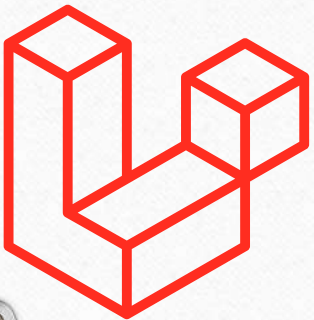




CRUD Operations with PHP & MySQL



Day 8: Introduction to Full stack track with php

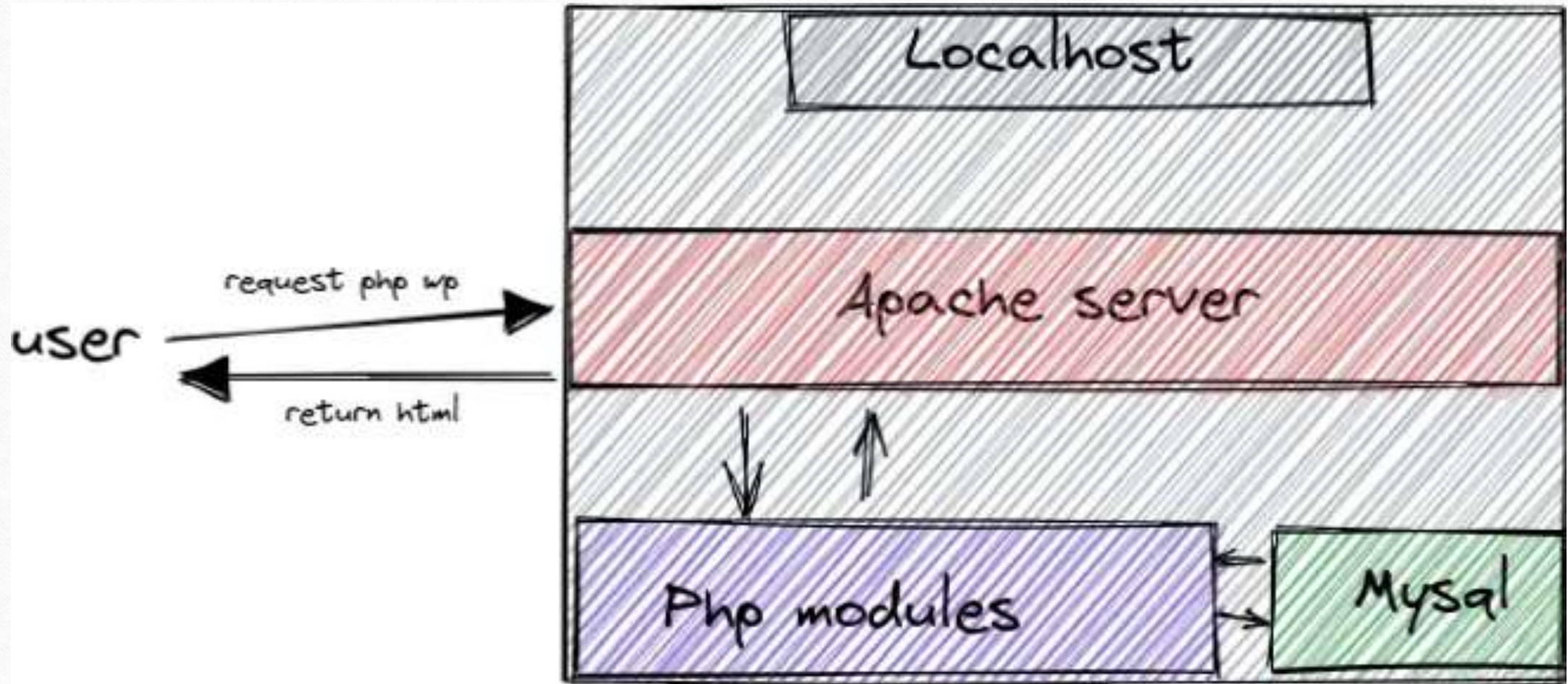
Eng. Ahmed Mohamed Abu-Bakr

Full stack web developer Laravel-react





Day 8: PHP





Day 8: Local Development Environment



XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]

XAMPP Control Panel v3.3.0

Service	Module	PID(s)	Port(s)	Actions
<input type="checkbox"/>	Apache			Start Admin Config Logs
<input type="checkbox"/>	MySQL			Start Admin Config Logs
<input type="checkbox"/>	FileZilla			Start Admin Config Logs
<input type="checkbox"/>	Mercury			Start Admin Config Logs
<input type="checkbox"/>	Tomcat			Start Admin Config Logs

