



# Graduation Project

## STUDENT

### ATTENDANCE

### TRACKING WEB APP

2024-2025

تحت اشراف  
د/عاطف رسلان  
أ/احمد سباعي

#### فريق العمل

اسم الطالب	الكود
كيرلس اسامه فوزي نجيب صليب	12268323
كيرلس اسحاق لبس دميان	12269281
كيرلس روماني سمير رياض	1226894
كيرلس ثروت تقى فهيم	12268518
يوسف اشرف صليب شنودة	12269176



# STUDENT ATTENDANCE

## أهدا

بكل مشاعر التقدير والاحترام، نهدي ثمرة هذا العمل إلى:

### الدكتور الفاضل (عاطف رسلان)

الذي لم يدخل علينا بعلمه، وجعل من دراستنا رحلة مليئة بالتحدي والإلهام. شكرًا لك على ما قدمته من وقت وجهد، وعلى إيمانك بقدراتنا، فكنت بحق قدوة ملهمة وداعمة لنا.

### المعيد (المهندس أحمد السباعي)

الذي حول كل درس إلى تجربة تعليمية ممتعة ومثيرة. شكرًا على تفانيك وإخلاصك في نقل المعرفة، وعلى جهودك المستمرة لتبسيط المفاهيم وتعزيز الفهم.

### الدكتور (سيد مشعال) - عميد المعهد

الذي يمثل نموذجًا للقيادة الحكيمة والرؤية المستقبلية. شكرًا على حرصك الدائم على تطوير المعهد، وتوفير بيئة تعليمية متميزة. كانت توجيهاتك وستظل مصدر إلهام لنا جميعًا.

إلى كل من ساهم في رحلتنا التعليمية، لكم مني أسمى آيات الشكر والتقدير.

# STUDENT ATTENDANCE

# الفهرس

الباب الأول: المقدمة

مشكلة الدراسة 2

فكرة المشروع 1

أهداف المشروع 4

مفهوم الـ Dynamic Webpages 3

الأدوات والتقنيات المستخدمة 5

الباب الثاني: دراسة النظام الحالي

منهجية التطوير 2

مقدمة المشروع 1

تقسيم المهام 4

مكونات الموقع 3

الباب الثالث: تحليل النظام

Use Case (UC) مخطط الـ 5

تحديد مواصفات ومتطلبات المشروع 1  
Requirements Specifications (SRS)

مخطط تدفق البيانات 2  
Flow Diagram (DFD)

مخطط الـ 4  
Diagram (AD)

# STUDENT ATTENDANCE

## الفهرس

الباب الرابع : تصميم النظام System Design

1 مخطط Relational Schema Diagram

2 إعداد قاموس البيانات Data Dictionary

الباب الخامس : تنفيذ النظام System Implementation

1 مقدمة

2 خطوات انشاء وتصميم قاعده البيانات  
MySQL Server

3 النشر والاختبار

Install 2 تثبيت بيئه العمل  
The Working  
Environments

4 تصميم صفحات المشروع  
المختلفة مع الأكواد  
Github و  
الصور

# **الباب الأول**

---

# فكرة المشروع

---

تتمثل فكرة المشروع في إنشاء نظام إلكتروني يسهل عملية تسجيل حضور الطلاب بشكل ديناميكي وحديث باستخدام رموز QR. يمكن للأساتذة إنشاء رمز QR لكل محاضرة يقوم الطلاب بمسحه لتسجيل حضورهم بسهولة، مما يقلل من الأخطاء ويوفر الوقت والجهد لجميع الأطراف.

## مشكلة الدراسة

---

المشكلة التي نسعى إلى حلها هي تعقيد وتكلف عملية تسجيل الحضور التقليدية التي تستهلك وقتاً كبيراً وتكون عرضة للأخطاء البشرية. بالإضافة إلى ذلك، غالباً ما تكون البيانات غير منتظمة وصعبة التحليل في الأنظمة التقليدية، مما يجعل من الصعب على المدرسين والإداريين متابعة الحضور بشكل فعال.

## مفهوم الـ Dynamic Webpages

تعرف صفحات الويب الديناميكية بأنها صفحات تُحدث محتواها أو عناصرها بشكل تلقائي بناءً على تفاعل المستخدم أو استجابة من الخادم. على عكس الصفحات الثابتة التي تحتوي على محتوى ثابت لا يتغير إلا بتشغيل مباشر للملفات، فإن الصفحات الديناميكية يمكن أن تعرض محتوى مختلف في كل مرة يتم فيها تحميل الصفحة بناءً على مجموعة من العوامل.

### الخصائص الرئيسية للصفحات الديناميكية:

- التكامل مع البيانات: يتم سحب البيانات من قواعد البيانات لتحديث محتوى الصفحة بناءً على طلب المستخدم.
- التفاعل مع المستخدم: يتم إنشاء هذه الصفحات للتفاعل مع المستخدم، مثل إدخال البيانات في نماذج أو النقر على الأزرار.
- المعالجة على الخادم: يتم استخدام (Framework Laravel) لمعالجة الطلبات على الخادم وتحديث المحتوى المعروض على الصفحة.

### أمثلة على الصفحات الديناميكية:

1. م الواقع التجارة الإلكترونية.
2. م الواقع الشبكات الاجتماعية التي تُحدث محتواها باستمرار دون الحاجة لإعادة تحميل الصفحة.

# STUDENT ATTENDANCE

## أهداف المشروع

- أتمتة عملية تسجيل الحضور باستخدام تقنية رمز QR، لضمان نظام سريع وموثوق.
- تسهيل عملية توليد رموز QR فريدة لكل حاضرة على الأساتذة، مما يقلل الوقت المستغرق في تسجيل الحضور يدوياً.
- توفير وسيلة مريحة وميسرة للطلاب لتسجيل حضورهم ببساطة عن طريق مسح الرمز.



# STUDENT ATTENDANCE

## الأدوات والتقنيات المستخدمة

المصادقة: تم تنفيذ Jetstream للمصادقة وتسجيل المستخدمين، مما يوفر عملية إدارة مستخدم آمنة وبسيطة.

04

إطار العمل الخلفي: Laravel، بالاشتراك مع Livewire للتفاعل الدинاميكي بين الواجهة الأمامية والخلفية. يسمح لنا Livewire ببناء واجهات حديثة وتفاعلية دون تعقيد إطار JavaScript كامل. (المصدر: JavaScript: (المصدر: Laravel كامل.

بيئة تطوير متكاملة (IDE): تم استخدام PhpStorm لكتابة وإدارة الكود بكفاءة، مما يزيد من الإنتاجية.

05

تصميم الواجهة الأمامية: تم استخدام Bootstrap v5 لإنشاء واجهة نظيفة وسهلة الاستخدام للتطبيق، مما يضمن تجربة استجابة وجذابة على جميع الأجهزة.

01

02

03



# STUDENT **ATTENDANCE**

**الباب الثاني**

# مقدمة المشروع

هذا المشروع هو تطبيق ويب مصمم لإدارة حضور الطلاب في المحاضرات باستخدام مسح رمز QR. جاءت الفكرة من الحاجة إلى تحديث عمليات تتبع الحضور، وجعلها أكثر كفاءة وأقل عرضة للأخطاء البشرية. يحل نظامنا محل الأساليب التقليدية بنظام مؤتمت يوفر الوقت ويقلل من عبء العمل الإداري لكل من الأساتذة والطلاب.



