id=?, title=?, video_path=?, pdf_path=? WHERE id=?");
        $stmt->execute([$course_id, $title, $videoPath, $pdfPath, $lecture_id]);
    } else {
        // CREATE
        $stmt = $pdo->prepare("INSERT INTO lectures (course_id, title, video_path, pdf_path) VALUES (?, ?, ?, ?)");
        $stmt->execute([$course_id, $title, $videoPath, $pdfPath]);
    }

    header('Location: admin_lectures.php');
    exit;
}

// --- Handle Delete ---
if (isset($_GET['delete_id'])) {
    $stmt = $pdo->prepare("DELETE FROM lectures WHERE id = ?");
    $stmt->execute([$_GET['delete_id']]);
    header('Location: admin_lectures.php');
    exit;
}

// --- Handle Edit Mode ---
```

```
$edit_mode = false;
$lecture_form = ['id' => '', 'course_id' => '', 'title' => '', 'video_path' => '', 'pdf_path' => ''];

if (isset($_GET['edit_id'])) {
    $stmt = $pdo->prepare("SELECT * FROM lectures WHERE id = ?");
    $stmt->execute([$_GET['edit_id']]);
    $lecture_form = $stmt->fetch(PDO::FETCH_ASSOC);
    $edit_mode = true;
}

// --- Fetch courses and lectures ---
$courses = $pdo->query("SELECT * FROM courses")->fetchAll(PDO::FETCH_ASSOC);
$lectures = $pdo->query("
  SELECT l.*, c.title AS course_title
  FROM lectures l
  JOIN courses c ON l.course_id = c.id
  ORDER BY l.id DESC
")->fetchAll(PDO::FETCH_ASSOC);
?>

<!DOCTYPE html>
<html>
<head>
    <title>Manage Lectures</title>
    <style>
```

```
        body { font-family: Arial; padding: 20px; background: #f7f7f7; }

        form { margin-bottom: 30px; background: #fff; padding: 20px; border-radius: 8px; }

        input, select, textarea { display: block; width: 100%; margin-top: 10px; padding: 8px; }

        button { margin-top: 10px; padding: 10px 15px; background: #007bff; color: white; border: none; border-radius: 4px; }

        button:hover { background: #0056b3; }

        .lecture-list { background: #fff; padding: 20px; border-radius: 8px; }

        .lecture-item { margin-bottom: 15px; border-bottom: 1px solid #ddd; padding-bottom: 10px; }

        .actions a { margin-right: 10px; text-decoration: none; color: #007bff; }

        .actions a:hover { text-decoration: underline; }

    </style>

</head>

<body>



<h2><?= $edit_mode ? 'Edit Lecture' : 'Add New Lecture' ?></h2>

<form method="post" enctype="multipart/form-data">

    <input type="hidden" name="lecture_id" value="<?= htmlspecialchars($lecture_form['id']) ?>">



    <label>Course:</label>

    <select name="course_id" required>

        <option value="">-- Select Course --</option>

        <?php foreach ($courses as $c): ?>

            <option value="<?= $c['id'] ?>" <?= $c['id'] == $lecture_form['course_id'] ? 'selected' : '' ?>>

                <?= htmlspecialchars($c['title']) ?>

            </option>

        <?php endforeach; ?>
    </select>

    <label>Lecture Title:</label>
    <input name="title" value="<?= htmlspecialchars($lecture_form['title']) ?>" required>

    <label>Video File: <?= $lecture_form['video_path'] ? '(Leave blank to keep existing)' : '' ?></label>
    <input type="file" name="video" accept="video/*">

    <label>PDF Notes: <?= $lecture_form['pdf_path'] ? '(Leave blank to keep existing)' : '' ?></label>
    <input type="file" name="pdf" accept="application/pdf">

    <button type="submit"><?= $edit_mode ? 'Update' : 'Create' ?> Lecture</button>
    <?php if ($edit_mode): ?>
        <a href="admin_lectures.php">Cancel Edit</a>
    <?php endif; ?>
</form>

<div class="lecture-list">
    <h2>All Lectures</h2>
    <?php foreach ($lectures as $l): ?>
        <div class="lecture-item">
            <strong><?= htmlspecialchars($l['course_title']) ?> - <?= htmlspecialchars($l['title']) ?></strong><br>
            <?php /* if ($l['video_path']): ?>
                ⍰ <a href="<?= htmlspecialchars($l['video_path']) ?>" target="_blank">Video</a>
            <?php endif; ?>
        <?php if ($l['pdf_path']): ?>
           ⍰ <a href="<?= htmlspecialchars($l['pdf_path']) ?>" target="_blank">PDF</a>
        <?php endif; */ ?>