Config
Netstat
Shell
Explorer
Services
Help
Quit

2:13:18 PM [main] Initializing Control Panel
2:13:18 PM [main] Windows Version: Enterprise 64-bit
2:13:18 PM [main] XAMPP Version: 7.4.29
2:13:18 PM [main] Control Panel Version: 3.3.0 [Compiled: Apr 6th 2021]
2:13:18 PM [main] You are not running with administrator rights! This will work for most application stuff but whenever you do something with services there will be a security dialogue or things will break! So think about running this application with administrator rights!
2:13:18 PM [main] XAMPP Installation Directory: "c:\xampp\
2:13:18 PM [main] Checking for prerequisites
2:13:19 PM [main] All prerequisites found
2:13:19 PM [main] Initializing Modules
2:13:19 PM [main] Starting Check-Timer
2:13:19 PM [main] Control Panel Ready

حاليا الجهاز يقدر يتعامل مع لغة
بس اين نضع ملفات **PHP**
الموقع

Open C:
And watch what
is inside it



Day 8: Local Development Environment



XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]

XAMPP Control Panel v3.3.0

Service	Module	PID(s)	Port(s)	Actions
<input type="checkbox"/>	Apache			Start Admin Config Logs
<input type="checkbox"/>	MySQL			Start Admin Config Logs
<input type="checkbox"/>	FileZilla			Start Admin Config Logs
<input type="checkbox"/>	Mercury			Start Admin Config Logs
<input type="checkbox"/>	Tomcat			Start Admin Config Logs

Config
Netstat
Shell
Explorer
Services
Help
Quit

2:13:18 PM [main] Initializing Control Panel
2:13:18 PM [main] Windows Version: Enterprise 64-bit
2:13:18 PM [main] XAMPP Version: 7.4.29
2:13:18 PM [main] Control Panel Version: 3.3.0 [Compiled: Apr 6th 2021]
2:13:18 PM [main] You are not running with administrator rights! This will work for most application stuff but whenever you do something with services there will be a security dialogue or things will break! So think about running this application with administrator rights!
2:13:18 PM [main] XAMPP Installation Directory: "c:\xampp\
2:13:18 PM [main] Checking for prerequisites
2:13:19 PM [main] All prerequisites found
2:13:19 PM [main] Initializing Modules
2:13:19 PM [main] Starting Check-Timer
2:13:19 PM [main] Control Panel Ready

حاليا الجهاز يقدر يفتح قاعدة
بيانات ويعدل عليها

Open

<http://localhost/phpmyadmin/>



Day 8: SESSION



- ✓ تستخدم لتخزين بيانات المستخدم مؤقتاً (مثلاً: حالة تسجيل الدخول)
- ✓ يتم حفظها في السيرفر
- ✓ تحتاج إلى session_start() في أول كل ملف

```
session_start();  
$_SESSION['username'] = 'Ahmed';  
$_SESSION['useremail'] = 'ahmed@g.c';  
var_dump($_SESSION);
```

← → ↻ ⚠ Not secure 192.168.235.115/PHP_training/Day1/PHP_Simple/supergoals.php

```
array(2) { ["username"]=> string(5) "Ahmed" ["useremail"]=> string(9) "ahmed@g.c" }
```

```
19 echo $_SESSION['username'] ;
```

Ahmed



Day 8: PHP Sourcing Files



Include & Require in PHP

تُستخدم لاستدعاء ملفات PHP خارجية داخل ملف حالي لإعادة استخدام الكود (modularity)

 1. `include "file.php";`

تُحاول تضمين الملف.

لو الملف مش موجود: يظهر Warning، ويكمل تنفيذ باقي السكريبت.

 2. `require "file.php";`

تُجبر على وجود الملف.

لو الملف غير موجود: يظهر Fatal Error ويتوقف السكريبت.

 3. `require_once` و `include_once`

الفرق: تمنع تضمين نفس الملف مرتين حتى لو تم استدعاؤه أكثر من مرة.



Day 8: 🧠 Why Databases? Why Not Excel?



الناس زمان كانت بتسجل بيانات في ملفات Excel.

طيب لو عندي مئات المستخدمين؟

صعب أبحث – مفيش علاقات – في مشاكل بالتكرار.

قواعد البيانات = هيكل منظم لتخزين البيانات + سرعة + علاقات + حماية.

✓ مثال سريع:

Excel = جدول واحد.

Database = عدة جداول بينهم علاقات (مثلاً: طلاب، مواد، درجات).



Day 8: 🧠 MySQL in Web Development



✓ أكثر قاعدة بيانات مشهورة مع PHP.

✓ سهلة – مجانية – مدعومة في كل سيرفر.

✓ من أشهر المشاريع اللي بتستخدم (MySQL): WordPress Facebook Laravel with MySQL

Where Databases Are Used

✓ تسجيل دخول

✓ تخزين منتجات في متجر إلكتروني

✓ تتبع الطلاب في منصة تعليمية

✓ رفع صور + وصفها

✓ تتبع الطلبات والشحن



Day 8: 🧠 Database Concepts



✓ قاعدة بيانات: هي مجموعة من الجداول.