# STUDENT ATTENDANCE

## منهجية التطوير

اعتمدنا منهجية التطوير الرشيقه (Agile)، مع التركيز على التغذية الراجعة المستمرة، والتطوير التكراري، والتحسينات المتزايدة. كان الفريق يعقد اجتماعات أسبوعية لمراجعة التقدم وضمان توافق جميع الأعضاء مع الأهداف.

## مكونات الموقع

إدارة حضور المحاضرات: يقوم الأساتذة بتوليد رمز QR فريد لكل محاضرة، ويقوم الطلاب بمسح الرمز لتسجيل حضورهم.

الأدوار المختلفة للمستخدمين: يتضمن النظام أدواتاً مختلفة مثل المدير والأستاذ والطالب، ولكل منهم ميزات خاصة تناسب دوره.

لوحة التحكم الإدارية: تم بناؤها باستخدام Filament PHP، وتتيح لوحة التحكم الإدارية إدارة حسابات المستخدمين والوصول الدراسية وسجلات الحضور.

واجهة المستخدم: تم تطوير واجهة سهلة الاستخدام للطلاب باستخدام Bootstrap v5، مما يوفر سهولة في التنقل والوصول السريع إلى ميزات الحضور.

## تقسيم المهام

تم تقسيم تطوير المشروع بين أعضاء الفريق على النحو التالي:

كيرلس أسامة: تطوير الخلفية، تكامل Livewire، والنشر.

كيرلس إسحاق: تطوير الواجهة الأمامية وتصميم Bootstrap.

كيرلس رومانى: تصميم وإدارة قاعدة البيانات.

كيرلس ثروت: تطوير لوحة التحكم باستخدام Filament PHP.

يوسف أشرف: اختبار الجودة وتحسين واجهة المستخدم وتجربة المستخدم.

# STUDENT **ATTENDANCE**

الباب الثالث

# STUDENT ATTENDANCE

## تحديد مواصفات ومتطلبات المشروع (SRS)

### المتطلبات الوظيفية:

- قدرة النظام على إنشاء رموز QR فريدة لكل محاضرة.
- تسجيل الحضور تلقائياً عند مسح رمز QR بواسطة الطالب.
- توفير لوحة تحكم للمسؤول لمراجعة تقارير الحضور.
- دعم تسجيل الدخول للمستخدمين بناءً على أدوارهم (طالب، أستاذ، مسؤول).

### المتطلبات غير الوظيفية:

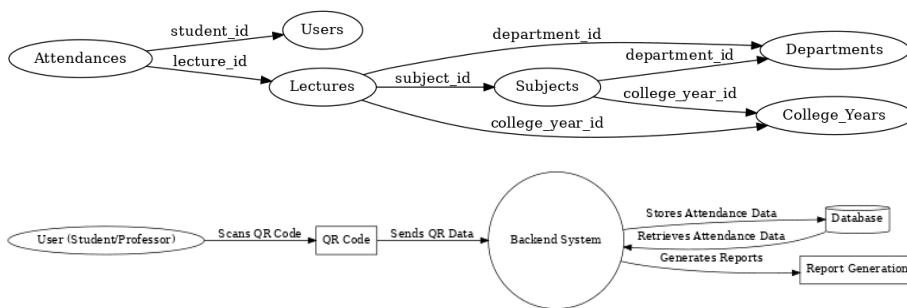
- يجب أن يكون النظام آمناً ويستخدم بروتوكولات HTTPS.
- يجب أن يدعم النظام تحميل البيانات بسرعة دون تأخير.
- واجهة المستخدم يجب أن تكون سهلة الاستخدام وتدعم الأجهزة المختلفة.
- يجب أن تكون قاعدة البيانات قابلة للتتوسيع لدعم أعداد كبيرة من المستخدمين.

# STUDENT ATTENDANCE

## مخطط تدفق البيانات Data Flow Diagram (DFD)

### مفهوم مخطط تدفق البيانات (DFD):

يُستخدم مخطط تدفق البيانات لتصوير تدفق البيانات بين مكونات النظام المختلفة. يساعد هذا المخطط في توضيح كيفية انتقال البيانات داخل النظام، بدءاً من المصدر (المستخدم) وحتى وصولها إلى قاعدة البيانات أو أي عملية أخرى.



### DFD للمشروع:

### المستوى الأول (Level 0):

#### المدخلات:

- المستخدم (طالب/أستاذ) يقوم بتسجيل الدخول.
- الطالب يقوم بمسح رمز QR.

#### العمليات:

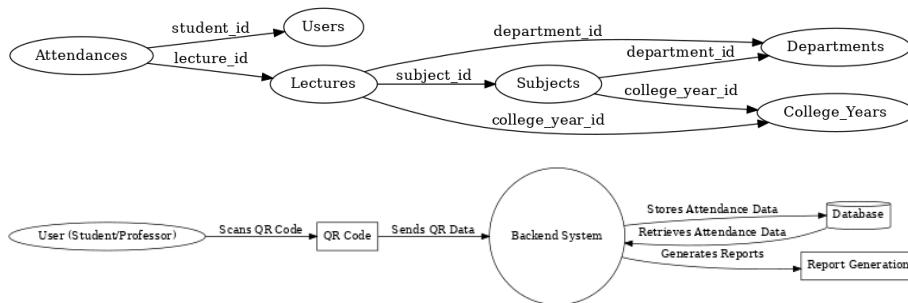
- توليد رمز QR من قبل الأستاذ.
- تسجيل بيانات الحضور.

#### المخرجات:

- حفظ بيانات الحضور في قاعدة البيانات.
- عرض التقارير.

# STUDENT ATTENDANCE

## مخطط تدفق البيانات Data Flow Diagram (DFD)



المستوى الأول (Level 1): DFD للمشروع

ال المستوى الأول (Level 1):

**عملية تسجيل الحضور:**

- الطالب يقوم بمسح رمز QR باستخدام هاتفه.
- يتم إرسال البيانات إلى الخادم للتحقق.
- يتم تسجيل الحضور في قاعدة البيانات.

**عملية توليد التقارير:**

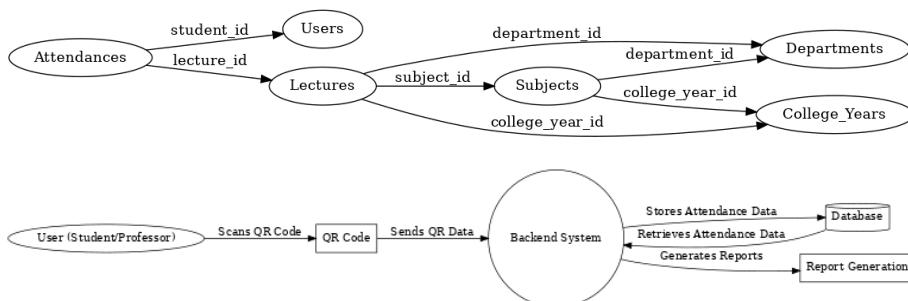
- الأستاذ أو المسؤول يطلب تقريراً من النظام.
- الخادم يستخرج البيانات من قاعدة البيانات.
- يعرض النظام التقرير عبر واجهة المستخدم.

