

كلية الهندسة

قسم الهندسة المعلوماتية

هندسة أمن النظم والشبكات الحاسوبية



Secure Networked Software Architecture for University Electronic-Attendance Systems

بنية شبكية برمجية آمنة لنظام الحضور والتفقد الإلكتروني للجامعات

إعداد :

بيان محمد سفيان النصاري

تala فراس المسالمه

سوق علي الخشمان

باشراف:

م.محمد يامن الحلاق

بنيار - 2026

الملخص

مع التطور المتتسارع في تقنيات التحول الرقمي واعتماد المؤسسات التعليمية على الأنظمة الذكية، برزت الحاجة إلى إيجاد حلول تقنية فعالة لمعالجة المشكلات المرتبطة بالأنظمة التقليدية لن فقد الحضور، مثل إهدر الوقت، وقابلية التلاعب، وعدم الدقة في تسجيل البيانات. يهدف هذا المشروع إلى تصميم وتنفيذ نظام تفقد حضور إلكتروني جامعي ذكي يعتمد على تقنية التعرف على الوجه (Face Recognition)، مدعوماً بنية شبكة وبرمجية متكاملة تضمن الدقة، السرعة، وأمن البيانات.

يقوم النظام المقترن على التقاط صور الطلبة داخل القاعات الدراسية باستخدام أجهزة مخصصة، ثم معالجة هذه الصور ومقارنتها مع قاعدة بيانات مركبة تحتوي على بيانات الوجوه المسجلة مسبقاً، ليتم تسجيل الحضور تلقائياً دون تدخل بشري. تم تصميم البنية البرمجية للنظام بحيث تضمن سهولة الإدارء، وقابلية التوسيع، والتكامل مع الأنظمة الجامعية الأخرى، في حين تم بناء بنية شبكة محاكية لبيئة جامعة حقيقة باستخدام أدوات محاكاة متقدمة تتيح ربط أجهزة القاعات مع الخادم центральный بشكل آمن وفعال

Abstract

With the rapid advancement of digital transformation technologies and the increasing adoption of smart systems in educational institutions, there has been a growing need for effective technological solutions to address the limitations of traditional attendance methods. Conventional attendance systems, such as manual sign-in sheets or verbal roll calls, are often time-consuming, prone to human error, and vulnerable to manipulation. This project aims to design and implement a smart electronic attendance system for universities based on face recognition technology, supported by a secure and integrated network and software architecture.

The proposed system operates by capturing students' facial images within lecture halls and automatically matching them against a centralized database containing pre-registered facial data. Upon successful identification, attendance records are generated without human intervention. The software architecture of the system is designed to ensure scalability, reliability, and ease of management, while the network architecture is implemented to securely connect classroom devices to a central server, ensuring efficient data transmission and protection of sensitive information.

Table of Contents

10	1.1 مقدمة الفصل
11	1.2 المشكلة العلمية
11	1.3 الهدف
12	1.4 أهمية المشروع
12	1.5 المراجعة الأدبية
12	1.5.1 تطور أنظمة تفقد الحضور في المؤسسات التعليمية
12	1.5.2 أنظمة الحضور المعتمدة على التعرف على الوجه
13	1.5.3 البنية البرمجية لأنظمة الحضور الذكية
13	1.5.4 البنية الشبكية وأمن أنظمة الحضور الإلكترونية
14	1.6 التحديات
15	1.7 التطبيقات العملية
16	الفصل الأول
17	2.1 الأساس النظري
17	2.2 التعرف على الوجه (Face Recognition)
18	2.3 كشف الوجه والتعرف عليه (Face Detection vs Face Recognition)
18	2.4 الأنظمة المعتمدة على واجهات البرمجة (API-Based Systems)
19	2.5 البنية متعددة الطبقات (Three-Tier Architecture)
19	2.6 منطقة العزل الشبكي (Demilitarized Zone – DMZ)
21	الفصل الثاني
22	3.1 نظرة عامة على الفصل