✓ جدول: (Table) يشبه صفحة Excel — فيه أعمدة (Columns) وصفوف (Rows)

✓ Column Types:

النوع	الاستخدام	ملاحظات
INT	رقم صحيح	أرقام فقط
VARCHAR	نص بطول محدد	سريع + يُفضل للحجم الصغير
TEXT	نص طويل	بطيء نسبياً لكن غير محدود الطول
DATE	تاريخ	لحفظ التاريخ

(indexing)

✓ Primary Key

هو العمود الأساسي الذي يميز كل صف.
لازم يكون Unique و Not Null مثال:
id هو Primary Key في جدول الطلاب.

✓ Foreign key

يربط الجدول بجدول ثاني عن طريقة
ال primary key.



Day 8: 🧠 Table Relationships



✓ قاعدة بيانات: هي مجموعة من الجداول.

✓ جدول (Table): يشبه صفحة Excel — فيه أعمدة (Columns) وصفوف (Rows)

العلاقة	الوصف	مثال
One to One	صف يرتبط بصف واحد فقط	كل موظف له ملف شخصي واحد فقط
One to Many	صف مرتبط بعدة صفوف أخرى	كل مستخدم له عدة طلبات
Many to Many	يتم من خلال جدول وسيط	طالب يدرس عدة مواد، والمادة يدرسها عدة طلاب



Day 8: 🧠 Table Relationships



✓ أولاً: علاقة One-to-One - واحد إلى واحد

📌 التعريف:

كل سجل في الجدول الأول مرتبط بسجل واحد فقط في الجدول الثاني، والعكس.

🎓 مثال:

عندنا جدول students وجدول student_profiles، كل طالب له ملف شخصي واحد فقط.

✓ ثانياً: علاقة One-to-Many واحد إلى متعدد

📌 التعريف:

كل سجل في الجدول الأول يمكن أن يرتبط بعدة سجلات في الجدول الثاني، لكن كل سجل في الجدول الثاني مرتبط بسجل واحد فقط في الأول.

🎓 مثال:

عندنا جدول teachers وجدول courses. كل مدرس ممكن يدرس أكثر من كورس، لكن الكورس الواحد له مدرس واحد فقط.



Day 8: 🧠 Table Relationships



✓ ثالثاً: علاقة Many-to-Many متعدد إلى متعدد

📌 التعريف:

كل سجل في الجدول الأول ممكن يرتبط بعدة سجلات في الجدول الثاني، والعكس صحيح.

🎓 مثال:

عندنا جدول students و courses، الطالب ممكن يسجل في أكثر من كورس، والكورس الواحد فيه طلاب كثير.

🧩 لحل هذه العلاقة، بنعمل جدول ثالث (Pivot Table) اسمه مثلاً enrollments.

نوع العلاقة	مثال حقيقي	شرح مبسط
One to One	طالب ↔ ملف شخصي	كل طالب له ملف واحد فقط والعكس
One to Many	مدرس ↔ كورسات	كل مدرس يدرس أكثر من كورس
Many to Many	طالب ↔ كورسات (عبر جدول enrollments)	الطالب يسجل في كذا كورس والعكس



Day 8: 🧠 columns datatypes



■ أنواع الأعمدة (Columns) في MySQL وأفضل استخدام لكل نوع

النوع	مثال مدخل	استخدامه	ملاحظات
INT	1, 20, 1500	أرقام صحيحة (, ID العمر...)	سهل البحث والترتيب
VARCHAR(n)	"Ali", "Ahmed123"	نص قصير (اسم، بريد...)	أسرع من TEXT لكن محدود بطول n
TEXT	فقرات طويلة	وصف، محتوى مقال، تعليق طويل	لا يمكن فهرسته بسهولة – أبطأ
DATE	"2025-07-21"	تواريخ (تاريخ ميلاد، تسجيل)	يدعم عمليات مقارنة وتصفية
DATETIME	"2025-07-21 10:30:00"	تاريخ مع وقت	مناسب للطوابع الزمنية
BOOLEAN	0 أو 1	true/false (نشط؟ مدير؟...)	سهل التحقق منه
FLOAT / DECIMAL	99.99 ، 5.25	أرقام عشرية (سعر، نسبة...)	DECIMAL أدق للفلوس



Day 8: 🧠 Table Relationships



■ ماذا لو خزنت كل البيانات كنص TEXT؟

- ✗ السرعة: TEXT أبطأ لأن النظام لا يستطيع استخدامه للفهرسة بكفاءة.
- ✗ الحجم: TEXT ييشغل مساحة أكبر على القرص.
- ✗ فقدان المعنى: لو سجلت العمر كنص، مش هتقدر تقول "هات اللي عمرهم < 20".
- ✗ مشاكل التحقق من البيانات: ممكن المستخدم يدخل "ahmed" بدل رقم أو تاريخ بدون ما تفهم القاعدة إنها مشكلة.