**العناصر الأساسية:**

- مصدر البيانات: المستخدمون (طلاب/أساتذة).
- وجهة البيانات: قاعدة البيانات المركزية.
- التدفقات: البيانات تنتقل بين الأجهزة (الهواتف، الخوادم) وقاعدة البيانات.

# STUDENT ATTENDANCE

## مخطط تدفق البيانات Data Flow Diagram (DFD)



المخطط DFD للمشروع:

المستوى الأول (Level 1):

عملية تسجيل الحضور:

- الطالب يقوم بمسح رمز QR باستخدام هاتفه.
- يتم إرسال البيانات إلى الخادم للتحقق.
- يتم تسجيل الحضور في قاعدة البيانات.

عملية توليد التقارير:

- الأستاذ أو المسؤول يطلب تقريراً من النظام.
- الخادم يستخرج البيانات من قاعدة البيانات.
- يعرض النظام التقرير عبر واجهة المستخدم.

العناصر الأساسية:

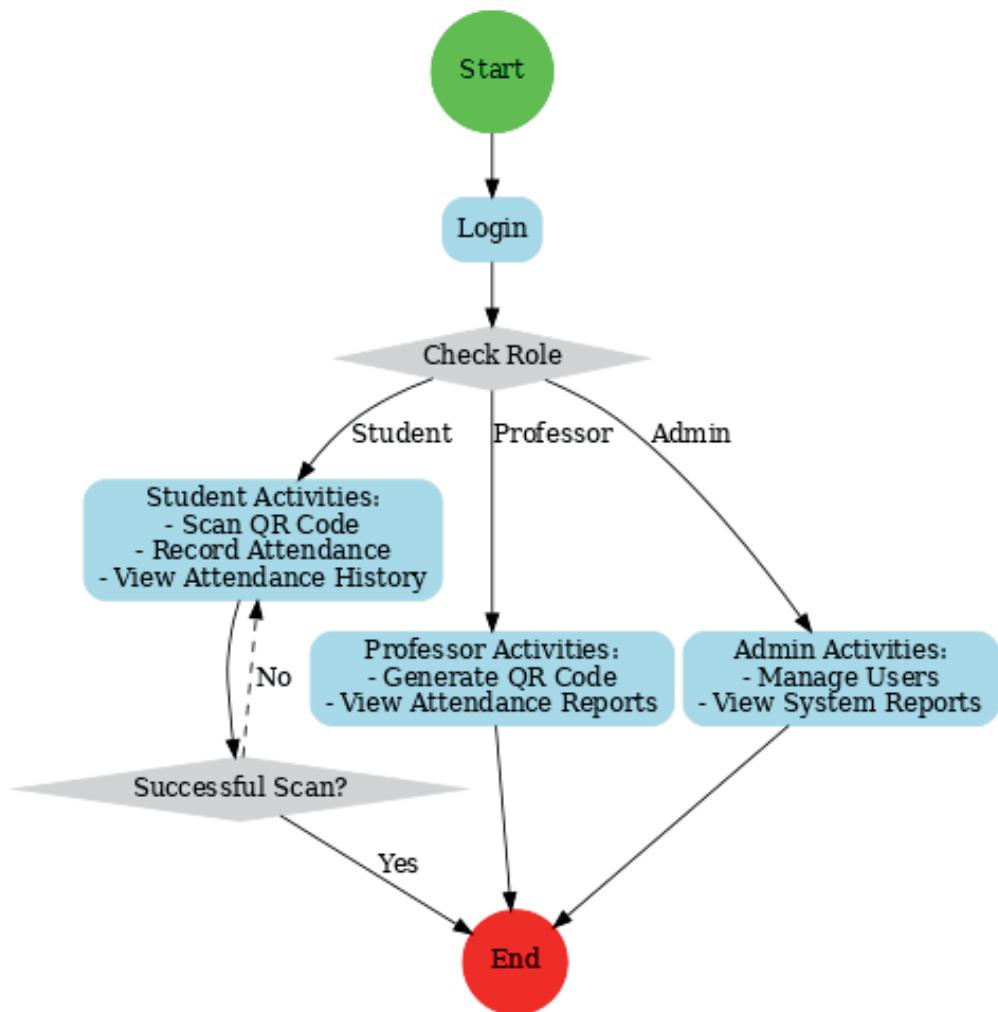
- مصدر البيانات: المستخدمون (طلاب/أساتذة).
- وجهة البيانات: قاعدة البيانات المركزية.
- التدفقات: البيانات تنتقل بين الأجهزة (الهواتف، الخوادم) وقاعدة البيانات.

# STUDENT ATTENDANCE

## Activity Diagram (AD) for Attendance Management System

### مفهوم المخطط:

مخطط النشاط (Activity Diagram) يُستخدم لتوضيح تسلسل الأنشطة والعمليات داخل النظام. يوضح المخطط كيف تبدأ العملية، ما هي الخطوات التي تتبعها، وكيف تنتهي.