                                <a href="view_lecture.php?id=<?= $l['id'] ?>">⍰ View Lecture</a>
        <div class="actions">
           <a href="?edit_id=<?= $l['id'] ?>">✏□ Edit</a>
           <a href="?delete_id=<?= $l['id'] ?>" onclick="return confirm('Delete this lecture?')">□ Delete</a>
        </div>
      </div>
   <?php endforeach; ?>
</div>

</body>
</html>
```

student_view.php — Student: View Courses & Lectures

```
<?php
```

```
require 'db.php';


// Optionally restrict access
if (!isset($_SESSION['role'])) {
    die("Access denied.");
}

// Fetch all courses
$courses = $pdo->query("SELECT * FROM courses")->fetchAll(PDO::FETCH_ASSOC);

// Prepare lecture data for all courses
$course_lectures = [];
$stmt = $pdo->prepare("SELECT * FROM lectures WHERE course_id = ?");
foreach ($courses as $course) {
    $stmt->execute([$course['id']]);
    $course_lectures[$course['id']] = $stmt->fetchAll(PDO::FETCH_ASSOC);
}
?>

<!DOCTYPE html>
<html>
<head>
    <title>Courses and Lectures</title>
    <style>
    body { font-family: Arial; padding: 20px; background: #f9f9f9; }
    .course { background: #fff; padding: 20px; margin-bottom: 20px; border-radius: 6px; }
    .lecture { margin-left: 20px; padding: 10px 0; border-top: 1px solid #ddd; }
    .lecture a { margin-right: 10px; color: #007bff; text-decoration: none; }
    .lecture a:hover { text-decoration: underline; }
  </style>
</head>
<body>

<h1>Courses and Lectures</h1>

<?php foreach ($courses as $course): ?>
  <div class="course">
    <h2><?= htmlspecialchars($course['title']) ?></h2>
    <p><?= nl2br(htmlspecialchars($course['description'])) ?></p>

    <?php if (!empty($course_lectures[$course['id']])): ?>
      <?php foreach ($course_lectures[$course['id']] as $lecture): ?>
        <div class="lecture">
          <strong><?= htmlspecialchars($lecture['title']) ?></strong><br>
          <a href="view_lecture1.php?id=<?= $lecture['id'] ?>">▯ View Lecture</a>
        </div>
      <?php endforeach; ?>
    <?php else: ?>
      <p><em>No lectures added yet.</em></p>
```

```
        <?php endif; ?>
    </div>
<?php endforeach; ?>

</body>
</html>
```

view_lecture1.php (student view lecture):-

```
<?php
require 'db.php';
if (!isset($_SESSION['role']) || $_SESSION['role'] !== 'student') {
    die('Access denied');
}



if (!isset($_GET['id']) || !is_numeric($_GET['id'])) {
    die("Lecture ID missing or invalid.");
}

$lecture_id = (int)$_GET['id'];
```

```
$stmt = $pdo->prepare("
    SELECT l.*, c.title AS course_title
    FROM lectures l
    JOIN courses c ON l.course_id = c.id
    WHERE l.id = ?
");
$stmt->execute([$lecture_id]);
$lecture = $stmt->fetch(PDO::FETCH_ASSOC);

if (!$lecture) {
    die("Lecture not found.");
}
?>

<!DOCTYPE html>
<html>
<head>
    <title>View Lecture: <?= htmlspecialchars($lecture['title']) ?></title>
    <style>
        body { font-family: Arial; padding: 20px; background: #f4f4f4; }
        h1, h2, h3 { color: #333; }
        .content { background: white; padding: 20px; border-radius: 8px; max-width: 900px; margin: auto; }
        video, iframe { width: 100%; max-width: 800px; height: 400px; margin-top: 15px; border: 1px solid #ddd; border-radius: 4px; }
        a.back { display: inline-block; margin-top: 20px; text-decoration: none; color: #007bff; }
        a.back:hover { text-decoration: underline; }
    </style>
</head>
<body>

<div class="content">
    <h1><?= htmlspecialchars($lecture['title']) ?></h1>
    <h3>Course: <?= htmlspecialchars($lecture['course_title']) ?></h3>

    <?php if (!empty($lecture['video_path'])): ?>
        <h2>Video Lecture</h2>
        <video controls>
            <source src="<?= htmlspecialchars($lecture['video_path']) ?>" type="video/mp4">
            Your browser does not support the video tag.
        </video>
    <?php endif; ?>

    <?php if (!empty($lecture['pdf_path'])): ?>
        <h2>PDF Notes</h2>
        <iframe src="<?= htmlspecialchars($lecture['pdf_path']) ?>"></iframe>
    <?php endif; ?>