● القاعدة الذهبية:

استخدم كل نوع في مكانه الصحيح، عشان الأداء، والوضوح، والتكامل.



Day 8: 🧠 Data Base Design



■ إزاي أبدأ تصميم قاعدة بيانات لتطبيق معين؟

مثال: تطبيق "نظام طلاب"

1. 📌 حدد (Entities) العناصر الرئيسية اللي هتعامل معاها وهيكون لها دور في السيستم

□ طلاب (Students)

□ كورسات (Courses)

□ درجات (Grades)



Day 8: 🧠 Data Base Design



إزاي أبدأ تصميم قاعدة بيانات لتطبيق معين؟

مثال: تطبيق "نظام طلاب"

2. لكل Entity : حدد الأعمدة الأساسية

📌 Students

الاسم	النوع
id	INT + PRIMARY KEY + AUTO_INCREMENT
name	VARCHAR(100)
email	VARCHAR(150)
birth_date	DATE

📌 Courses

الاسم	النوع
id	INT + PRIMARY KEY
title	VARCHAR(100)
hours	INT

📌 Grades

الاسم	النوع
student_id	INT (FOREIGN KEY to students.id)
course_id	INT (FOREIGN KEY to courses.id)
grade	FLOAT



Day 8: 🧠 Data Base Design



■ إزاي أبدأ تصميم قاعدة بيانات لتطبيق معين؟

مثال: تطبيق "نظام طلاب"

📌 3. حدد العلاقات:

طالب ← له أكثر من كورس (Many to Many)
نعمل جدول grades كجدول وسيط (intermediate table)

← خلاصة
END

- ☐ لا تستخدم TEXT إلا لما يكون المحتوى فعلاً طويل زي وصف منتج.
- ☐ استخدم INT, DATE, BOOLEAN, VARCHAR حسب نوع الداتا.
- ☐ تصميم قواعد البيانات يبدأ بتحليل الكيانات والعلاقات. MySQL قوية لأنها بتفهم نوع الداتا وبتساعدك تبني أنظمة قوية وأمنة.



Day 8: 🧠 Data Base Design



🧠 ملاحظات مهمة:

الـ ERD مش مجرد رسم، ده بيساعدك تشوف الأخطاء قبل ما تكتب كود.
لو تخطيت خطوة التطبيق، ممكن يحصل تكرار كبير في البيانات، يؤدي لمشاكل لاحقاً.

Primary Key = الهوية،

Foreign Key = الربط.



Day 8: 🧠 What is Normalization



Normalization هو عملية تنظيم البيانات داخل قاعدة البيانات بطريقة تقلل التكرار وتمنع التعارض (inconsistency).

✓ Why Normalize?

تقليل استهلاك المساحة 📁

منع التكرار المزعج (redundancy) 🔄

سهولة التعديل والتحديث 🛠️

تحسين أداء الاستعلامات (queries) 🔍

🧱 Normalization Levels (Up to 3NF)

◆ 1NF – First Normal Form

"كل خلية تحتوي على قيمة واحدة فقط"

◆ 2NF – Second Normal Form

"كل عمود يعتمد فقط على الـ 'Primary Key'"

◆ 3NF – Third Normal Form

"ما فيش عمود يعتمد على عمود غير المفتاح الأساسي"



Day 8: 🧠 Before DB creation



🧩 Let's Create a Database Design Flow

🎯 الخطوة 1: فهم النظام – أسأل نفسك

✅ "أنا ببني نظام بيعمل إيه؟"

✅ مثال عملي:-

✓ منصة كورسات أونلاين

➤ نظام بيسمح:-

☐ تسجيل الطلاب

☐ عرض كورسات

☐ كل طالب ممكن يسجل في أكثر من كورس

☐ كل كورس ممكن يكون فيه أكثر من طالب

☐ ممكن أحتاج أسجل درجات أو تقييم



Day 8: 🧠 Before DB creation



🧩 Let's Create a Database Design Flow

الخطوة 2: 🎯

تحديد الكيانات الأساسية (Entities)

🎓 Student

- id (PK)
- Name
- Email
- Date-of-birth
- phone

📘 Course

- id (PK)
- Title
- Description
- Hours
- price

📊 Enrollment

- (student_id) (FK → Student)
- course_id (FK → Course)
- Grade
- enrollment_date

من الفهم السابق أستنتج:

👤 Student

📘 Course

📄 Enrollment (العلاقة اللي بينهم)