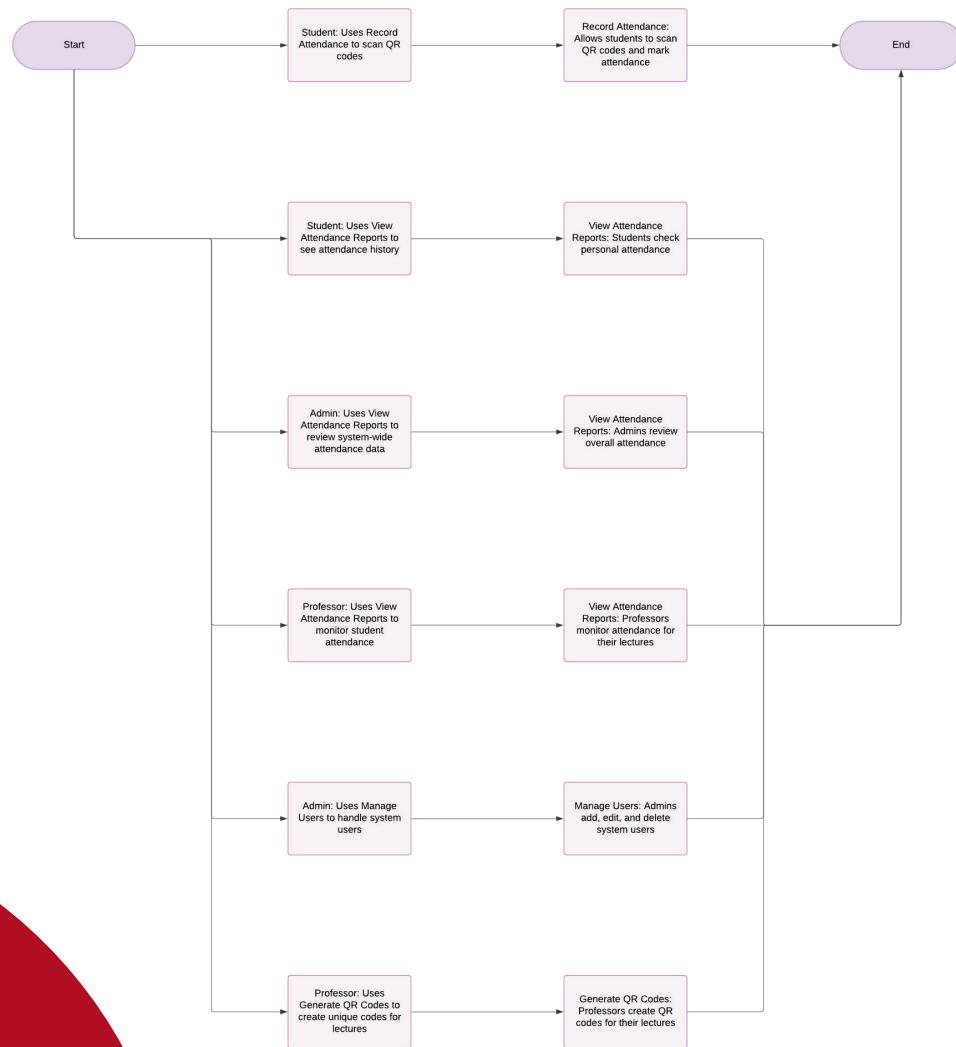
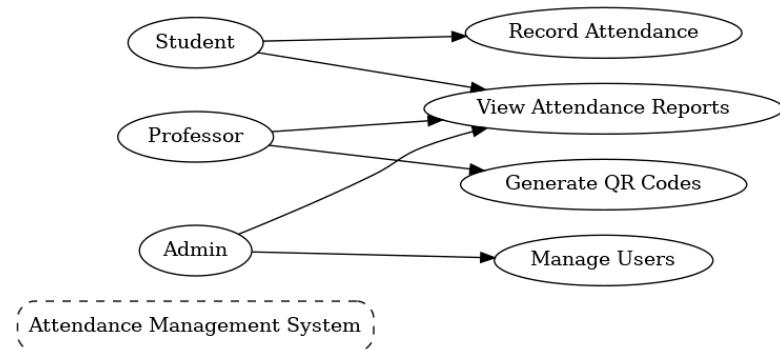


# STUDENT ATTENDANCE

## Use Case (UC) مخطط حالات الاستخدام

### مفهوم المخطط:

مخطط حالة الاستخدام (Use Case Diagram) يُستخدم لتوضيح العلاقة بين المستخدمين (Actors) وحالات الاستخدام (System) والنظام (Cases).



# STUDENT **ATTENDANCE**

**الباب الرابع**

# STUDENT ATTENDANCE

## Relational Schema Diagram مخطط المنهج

مخطط المنهج Relational Schema Diagram يمثل البنية المنطقية لقاعدة البيانات، ويوضح الكيانات (Entities) والعلاقات (Relationships) بينها، مع تحديد الخصائص (Attributes) التي تميز كل كيان.



# STUDENT ATTENDANCE

## إعداد قاموس البيانات

### 1. جدول Users

The screenshot shows the 'Table structure' view for the 'Users' table in MySQL Workbench. The table has 18 columns:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	<code>id</code>	<code>bigint(20)</code>		<code>UNSIGNED</code>	No	<code>None</code>		<code>AUTO_INCREMENT</code>	Change  Drop  More
2	<code>name</code>	<code>varchar(255)</code>	<code>utf8mb4_unicode_ci</code>		No	<code>None</code>			Change  Drop  More
3	<code>email</code>	<code>varchar(255)</code>	<code>utf8mb4_unicode_ci</code>		No	<code>None</code>			Change  Drop  More
4	<code>email_verified_at</code>	<code>timestamp</code>			Yes	<code>NULL</code>			Change  Drop  More
5	<code>role</code>	<code>enum('student', 'admin', 'professor', 'instructor')</code>	<code>utf8mb4_unicode_ci</code>		No	<code>student</code>			Change  Drop  More
6	<code>college_year_id</code>	<code>bigint(20)</code>		<code>UNSIGNED</code>	Yes	<code>NULL</code>			Change  Drop  More
7	<code>college_card_id</code>	<code>int(11)</code>			Yes	<code>NULL</code>			Change  Drop  More
8	<code>department_id</code>	<code>bigint(20)</code>		<code>UNSIGNED</code>	Yes	<code>NULL</code>			Change  Drop  More
9	<code>phone_number</code>	<code>varchar(255)</code>	<code>utf8mb4_unicode_ci</code>		Yes	<code>NULL</code>			Change  Drop  More
10	<code>password</code>	<code>varchar(255)</code>	<code>utf8mb4_unicode_ci</code>		No	<code>None</code>			Change  Drop  More
11	<code>two_factor_secret</code>	<code>text</code>	<code>utf8mb4_unicode_ci</code>		Yes	<code>NULL</code>			Change  Drop  More
12	<code>two_factor_recovery_codes</code>	<code>text</code>	<code>utf8mb4_unicode_ci</code>		Yes	<code>NULL</code>			Change  Drop  More
13	<code>two_factor_confirmed_at</code>	<code>timestamp</code>			Yes	<code>NULL</code>			Change  Drop  More
14	<code>remember_token</code>	<code>varchar(100)</code>	<code>utf8mb4_unicode_ci</code>		Yes	<code>NULL</code>			Change  Drop  More
15	<code>current_team_id</code>	<code>bigint(20)</code>		<code>UNSIGNED</code>	Yes	<code>NULL</code>			Change  Drop  More
16	<code>profile_photo_path</code>	<code>varchar(2048)</code>	<code>utf8mb4_unicode_ci</code>		Yes	<code>NULL</code>			Change  Drop  More
17	<code>created_at</code>	<code>timestamp</code>			Yes	<code>NULL</code>			Change  Drop  More
18	<code>updated_at</code>	<code>timestamp</code>			Yes	<code>NULL</code>			Change  Drop  More

# STUDENT ATTENDANCE

## إعداد قاموس البيانات

### 2. جدول Lectures

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change  Drop  More
2	topic	varchar(255)	utf8mb4_unicode_ci		No	None			Change  Drop  More
3	subject_id	bigint(20)		UNSIGNED	No	None			Change  Drop  More
4	created_by	bigint(20)		UNSIGNED	No	None			Change  Drop  More
5	start_time	datetime			No	None			Change  Drop  More
6	created_at	timestamp			Yes	NULL			Change  Drop  More
7	updated_at	timestamp			Yes	NULL			Change  Drop  More

### 3. جدول Attendances

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change  Drop  More
2	lecture_id	bigint(20)		UNSIGNED	No	None			Change  Drop  More
3	student_id	bigint(20)		UNSIGNED	No	None			Change  Drop  More
4	scanned_at	timestamp			Yes	NULL			Change  Drop  More
5	status	enum('present', 'absent', 'late')	utf8mb4_unicode_ci		No	present			Change  Drop  More
6	remarks	varchar(255)	utf8mb4_unicode_ci		Yes	NULL			Change  Drop  More
7	created_at	timestamp			Yes	NULL			Change  Drop  More
8	updated_at	timestamp			Yes	NULL			Change  Drop  More