22	3.2 معمارية النظام
23	3.3 التقنيات والأدوات المستخدمة
25	3.4 مكونات النظام (تفكيك الوحدات)
27	3.5 تصميم قاعدة البيانات
28	3.6 خدمة التعرف على الوجه
29	3.7 عملية تسجيل الطالب (Enrollment)
29	3.8 التحقق من الحضور (Verification)
30	3.9 التفويض والتحكم بالوصول (Authorization)
31	3.10 التدقيق والمحاسبة (Accounting & Auditing)
32	3.11 واجهات النظام ومسارات الصفحات وواجهات API
32	3.12 اعتبارات الأمان والخصوصية
34	3.13 سيناريو العرض العملي (Demo)
34	3.14 القيود والتحسينات المستقبلية
34	3.15 الخلاصة
35	الفصل الثالث
36	3.1 تصميم الشبكة
37	3.2 المكونات الشبكية وأدوارها
37	3.3 التقسيم الشبكي المعتمد (Segmentation Model)
38	3.4 منهجية التنفيذ والربط داخل VMware و GNS3

38	3.5 العنونة Addressing
38	3.5.1 منهجية تصميم خطة العنونة
39	3.5.2 الشبكات المعتمدة
40	3.5.3 المنطق في توزيع العنوانين
40	3.5.4 ضبط العنونة على الأجهزة الافتراضية
41	3.5.5 التحقق من ثبات العنونة بعد إعادة التشغيل
41	3.6 التحقق من الاتصال (Connectivity Verification) واختبارات التشغيل الشبكي
41	3.6.1 التتحقق من حالة واجهات الموجه (Router Interface Verification)
42	3.6.2 التتحقق من الاتصال داخل كل شبكة إلى البوابة (Gateway Reachability)
42	3.6.3 التتحقق من جداول ARP على الراوتر (L2/L3 Validation via ARP)
43	3.6.4 اختبار التوجيه بين الشبكات (Inter-Subnet Routing)
43	3.6.5 التتحقق من مسار الخدمة (Service Path Validation)
44	3.7 سياسات التحكم بالوصول (ACL) وآلية التطبيق والتحقق
44	3.7.1 أهداف سياسة الـ ACL
44	3.7.2 منطق السياسة (Policy Logic) حسب المناطق
45	3.7.3 منهجية التطبيق المرحلية (Strict ثم Permissive)
46	3.7.4 اختبارات التحقق النهائية المرتبطة
47	3.7.5 ربط السياسة بالنتيجة الوظيفية النهائية (تسجيل الحضور)
48	3.8 التحقق النهائي من الهدف الوظيفي: تمكين الطالب من تسجيل الحضور عبر مسار شبكي آمن
48	3.8.1 تعريف سيناريو الاستخدام النهائي (End-to-End Scenario)

48	3 التحضير للاختبار النهائي (Prerequisites)
49	3 تصميم مسار الخدمة (Service Path) كمتطلب تحقق
49	3 خطوات التحقق العملي ونتائجها المتوقعة (Tests & Expected Results)
51	3 تفسير النتائج وربطها بأهداف الأمن والوظيفة
51	3 نتائج القسم الشبكي (Network Section Outcome)
52	الفصل الرابع
53	4 نتائج البنية البرمجية للنظام
55	4 نتائج البنية الشبكية والأمنية
56	4 نتائج التسجيل المركزي للأحداث والتدقيق
56	4 مقارنة النتائج مع أهداف المشروع
56	4 التحديات التي ظهرت أثناء التنفيذ
57	الخاتمة
58	قائمة المصطلحات
59	قائمة الاختصارات (List of Abbreviations)
60	References

List of Tables

جدول 1.1 : اهم مقالات الدراسة في أنظمة تفقد الحضور الالكتروني 14
جدول 2.1 : مجموعة التقنيات (تنفيذ النموذج الأولي) 25
جدول 2.2 : المكونات الرئيسية ومسؤولياتها في (Django) 27
جدول 2.3: الجداول الأساسية والحقول الرئيسية 29
جدول 2.4: أمثلة على نقاط النهاية المستخدمة في النموذج الأولي 33
جدول 2.5: ضوابط الأمان المطبقة أو التي تم النظر فيها 35
جدول 3.1 : توزيع العناوين 41