Day 8: 🧠 Before DB creation



🧩 Lt's Choose Data Type of each columns

الجدول	العمود	النوع المناسب	السبب
students	id	INT (PK)	مفتاح أساسي للتعريف
	name	VARCHAR(100)	نص قصير
	email	VARCHAR(150)	بريد - محدود الحجم
	phone	VARCHAR(20)	أرقام قد تبدأ بصفر أو علامة +
	date_of_birth	DATE	علشان نعمل حساب عمر
courses	id	INT (PK)	مفتاح أساسي
	title	VARCHAR(100)	عنوان بسيط
	description	TEXT	شرح طويل
	hours	DECIMAL(5,2)	عدد ساعات
	price	DECIMAL(8,2)	رقم عشري للفلوس
enrollments	id	INT (PK)	مفتاح
	student_id	INT (FK)	ربط بالطلاب
	course_id	INT (FK)	ربط بالكورسات
	grade	FLOAT	ممكن تكون 95.5
	enrollment_date	DATETIME	وقت التسجيل بدقة



Day 8: 🧠 Before DB creation



🧩 Let's Normalize Data

■ 1NF:

✓ كل خلية تحتوي على قيمة واحدة فقط

ما فيش كورسات داخل خلية واحدة – كل كورس صف مستقل.

■ 2NF:

✓ كل البيانات تعتمد فقط على الـ PK

مفیش email داخل – enrollments هي موجودة فقط عند الطلاب.

■ 3NF:

✓ الأعمدة لا تعتمد على أعمدة أخرى غير الـ PK

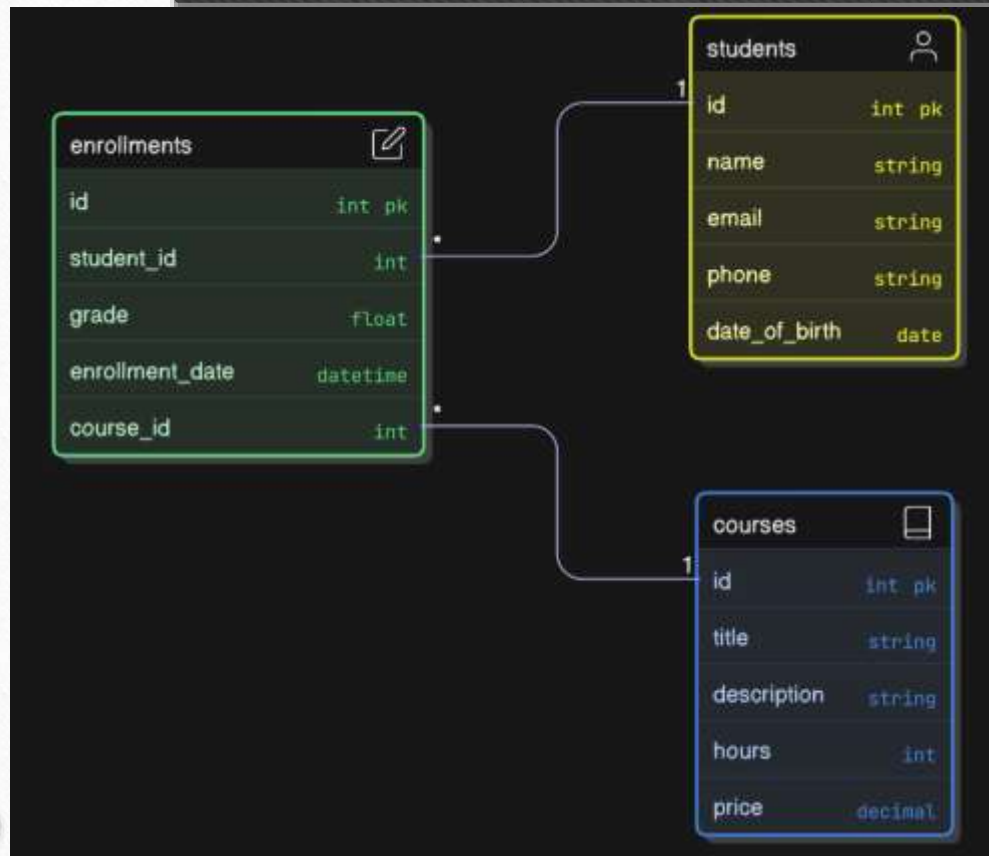
السعر والشرح موجودين في جدول الكورسات، مث في enrollments.