# STUDENT ATTENDANCE

## إعداد قاموس البيانات

### 4. جدول Subjects

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change  Drop  More
2	name	varchar(255)	utf8mb4_unicode_ci		No	None			Change  Drop  More
3	description	text	utf8mb4_unicode_ci		Yes	NULL			Change  Drop  More
4	professor_id	bigint(20)		UNSIGNED	No	None			Change  Drop  More
5	department_id	bigint(20)		UNSIGNED	No	None			Change  Drop  More
6	college_year_id	bigint(20)		UNSIGNED	Yes	NULL			Change  Drop  More
7	instructor_id	bigint(20)		UNSIGNED	Yes	NULL			Change  Drop  More
8	created_at	timestamp			Yes	NULL			Change  Drop  More
9	updated_at	timestamp			Yes	NULL			Change  Drop  More

### 5. جدول Departments

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change  Drop  More
2	name	varchar(255)	utf8mb4_unicode_ci		No	None			Change  Drop  More
3	description	text	utf8mb4_unicode_ci		Yes	NULL			Change  Drop  More
4	created_at	timestamp			Yes	NULL			Change  Drop  More
5	updated_at	timestamp			Yes	NULL			Change  Drop  More

### 6. جدول College Years

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change  Drop  More
2	year_name	varchar(255)	utf8mb4_unicode_ci		No	None			Change  Drop  More
3	description	text	utf8mb4_unicode_ci		Yes	NULL			Change  Drop  More
4	created_at	timestamp			Yes	NULL			Change  Drop  More
5	updated_at	timestamp			Yes	NULL			Change  Drop  More

# STUDENT **ATTENDANCE**

## الباب الخامس

---

# STUDENT ATTENDANCE

## مقدمة

تنفيذ النظام تعتبر خطوة حاسمة في عملية تطوير أي مشروع، حيث يتم خلالها تحويل التصميمات الفنية والمواصفات إلى نظام حقيقي يمكن استخدامه. يهدف تنفيذ النظام إلى تطبيق المراحل المختلفة المحددة في عملية التطوير وضمان أن النظام يعمل بشكل صحيح وفقاً للاحتياجات والمواصفات المحددة.

### مكونات تنفيذ النظام:

#### 1. تركيب البرمجيات والأنظمة:

- يتضمن هذا الخطوة تثبيت البرمجيات المطلوبة والأنظمة المخصصة لتشغيل النظام بناءً على المتطلبات المحددة في مرحلة التصميم.

#### 2. تطوير واختبار البرمجيات:

- يتم تطوير الأكواد والمعكونات البرمجية واختبارها بشكل مستمر لضمان جودتها وأدائها الصحيح وفقاً للمتطلبات المحددة.

#### 3. قواعد البيانات:

- يشمل هذا الجانب إنشاء وتهيئة قواعد البيانات وضمان استقرارها وأمانها وفعاليتها في تخزين وإدارة البيانات.

#### 4. تدريب المستخدمين:

- يتم تدريب المستخدمين على استخدام النظام الجديد بشكل فعال وفقاً لأفضل الممارسات والإرشادات المحددة.

#### 5. توثيق النظام:

- يجب توثيق جميع العمليات والتعليمات المتعلقة بالنظام لضمان سهولة الصيانة والتحديث في المستقبل.

#### 6. اختبار النظام:

- يتم اختبار النظام بشكل شامل للتأكد من أنه يعمل بشكل صحيح وفقاً للمواصفات المحددة ولا يوجد به أخطاء أو عيوب.

# STUDENT ATTENDANCE

## Install The Working (بيئة العمل Environments)

**المتطلبات الأساسية:**

الخادم المحلي (Local Server) : VPS/VDS Server

نظام التشغيل: Ubuntu 20.04 أو أحدث.

**الأدوات المطلوبة:**



MySQL Server



أو أحدث PHP 8.0



Composer



Node.js



Laravel Framework

# STUDENT ATTENDANCE

## Install The Working (العمل Environments

### الخطوات التفصيلية:

1. تحديث النظام:

```
sudo apt update && sudo apt upgrade -y
```

2. تثبيت Apache :

```
sudo apt install apache2 -y
```

3. تثبيت MySQL :Server

```
sudo apt install mysql-server -y
```

# STUDENT ATTENDANCE

## Install The Working (بيئة العمل) (Environments)

**الخطوات التفصيلية:**

**تأمين تثبيت MySQL**

```
...  
sudo mysql_secure_installation
```

**أضف مستودع PHP**

```
...  
sudo apt install software-properties-common  
sudo add-apt-repository ppa:ondrej/php  
sudo apt update
```

**قم بتثبيت PHP 8.0 أو أحدث:**

```
...  
sudo apt install php8.0 libapache2-mod-php8.0 php8.0-mysql php8.0-curl php8.0-xml  
php8.0-mbstring php8.0-zip -y
```

# STUDENT ATTENDANCE

## Install The Working (بيئة العمل) Environments

الخطوات التفصيلية:

تثبيت Composer :

```
sudo apt install composer -y
```

أضف مستودع Node.js :

```
curl -fsSL https://deb.nodesource.com/setup_16.x | sudo -E bash -
```

قم بتثبيت Node.js :

```
sudo apt install nodejs -y
```

# STUDENT ATTENDANCE

## تثبيت بيئة العمل (Working Environment) لمشروع Laravel على Ubuntu

### الخطوات التفصيلية:

1. نسخ المشروع إلى الخادم:

```
scp -r /local/path/to/project user@server:/var/www/html/attendance-system
```

2. إعداد أدوات الملفات:

```
sudo chown -R www-data:www-data /var/www/html/attendance-system
sudo chmod -R 755 /var/www/html/attendance-system
```

3. إعداد بيئة العمل : (Environment)

```
cd /var/www/html/attendance-system
```

# STUDENT ATTENDANCE

## تثبيت بيئة العمل (Working Environment) لمشروع Laravel على Ubuntu

الخطوات التفصيلية:

قم بإنشاء ملف .env.

```
cp .env.example .env
```

قم بتحديث إعدادات قاعدة البيانات في .env.

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=attendance_system
DB_USERNAME=root
DB_PASSWORD=your_password
```

4. تثبيت التبعيات:

```
composer install
```

# STUDENT ATTENDANCE

## تثبيت بيئة العمل (The Working Environment) لمشروع Laravel على Ubuntu

**الخطوات التفصيلية:**

**تثبيت حزم npm:**

```
...  
npm install  
npm run build
```

**إنشاء مفتاح التطبيق:**

```
...  
php artisan key:generate
```

**إعداد قاعدة البيانات:**

```
...  
php artisan migrate
```

# STUDENT ATTENDANCE

## تثبيت بيئة العمل (Laravel) لمشروع Environments على Ubuntu

**الخطوات التفصيلية:**

قم بترحيل الجداول إلى قاعدة البيانات:

```
php artisan migrate --seed
```

:Apache خادم إعداد

```
<VirtualHost *:80>
    ServerName your-domain.com
    DocumentRoot /var/www/html/attendance-system/public

    <Directory /var/www/html/attendance-system>
        AllowOverride All
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

قم بتمكين الموقع:

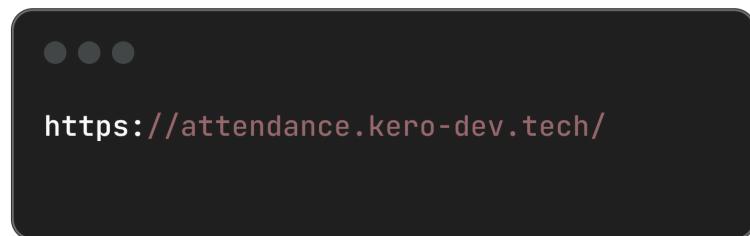
```
sudo a2ensite attendance-system
sudo systemctl reload apache2
```

# STUDENT ATTENDANCE

تثبيت بيئة العمل (Working Environment) لمشروع Laravel على Ubuntu

افتح متصفحك وانتقل إلى:

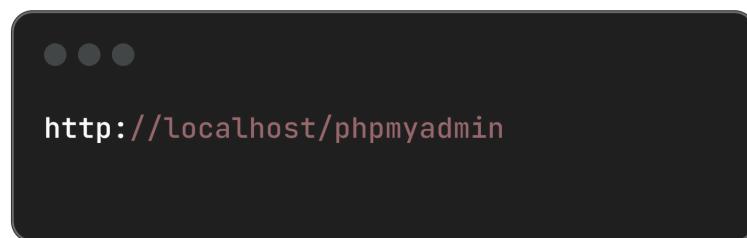
Domain Name



# STUDENT ATTENDANCE

## خطوات إنشاء قاعدة البيانات باستخدام sql ورفع ملف phpMyAdmin

تسجيل الدخول إلى  
:phpMyAdmin



إنشاء قاعدة بيانات جديدة:

انقر على "New" في الشريط الجانبي الأيسر.

أدخل اسم قاعدة البيانات (مثال: attendance\_system).

اختر الترميز utf8mb4\_general\_ci لضمان دعم النصوص المتععددة اللغات.

انقر على "Create".

# STUDENT ATTENDANCE

## خطوات إنشاء قاعدة البيانات باستخدام sql ورفع ملف phpMyAdmin

### رفع ملف sql.

اختر قاعدة البيانات التي أنشأتها (مثل attendance\_system). انقر على تبويب "Import" من القائمة العلوية. انقر على "Choose File" واختر ملف sql من جهازك. اضغط على "Go" لاستيراد الجداول والبيانات.

The screenshot shows the phpMyAdmin interface with the database 'attendance\_tracker' selected. On the left, a tree view lists various tables: attendances, cache, cache\_locks, college\_years, departments, failed\_jobs, jobs, job\_batches, lectures, migrations, password\_reset\_tokens, personal\_access\_tokens, sessions, subjects, and users. The main panel displays a table of 15 tables with their details:

Table	Action	Rows	Type	Collation	Size	Overhead
attendances	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	0	InnoDB	utf8mb4_unicode_ci	48.0 Kib	-
cache	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	12	InnoDB	utf8mb4_unicode_ci	16.0 Kib	-
cache_locks	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	0	InnoDB	utf8mb4_unicode_ci	16.0 Kib	-
college_years	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	4	InnoDB	utf8mb4_unicode_ci	16.0 Kib	-
departments	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	3	InnoDB	utf8mb4_unicode_ci	32.0 Kib	-
failed_jobs	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	0	InnoDB	utf8mb4_unicode_ci	16.0 Kib	-
jobs	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	0	InnoDB	utf8mb4_unicode_ci	16.0 Kib	-
job_batches	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	0	InnoDB	utf8mb4_unicode_ci	16.0 Kib	-
lectures	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	1	InnoDB	utf8mb4_unicode_ci	48.0 Kib	-
migrations	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	13	InnoDB	utf8mb4_unicode_ci	16.0 Kib	-
password_reset_tokens	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	0	InnoDB	utf8mb4_unicode_ci	16.0 Kib	-
personal_access_tokens	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	0	InnoDB	utf8mb4_unicode_ci	16.0 Kib	-
sessions	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	4	InnoDB	utf8mb4_unicode_ci	48.0 Kib	-
subjects	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	5	InnoDB	utf8mb4_unicode_ci	80.0 Kib	-
users	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	2	InnoDB	utf8mb4_unicode_ci	96.0 Kib	0 B
15 tables		Sum		44 InnoDB utf8mb4_0900_ai_ci 496.0 Kib		0 B

# STUDENT ATTENDANCE

## تصميم صفحات المشروع المختلفة مع الأكواد والصور و Github



حول المشروع على  
:GitHub

تم رفع مشروع Attendance Management System إلى GitHub لتوثيق الأكواد، الصور، والوثائق الخاصة بالمشروع، مما يسهل على الفريق الوصول إليه وإدارته.

رابط المستودع:

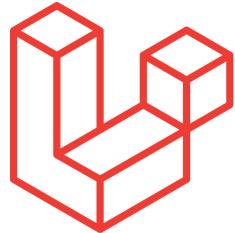
<https://github.com/ker00sama-dev/AttendanceTracker>

The screenshot shows the GitHub repository page for 'AttendanceTracker'. At the top, it displays 'AttendanceTracker' and 'Private'. It has 1 branch and 0 tags. There are buttons for 'Unwatch', 'Fork', 'Star', and settings. Below this, there's a search bar, a file browser, and a commit history table. The commit history table has columns for author, file, message, and time. On the right side, there are sections for 'About' (no description), 'Readme', 'Activity', 'Stars', 'Watching', 'Forks', 'Releases' (none), 'Packages' (none), 'Contributors' (2 users: ker00sama-dev and zsnakeee), and 'Languages' (none).

Author	File	Message	Time
ker00sama-dev	Update AttendanceQRController.php	Update AttendanceQRController.php	606b12c - last week
	app	frist commit	2 weeks ago
	bootstrap	upate	last week
	config	new db	2 weeks ago
	database	frist commit	2 weeks ago
	docker	new	last week
	lang	update	last week
	public	update	last week
	resources	update	last week
	routes	update	last week
	storage	frist commit	2 weeks ago
	tests	frist commit	2 weeks ago
	.editorconfig	frist commit	2 weeks ago
	.env	new	2 weeks ago
	.env.example	frist commit	2 weeks ago

# STUDENT ATTENDANCE

تصميم صفحات المشروع المختلفة مع  
الأكواد والصور و Github



بعض لقطات الأكواد:

لقطات ملفات المشروع بالكامل :

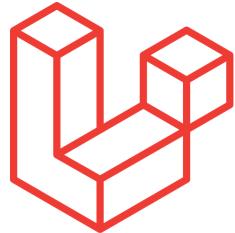
The screenshot shows a code editor with the following details:

- Project Structure:** The left sidebar shows the project structure under "attendancetracker". It includes "app", "routes", "storage", "tests", "vendor", and various configuration files like ".env", ".env.example", ".eslintrc.json", ".gitignore", ".prettierrc.json", and ".phpunit.xml".
- Code Editor:** The main area displays the content of `AttendanceQRController.php`. The code handles QR scanning and attendance recording.
- Code Content:**

```
12 class AttendanceQRController extends Controller
13 {
14     public function scan($id)
15     {
16         // Display success message and redirect to home
17         sweetalert()->success(__('Attendance recorded successfully!'));
18         return redirect()->route('home');
19     }
20     catch (\Exception $e) {
21         // Log the error for debugging
22         Log::error( message: 'Error in attendance scan: ' . $e->getMessage());
23     }
24 }
25
26 public static function doAttend($user_id, $lecture_id) : void
27 {
28     try {
29         // Validate lecture
30         $lecture = Lecture::find($lecture_id);
31         if (!$lecture || !$lecture instanceof \App\Models\Lecture) {
32             throw new \Exception(__('Lecture not found.'));
33         }
34
35         // Validate user
36         $user = User::find($user_id);
37         if (!$user) {
38             throw new \Exception(__('User not found.'));
39         }
40
41         // Current time and lecture start time
42         $scanned_at = Carbon::now();
43         $start_time = Carbon::parse($lecture->start_time);
44
45         // Check if the scan is within 1 hour of the lecture's start time Fawzi, 12/3/2024 2:39 AM - update
46         if ($scanned_at->diffInMinutes($start_time) > 60) {
47             throw new \Exception(__('Attendance scan time exceeded. You cannot scan after 1 hour.'));
48         }
49     }
50 }
```
- Bottom Status Bar:** Shows the file path as `attendancetracker/app/Http/Controllers/AttendanceQRController.php > AttendanceQRController > doAttend`, PHP version 8.2, Laravel 11.33, and other system information.

# STUDENT ATTENDANCE

## تصميم صفحات المشروع المختلفة مع الأكواد والصور و Github



Create Account Backend Code:

بعض لقطات الأكواد:

```
class CreateNewUser implements CreateNewUsers
{
    /**
     * Validate and create a newly registered user.
     *
     * @param array<string, string> $input
     */
    public function create(array $input): User
    {
        $input['phone_number'] = new PhoneNumber($input['phone_number'], country: 'EG');

        Validator::make($input, [
            'name' => [
                'required',
                'string',
                'max:255',
                'regex:/^(\S+\s+){3}\S+$/',
            ],
            'email' => ['required', 'string', 'email', 'max:255', 'unique:users'],
            'college_card_id' => ['required', 'integer', 'unique:users'],
            'department_id' => ['required', 'integer'],
            'college_year_id' => ['required', 'integer'],
            'password' => $this->passwordRules(),
            'terms' => Jetstream::hasTermsAndPrivacyPolicyFeature() ? ['accepted', 'required'] : '',
        ])->validate();

        return User::create([
            'name' => $input['name'],
            'email' => $input['email'],
            'college_card_id' => $input['college_card_id'],
            'phone_number' => $input['phone_number'],
            'password' => Hash::make($input['password']),
        ]);
    }
}
```

### Scan function:

```
AttendanceQRController PHP

public function scan($id)
{
    try {
        // Get the authenticated user's ID
        $user_id = auth()->id();

        Log::error('Error in attendance scan: ' . $id);

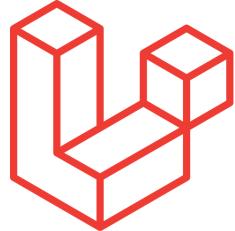
        // Attempt to record attendance
        $scan = self::doAttend($user_id, $id);

        // Display success message and redirect to home
        sweetalert()->success(__('Attendance recorded successfully!'));
        return redirect()->route('home');
    } catch (\Exception $e) {
        // Log the error for debugging
        Log::error('Error in attendance scan: ' . $e->getMessage());

        // Display error message and redirect to home
        sweetalert()->error($e->getMessage());
        return redirect()->route('home');
    }
}
```

# STUDENT ATTENDANCE

تصميم صفحات المشروع المختلفة مع  
الأكواد والصور و Github



بعض لقطات الأكواد:

## Attendance QR Controller: doAttend Function

```
AttendanceQRController
PHP

public static function doAttend($user_id, $lecture_id) : void
{
    try {
        // Validate lecture
        $lecture = Lecture::find($lecture_id);
        if (!$lecture || !$lecture instanceof \App\Models\Lecture) {
            throw new \Exception(__('Lecture not found.'));
        }

        // Validate user
        $user = User::find($user_id);
        if (!$user) {
            throw new \Exception(__('User not found.'));
        }

        // Current time and lecture start time
        $scanned_at = Carbon::now();
        $start_time = Carbon::parse($lecture->start_time);

        // Check if the scan is within 1 hour of the lecture's start time
        if ($scanned_at->diffInMinutes($start_time) > 60) {
            throw new \Exception(__('Attendance scan time exceeded. You cannot scan after 1 hour.'));
        }

        // Determine the attendance status
        $status = $scanned_at->diffInMinutes($start_time) <= 30 ? 'present' : 'late';

        // Check if the role is student
        if ($user->role !== 'student') {
            throw new \Exception(__('Only students can mark attendance.'));
        }

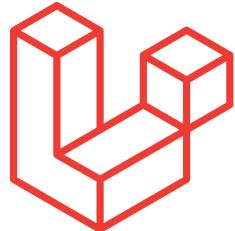
        // Check if attendance record already exists
        $attendance = Attendance::where('student_id', $user_id)->where('lecture_id', $lecture_id)->first();

        if ($attendance) {
            // You Are Already Marked Attendance
            throw new \Exception(__('You have already marked attendance.'));
        } else {
            // Create a new attendance record
            Attendance::create([
                'student_id' => $user_id,
                'lecture_id' => $lecture_id,
                'scanned_at' => $scanned_at,
                'status' => $status,
            ]);
        }
    } catch (\Exception $e) {
        // Log the error for debugging
        Log::error(__('Error in doAttend: :message', ['message' => $e->getMessage()]));
    }

    // Rethrow the exception with the localized message
    throw new \Exception(__('{$e->getMessage()}'));
}
```

# STUDENT ATTENDANCE

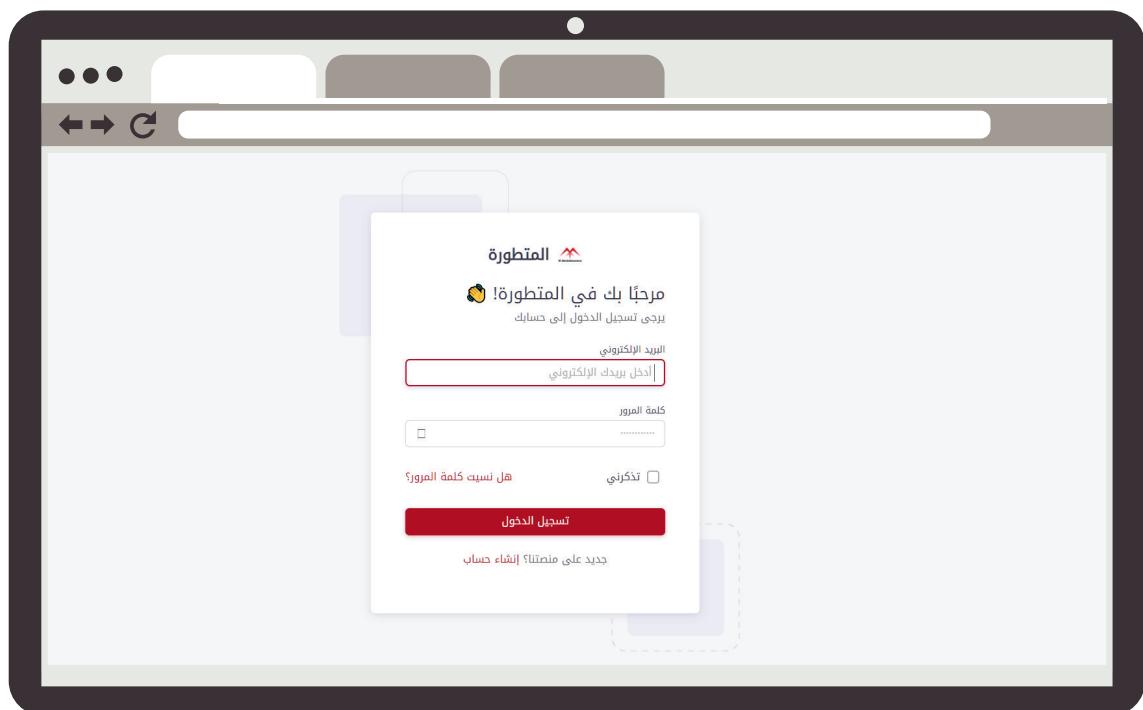
## تصميم صفحات المشروع المختلفة مع الأكواد والصور و Github