    <a class="back" href="admin_lectures.php">← Back to Lectures</a>
</div>
```

```
</body>
</html>
```

Admin side view_lecture.php file code :-

```
<?php
require 'db.php';
if (!isset($_SESSION['role']) || $_SESSION['role'] !== 'admin') {
    die('Access denied');
}



if (!isset($_GET['id']) || !is_numeric($_GET['id'])) {
    die("Lecture ID missing or invalid.");
}

$lecture_id = (int)$_GET['id'];

$stmt = $pdo->prepare("
    SELECT l.*, c.title AS course_title
    FROM lectures l
    JOIN courses c ON l.course_id = c.id
    WHERE l.id = ?
");
```

```
$stmt->execute([$lecture_id]);
$lecture = $stmt->fetch(PDO::FETCH_ASSOC);

if (!$lecture) {
    die("Lecture not found.");
}
?>

<!DOCTYPE html>
<html>
<head>
    <title>View Lecture: <?= htmlspecialchars($lecture['title']) ?></title>
    <style>
        body { font-family: Arial; padding: 20px; background: #f4f4f4; }
        h1, h2, h3 { color: #333; }
        .content { background: white; padding: 20px; border-radius: 8px; max-width: 900px; margin: auto; }
        video, iframe { width: 100%; max-width: 800px; height: 400px; margin-top: 15px; border: 1px solid #ddd; border-radius: 4px; }
        a.back { display: inline-block; margin-top: 20px; text-decoration: none; color: #007bff; }
        a.back:hover { text-decoration: underline; }
    </style>
</head>
<body>

<div class="content">
    <h1><?= htmlspecialchars($lecture['title']) ?></h1>
```

```
  <h3>Course: <?= htmlspecialchars($lecture['course_title']) ?></h3>

  <?php if (!empty($lecture['video_path'])): ?>
    <h2>Video Lecture</h2>
    <video controls>
      <source src="<?= htmlspecialchars($lecture['video_path']) ?>" type="video/mp4">
      Your browser does not support the video tag.
    </video>
  <?php endif; ?>

  <?php if (!empty($lecture['pdf_path'])): ?>
    <h2>PDF Notes</h2>
    <iframe src="<?= htmlspecialchars($lecture['pdf_path']) ?>"></iframe>
  <?php endif; ?>

  <a class="back" href="admin_lectures.php">← Back to Lectures</a>
</div>

</body>
</html>
```

Here's the full code for admin_assign_courses.php, a simple admin panel to assign/unassign courses to students.

⍰ Features of admin_assign_courses.php:

Admin can:

⍰ See list of students.

⍰ Assign courses to each student using checkboxes.

⍰ View and edit current course assignments.

Fully uses student_courses table for assignments.

⍰ Prerequisites

Ensure the following table exists:

sql

```
CREATE TABLE student_courses (
  student_id INT NOT NULL,
  course_id INT NOT NULL,
  PRIMARY KEY (student_id, course_id),
  FOREIGN KEY (student_id) REFERENCES users(id) ON DELETE CASCADE,
  FOREIGN KEY (course_id) REFERENCES courses(id) ON DELETE CASCADE
);
```

⍰ Full Code: admin_assign_courses.php

```php
<?php
require 'db.php';

if (!isset($_SESSION['role']) || $_SESSION['role'] !== 'admin') {
    die('Access denied');
}

// Handle form submission
if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['student_id'])) {
    $student_id = (int)$_POST['student_id'];

    // Remove existing assignments
    $stmt = $pdo->prepare("DELETE FROM student_courses WHERE student_id = ?");
    $stmt->execute([$student_id]);

    // Add new assignments
    if (isset($_POST['course_ids']) && is_array($_POST['course_ids'])) {
        $stmt = $pdo->prepare("INSERT INTO student_courses (student_id, course_id) VALUES (?, ?)");
        foreach ($_POST['course_ids'] as $course_id) {
            $stmt->execute([$student_id, (int)$course_id]);
        }
    }

    $message = "Courses updated successfully for the student.";
}

// Fetch students (only students, not admins)
$students = $pdo->query("SELECT * FROM users WHERE role = 'student'")-
>fetchAll(PDO::FETCH_ASSOC);

// Fetch all courses
$courses = $pdo->query("SELECT * FROM courses")->fetchAll(PDO::FETCH_ASSOC);

// If a student is selected (for editing)
$selected_student_id = isset($_GET['student_id']) ? (int)$_GET['student_id'] : null;
$assigned_courses = [];

if ($selected_student_id) {
   $stmt = $pdo->prepare("SELECT course_id FROM student_courses WHERE student_id = ?");
   $stmt->execute([$selected_student_id]);
   $assigned_courses = $stmt->fetchAll(PDO::FETCH_COLUMN);
}
?>

<!DOCTYPE html>
<html>
<head>
   <title>Assign Courses to Students</title>
   <style>
      body { font-family: Arial, sans-serif; padding: 20px; background: #f5f5f5; }
```