Day 8: 🧠 Before DB creation



🌱 Lt's create ERD



✅ الخطوة 7:

تحويل إلى قاعدة بيانات فعلية

📌 نروح على phpMyAdmin

نبدأ بإنشاء قاعدة البيانات training_system

ثم ننفذ الجداول السابقة يدويًا أو بـ SQL

🌱 ملخص الأسئلة التي أسألها قبل تصميم أي قاعدة بيانات:

إيه الكيانات في النظام ده؟ كل كيان فيه إيه؟ (صفاته) إيه العلاقات بينهم؟ إيه

نوع كل عمود؟ هل في أي تكرار أقدر أخلص منه؟ إزاي أرسم العلاقة

(ERD) ؟ إزاي أحول دا إلى SQL وأبدأ بتنفيذ فعلي؟



Day 8: 🧠 phpMyAdmin



🧩 It's Create a database in phpMyAdmin

🎯 السيناريو: نظام تسجيل كورسات للطلاب

1. فتح 📄 phpMyAdmin:

اكتب في المتصفح: <http://localhost/phpMyAdmin>

2. إنشاء قاعدة بيانات جديدة 📄

اضغط "New"

اسم القاعدة: training_system

Charset: utf8mb4_general_ci

اضغط "Create"



Day 8: 🧠 phpMyAdmin



🧩 Let's Create a database in phpMyAdmin

SQL = Structured Query Language
هي اللغة المستخدمة للتعامل مع قواعد البيانات العلائقية (مثل MySQL – PostgreSQL)

🧩 تصنيفات أوامر SQL الأساسية:

التصنيف	الوظيفة الأساسية	أشهر الأوامر
 DML	Data Manipulation Language	SELECT, INSERT, UPDATE, DELETE
 DDL	Data Definition Language	CREATE, ALTER, DROP, RENAME
 DCL	Data Control Language	GRANT, REVOKE
 TCL	Transaction Control Language	COMMIT, ROLLBACK, SAVEPOINT



Day 8: 🧠 phpMyAdmin



🧩 It's Create a database in phpMyAdmin

SQL = Structured Query Language
هي اللغة المستخدمة للتعامل مع قواعد البيانات العلائقية (مثل MySQL – PostgreSQL)

النوع	الكود	الوصف
🔍 SELECT	<code>SELECT * FROM students</code>	استعلام عن كل البيانات
+ INSERT	<code>INSERT INTO students (name, email) VALUES ('Ali', 'ali@mail.com')</code>	إدخال بيانات جديدة
✏️ UPDATE	<code>UPDATE students SET name = 'Omar' WHERE id = 1</code>	تعديل بيانات
✖️ DELETE	<code>DELETE FROM students WHERE id = 1</code>	حذف بيانات



Day 8: 🧠 phpMyAdmin



🧩 Let's Create a database in phpMyAdmin

SQL = Structured Query Language

هي اللغة المستخدمة للتعامل مع قواعد البيانات العلائقية (مثل MySQL – PostgreSQL)

النوع	الكود	الوصف
🧱 CREATE	CREATE TABLE students (...)	إنشاء جدول جديد
🔧 ALTER	ALTER TABLE students ADD phone VARCHAR(20)	تعديل هيكل جدول
🗑️ DROP	DROP TABLE students	حذف جدول بالكامل
🔗 JOIN	SELECT * FROM students JOIN courses ON ...	الربط بين جدولين
🎯 WHERE	SELECT * FROM users WHERE email LIKE '%@gmail.com'	شرط للتصفية
📊 ORDER BY	SELECT * FROM courses ORDER BY price DESC	ترتيب النتائج
📄 LIMIT	SELECT * FROM students LIMIT 5	عرض عدد معين فقط



Day 8: 🧠 phpMyAdmin



🧩 Lt's Create a database in phpMyAdmin

```
CREATE DATABASE IF NOT EXISTS training_system
CHARACTER SET utf8mb4
COLLATE utf8mb4_general_ci;

USE training_system;

CREATE TABLE students (
  id INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(100) NOT NULL,
  email VARCHAR(150) UNIQUE,
  phone VARCHAR(20),
  date_of_birth DATE
);

CREATE TABLE courses (
  id INT AUTO_INCREMENT PRIMARY KEY,
  title VARCHAR(100) NOT NULL,
  description TEXT,
  hours DECIMAL(4,2),
  price DECIMAL(8,2)
);

CREATE TABLE enrollments (
  id INT AUTO_INCREMENT PRIMARY KEY,
  student_id INT NOT NULL,
  course_id INT NOT NULL,
  grade FLOAT DEFAULT NULL,
  enrollment_date DATETIME DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (student_id) REFERENCES students(id) ON DELETE CASCADE,
  FOREIGN KEY (course_id) REFERENCES courses(id) ON DELETE CASCADE
);
```

```
USE training_system;
```

```
INSERT INTO students (name, email, phone, date_of_birth)
VALUES
  ('Ahmed Ali', 'ahmed@example.com', '0111111111', '2000-05-10'),
  ('Sara Youssef', 'sara@example.com', '0122222222', '2001-01-20');
```

```
INSERT INTO courses (title, description, hours, price)
VALUES
  ('Web Development', 'Intro to HTML, CSS, PHP', 40.00, 1500.00),
  ('MySQL Basics', 'Database fundamentals', 20.00, 1000.00);
```

```
INSERT INTO enrollments (student_id, course_id, grade)
VALUES
  (1, 1, 85.5),
  (2, 2, 90.0);
```