بعض لقطات الشاشات:

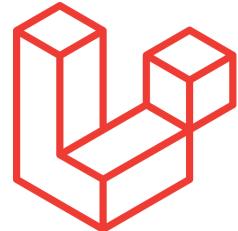
### Login Page

الوصف: تمكّن المستخدمين (طلاب، أساتذة، مشرفين) من تسجيل الدخول.



# STUDENT ATTENDANCE

## تصميم صفحات المشروع المختلفة مع الأكواد والصور و Github



بعض لقطات الشاشات:

### Register Page

الوصف: صفحة تسجيل المستخدمين الجدد في النظام، سواء كانوا طلاباً أو أستاذة أو إداريين.

المتطورة

انضم إلينا اليوم لمستقبل أكثر إشراقاً!

قم بإنشاء حساب طالب الخاص بك وإندا في استكشاف عالم من المعرفة.

الاسم بالكامل (إرجاعي عربي)  
أدخل اسمك الكامل

البريد الإلكتروني  
أدخل عنوان بريدك الإلكتروني

رقم بطاقه الائتمانية يجب أن يكون صحيفاً للتسجيل الدخلي  
أدخل رقم بطاقة الائتمان

الفرقة الدراسية  
اختر الفرقه الدراسية

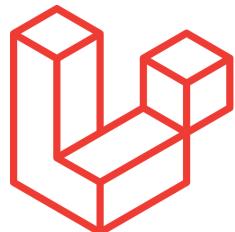
الفصل  
اختر الفصل الدراسي بك

رقم الهاتف (بيان)  
01000000000  
01000000000

كلمة المرور

# STUDENT ATTENDANCE

## تصميم صفحات المشروع المختلفة مع الأكواد والصور و Github

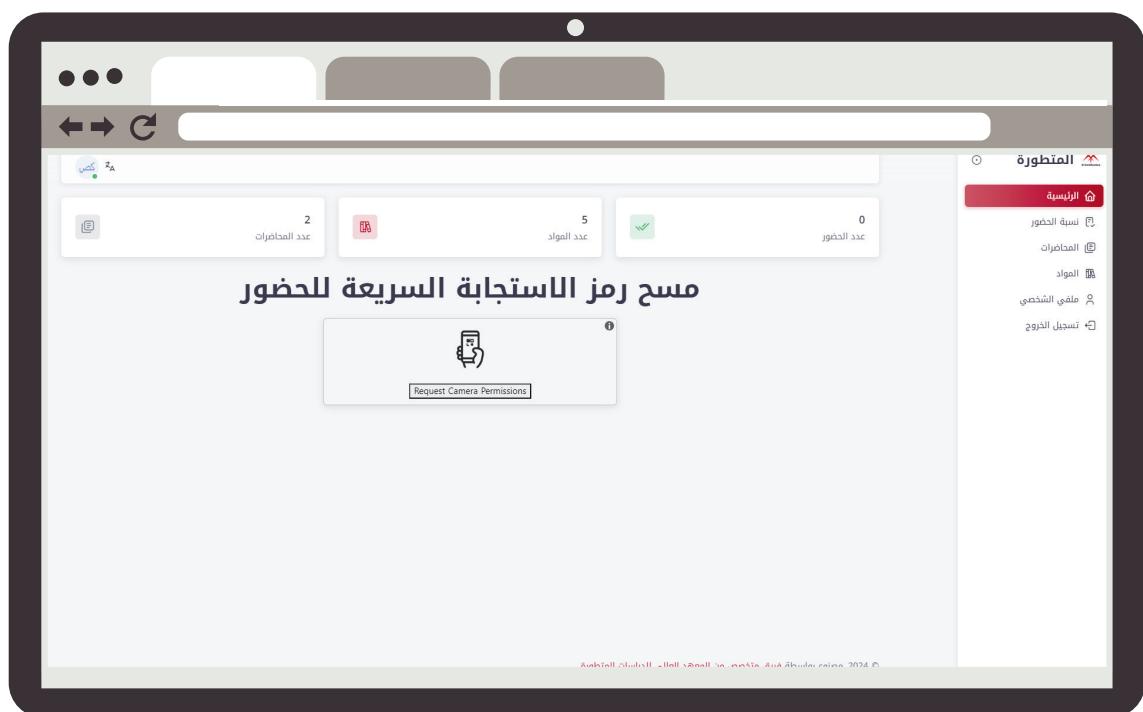


بعض لقطات الشاشات:

### مقدمة:

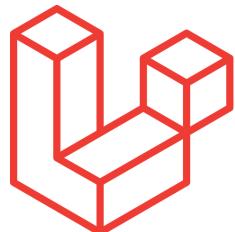
واجهة المستخدم في النظام تم تصميمها لتكون سهلة الاستخدام وبديهية، مع مراعاة احتياجات الطلاب والأساتذة والإداريين. تم استخدام تقنيات حديثة مثل Bootstrap v5 لضمان توافق الواجهة مع مختلف الأجهزة والشاشات.

### User Interface



# STUDENT ATTENDANCE

## تصميم صفحات المشروع المختلفة مع الأكواد والصور و Github



بعض لقطات الشاشات:

### مقدمة:

واجهة الادارة هي قسم خاص في النظام يتيح للمسؤولين إدارة النظام بكفاءة وسهولة. تم تصميمها لتكون مرنة وقابلة للتكييف مع المهام الإدارية.

### Admin Interface



# STUDENT ATTENDANCE

## النشر والاختبار

تم نشر المشروع بنجاح على خادم يديره كيرلوس أسامة، ويمكن الوصول إلى النظام من خلال النطاق  
<https://attendance.kero-dev.tech>

تم النشر لضمان وصول النظام للختبار وأغراض العرض. تم إجراء اختبارات صارمة، بما في ذلك اختبارات الوحدات لكل وحدة واختبارات المستخدم لضمان تجربة استخدام سلسة.