```
        .container { max-width: 800px; margin: auto; background: white; padding: 20px; border-radius: 8px; box-shadow: 0 2px 6px rgba(0,0,0,0.1); }
        h2 { margin-bottom: 20px; }
        select, button { padding: 10px; margin: 10px 0; width: 100%; }
        .checkboxes label { display: block; margin: 5px 0; }
        .message { background: #d4edda; color: #155724; padding: 10px; border-radius: 4px; margin-bottom: 20px; }
        .back-link { display: inline-block; margin-top: 20px; text-decoration: none; color: #007bff; }
        .back-link:hover { text-decoration: underline; }
    </style>
</head>
<body>

<div class="container">
    <h2>Assign Courses to Students</h2>

    <?php if (isset($message)): ?>
        <div class="message"><?= htmlspecialchars($message) ?></div>
    <?php endif; ?>

    <form method="get" action="">
        <label for="student_id">Select Student:</label>
        <select name="student_id" id="student_id" onchange="this.form.submit()">
            <option value="">-- Choose a student --</option>
            <?php foreach ($students as $student): ?>
                <option value="<?= $student['id'] ?>" <?= $selected_student_id == $student['id'] ? 'selected' : '' ?>>
        <?= htmlspecialchars($student['username']) ?>
      </option>
    <?php endforeach; ?>
  </select>
</form>

<?php if ($selected_student_id): ?>
  <form method="post">
    <input type="hidden" name="student_id" value="<?= $selected_student_id ?>">
    <div class="checkboxes">
      <label><strong>Assign Courses:</strong></label>
      <?php foreach ($courses as $course): ?>
        <label>
          <input
            type="checkbox"
            name="course_ids[]"
            value="<?= $course['id'] ?>"
            <?= in_array($course['id'], $assigned_courses) ? 'checked' : '' ?>
          >
          <?= htmlspecialchars($course['title']) ?>
        </label>
      <?php endforeach; ?>
    </div>
    <button type="submit">Save Assignments</button>
  </form>
```

```
    <?php endif; ?>

    <a class="back-link" href="admin_courses.php">← Back to Admin Panel</a>
</div>

</body>
</html>
```

Modify student_view.php to Show Only Assigned Courses

Here’s the updated logic and code for student_view.php:

⍰ Updated student_view.php

```
<?php
require 'db.php';

if (!isset($_SESSION['role']) || $_SESSION['role'] !== 'student') {
    die("Access denied.");
}

$student_id = $_SESSION['user_id'];
```

```
// Get assigned courses
$stmt = $pdo->prepare("
    SELECT c.*
    FROM courses c
    JOIN student_courses sc ON c.id = sc.course_id
    WHERE sc.student_id = ?
");
$stmt->execute([$student_id]);
$courses = $stmt->fetchAll(PDO::FETCH_ASSOC);

// Fetch lectures for each assigned course
$course_lectures = [];
$lecture_stmt = $pdo->prepare("SELECT * FROM lectures WHERE course_id = ?");
foreach ($courses as $course) {
    $lecture_stmt->execute([$course['id']]);
    $course_lectures[$course['id']] = $lecture_stmt->fetchAll(PDO::FETCH_ASSOC);
}
?>
<!DOCTYPE html>
<html>
<head>
    <title>Your Courses</title>
    <style>
        body { font-family: Arial; padding: 20px; background: #f9f9f9; }
        .course { background: #fff; padding: 20px; margin-bottom: 20px; border-radius: 6px; }
        .lecture { margin-left: 20px; padding: 10px 0; border-top: 1px solid #ddd; }
        .lecture a { margin-right: 10px; color: #007bff; text-decoration: none; }
        .lecture a:hover { text-decoration: underline; }
    </style>
</head>
<body>

<h1>Your Assigned Courses</h1>

<?php if (empty($courses)): ?>
    <p><em>No courses assigned yet. Please contact your admin.</em></p>
<?php else: ?>
    <?php foreach ($courses as $course): ?>
        <div class="course">
            <h2><?= htmlspecialchars($course['title']) ?></h2>
            <p><?= nl2br(htmlspecialchars($course['description'])) ?></p>

            <?php if (!empty($course_lectures[$course['id']])): ?>
                <?php foreach ($course_lectures[$course['id']] as $lecture): ?>
                    <div class="lecture">
                        <strong><?= htmlspecialchars($lecture['title']) ?></strong><br>
                        <a href="view_lecture1.php?id=<?= $lecture['id'] ?>">□ View Lecture</a>
                    </div>
                <?php endforeach; ?>
            <?php else: ?>
```

```
                <p><em>No lectures yet.</em></p>

            <?php endif; ?>

        </div>

    <?php endforeach; ?>

<?php endif; ?>


</body>

</html>
```