Day 8: 🧠 phpMyAdmin



🌱 Let's Connect PHP to MySQL + Print Student Table

```
<?php
$host = "localhost";
$user = "root";
$pass = "";
$dbname = "training_system";

// إنشاء الاتصال
$conn = new mysqli($host, $user, $pass, $dbname);

// التحقق من الاتصال
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
?>
```

```
<?php include "db.php";
$sql = "SELECT * FROM students";
$result = $conn->query($sql);

if ($result && $result->num_rows > 0):
    while ($row = $result->fetch_assoc()):
        ?>
        <tr>
            <td><?= $row['id'] ?></td> <td><?= $row['name'] ?></td> <td><?=
$row['email'] ?></td> <td><?= $row['phone'] ?></td> <td><?=
$row['date_of_birth'] ?></td>
        </tr>
    <?php
        endwhile;
    else:
        echo "<tr><td colspan='5'>No students found.</td></tr>";
    endif;
?>
```



Day 8: First CRUD Project With DB



student dashboard Task

Project Structure

training_system/

— db.php

— navbar.php

— index.php

— add_student.php

— insert_student.php

— edit_student.php

— update_student.php

— delete_student.php

← عرض الطلاب

← نموذج إضافة طالب



Day 8: 🧠 phpMyAdmin



🧩 Lt's Connect PHP to MySQL + Print Student Table

الوظيفة	الدالة
الاتصال بقاعدة البيانات	<code>mysqli_connect()</code>
تنفيذ استعلام	<code>mysqli_query()</code>
جلب النتائج كمصفوفة مفتاحية	<code>mysqli_fetch_assoc()</code>
عدد الصفوف المرتجعة	<code>mysqli_num_rows()</code>
يعرض آخر خطأ	<code>mysqli_error()</code>
إغلاق الاتصال	<code>mysqli_close()</code>



Day 8: CRUD



CRUD هي اختصار لـ:

- ✓ Create: إنشاء بيانات جديدة .
- ✓ Read: قراءة البيانات وعرضها .
- ✓ Uppdate: تعديل البيانات .
- ✓ Delte: حذف البيانات .

هذه هي العمليات الأساسية لأي تطبيق ويب ديناميكي يعمل مع قاعدة بيانات .



Day 8: Structure of a CRUD Page



(Students Example)

Each entity (like student) will usually have:

- ☐ Listing page: Shows all entries.
- ☐ Add form: Submits new data.
- ☐ Edit form: Pre-filled data to update.
- ☐ Delete action: Usually a button with confirmation.

→ index.php عرض البيانات

→ add_student.php إضافة بيانات جديدة

→ insert_student.php تخزين البيانات الجديدة

→ update_student.php تحديث البيانات في قاعدة البيانات

→ delete_student.php حذف البيانات



Day 8: Structure of a CRUD Page



(Students Example)

إضافة طالب جديد:

☐ المستخدم يفتح add_student.php

☐ يدخل البيانات ويضغط Submit

☐ البيانات تروح لـ insert_student.php

☐ يتم تنفيذ أمر SQL

INSERT INTO students (...) VALUES (...)

☐ بعد الحفظ يتم التوجيه إلى index.php لعرض التحديث

☐ نفس النمط بيتكرر في التعديل والحذف.



Day 8: Structure of a CRUD Page



CRUD Security Tips

استخدم validation و sanitization على كل مدخلات المستخدم
دائمًا افحص البيانات قبل إدخالها في قاعدة البيانات
استخدم `mysqli_real_escape_string()` أو prepared statements
تأكد من وجود تأكيد قبل الحذف (Confirm box)
لا تعتمد فقط على ID من URL بدون تحقق أو صلاحيات



Day 8: Structure of a CRUD Page



Frontend Best Practices (with Bootstrap)

استخدام Bootstrap cards/tables لتنظيم البيانات

تنسيق المدخلات داخل forms بطريقة واضحة

تقديم ملاحظات للمستخدم (Success / Error alerts)

استخدام Navbar لتسهيل التنقل بين الأقسام

الحفاظ على تصميم متناسق ومتجاوب مع الأجهزة



Day 8: Structure of a CRUD Page



Live Task – Students CRUD

Training System

[Students](#) [Courses](#) [Enrollments](#)

Student List

[+ Add Student](#)

Name	Email	Phone	DOB	Actions
Ahmed Ali	ahmed@example.com	055555555555	2000-05-10	Edit Delete
Sara Youssef	sara@example.com	01222222222	2001-01-20	Edit Delete
ahmed mohamed	ahmedabubakr148@gmail.com	01113284597	2025-07-08	Edit Delete



Day 8: Structure of a CRUD Page



Task – Courses CRUD

المطلوب بشكل فردي: 

إنشاء نفس النظام على جدول courses
المطلوب في الكورسات:-

courses.php ☐

add_course.php ☐

insert_course.php ☐

edit_course.php ☐

update_course.php ☐

pdelete_course.php ☐

الحقول: title, description, hours, price ➤

استخدم Bootstrap بنفس تنسيق الطلاب ✓



Day 8: JOINS



ما هو الـ JOIN؟
هو طريقة لجلب البيانات من جدولين أو أكثر بناءً على علاقة (عادةً باستخدام مفتاح خارجي FOREIGN KEY)

✓ 1. INNER JOIN الربط الداخلي

التعريف:

بيعرض فقط السجلات التي ليها علاقة بين الجدولين.
مثال: عندنا جدولين:

```
students(id, name)
enrollments(id, student_id, course)
```

```
SELECT students.name, enrollments.course
FROM students
INNER JOIN enrollments ON students.id = enrollments.student_id;
```

النتيجة:

يعرض الطلاب التي سجلوا فعلياً في كورسات فقط.

لما تكتب JOIN بدون ما تحدد النوع، فهو اختصار مباشر لـ INNER JOIN.



Day 8: JOINS



ما هو الـ JOIN؟

هو طريقة لجلب البيانات من جدولين أو أكثر بناءً على علاقة (عادةً باستخدام مفتاح خارجي (FOREIGN KEY)).

✓ 2. LEFT JOIN الربط من اليسار

التعريف:

يعرض كل السجلات من الجدول الأول (اليسار)،
وبيجيب البيانات من الجدول الثاني إذا كانت موجودة، وإلا يحط NULL.

```
students(id, name)
enrollments(id, student_id, course)
```

```
SELECT students.name, enrollments.course
FROM students
LEFT JOIN enrollments ON students.id = enrollments.student_id;
```

النتيجة:

يعرض كل الطلاب، حتى اللي مش مسجلين في كورسات، ويظهر NULL في خانة الكورس.



Day 8: JOINS



ما هو الـ JOIN؟

هو طريقة لجلب البيانات من جدولين أو أكثر بناءً على علاقة (عادةً باستخدام مفتاح خارجي (FOREIGN KEY)).

✓ 3. RIGHT JOIN الربط من اليمين

التعريف:

عكس LEFT JOIN ، يعرض كل السجلات من الجدول الثاني (اليمين)، ويجيب بيانات الجدول الأول لو لقاها.

```
students(id, name)
enrollments(id, student_id, course)
```

```
SELECT students.name, enrollments.course
FROM students
RIGHT JOIN enrollments ON students.id = enrollments.student_id;
```

النتيجة:

يعرض كل الكورسات المسجل فيها طلاب، ولو في كورسات ما لهاش طالب مسجل، يظهر NULL في خانة الطالب.



Day 8: JOINS



ما هو الـ JOIN؟

هو طريقة لجلب البيانات من جدولين أو أكثر بناءً على علاقة (عادةً باستخدام مفتاح خارجي FOREIGN KEY).

نوع الـ JOIN	يعرض من الجدول الأول؟	يعرض من الجدول الثاني؟	فقط لو في علاقة؟
INNER JOIN	نعم ✓	نعم ✓	نعم ✓
LEFT JOIN	نعم ✓	نعم / NULL ✓	✗
RIGHT JOIN	نعم / NULL ✓	نعم ✓	✗
FULL OUTER JOIN	نعم / NULL ✓	نعم / NULL ✓	✗



Day 8: JOINS



✓ مثال عملي سريع:

id	name
1	Ahmed
2	Mariam
3	Samir

id	student_id	course
1	1	Math
2	1	Physics
3	2	Chemistry

- ✓ INNER JOIN ← يعرض أحمد ومريم فقط
- ✓ LEFT JOIN ← يعرض أحمد، مريم، سامر (سامر ليس لديه كورسات = NULL)
- ✓ RIGHT JOIN ← يعرض أحمد، مريم فقط، لأن كل الكورسات تخصهم فقط
- ✓ FULL OUTER JOIN ← يعرض الكل حتى لو مفيش تطابق



Day 8: Structure of a CRUD Page



Task – Enrollments CRUD

إنشاء CRUD لإدارة التسجيلات enrollments
المطلوب:

ربط طالب بكورس

تخزين grade

عرض قائمة تحتوي:

اسم الطالب

عنوان الكورس

التقدير

التاريخ

استخدم العلاقات JOIN بين الجداول

استخدم Select dropdowns لاختيار الطالب والكورس

The End Of Day 8



Feel free to contact me any time

linktr.ee/ahmed_abubakr