List of Forms

الشكل 1.1 : المخطط العام لآلية عمل نظام التعرف على الوجه	19
الشكل 1.2: الفرق بين كشف الوجه و التعرف على الوجه.....	19
الشكل 1.3 : البنية متعددة الطبقات (three-tier architecture)	20
الشكل 1.4 : نموذج بنية شبكية باستخدام DMZ	21
الشكل 2.1 : نظرة عامة على بنية النظام	24
الشكل 2.2 : تصميم قاعدة البيانات و علاقات الجداول	28
الشكل 2.3: تسلسل عملية التسجيل	30
الشكل 2.4: تسلسل التحقق من الحضور	31
الشكل 2.5: مخطط تدفق قرار التفويض	32
الشكل (3.1) : الشبكة	38
الشكل 3.2 : الشبكة باستخدام GNS3	41
الشكل 3.3 : اعدادات nginx داخل netplan (مثال)	42
الشكل 3.4 : واجهات الراوتر up	43
الشكل 3.5 : تنفيذ اختبار ping من nginx نحو بوابته 10.10.200.1 (مثال)	44
الشكل 3.6 : تحديد الأجهزة المتصلة بالشبكة	44
الشكل (4.1): نتيجة التعرف على الوجه و تسجيل الحضور بنجاح	55
الشكل (4.2): نتيجة عملية التتحقق (Verification) و تسجيل الحضور مع قيمة الثقة	56
الشكل (4.3): نتيجة عملية التتحقق (Verification) مع فشل التفويض (Authorization Failure)	56
الشكل (4.4): مثال على مخرجات نظام الحضور الإلكتروني (IN / FORBIDDEN)	57

1.1 مقدمة الفصل

شهد قطاع التعليم العالي خلال السنوات الأخيرة تطوراً متسارعاً نتيجة النboom الكبير في تقنيات المعلومات والاتصالات، الأمر الذي دفع الجامعات والمؤسسات التعليمية إلى تبني أنظمة إلكترونية حديثة تهدف إلى تحسين جودة العملية التعليمية والإدارية على حد سواء. وقد أصبح الاعتماد على الحلول الرقمية ضرورة ملحة لمواجهة التحديات المتزايدة المرتبطة بإدارة أعداد كبيرة من الطلبة، وضمان دقة البيانات، وتسريع الإجراءات الإدارية المختلفة. ويأتي هذا التوجه ضمن إطار التحول الرقمي في التعليم العالي، الذي أشارت إليه عدة دراسات باعتباره عاملاً رئيسياً في تحسين كفاءة المؤسسات التعليمية ورفع مستوى الاعتمادية في خدماتها الأكademية والإدارية [1][2].

تعد عملية فقد حضور الطلبة من العناصر الأساسية في النظام الأكاديمي الجامعي، لما لها من تأثير مباشر على تقييم التزام الطلبة بالمساقات الدراسية، ومتابعة انتظامهم داخل القاعات الدراسية، واتخاذ القرارات الأكاديمية المناسبة. إلا أن الأساليب التقليدية المستخدمة في تسجيل الحضور، مثل التوقيع اليدوي أو النداء الشفهي، تعاني من العديد من أوجه القصور، من أبرزها استهلاك وقت المحاضرة، وزيادة العبء الإداري على أعضاء الهيئة التدريسية، إضافة إلى احتمالية وقوع أخطاء بشرية أو حالات تلاعب وانتفال شخصية. وقد أكدت دراسات متعددة أن هذه الأساليب تفتقر إلى الدقة والموثوقية، خصوصاً في البيانات التعليمية ذات الأعداد الكبيرة من الطلبة [3][4].

ومع التطور الملحوظ في مجال الذكاء الاصطناعي ومعالجة الصور، برزت تقنيات القياسات الحيوية كحلول تقنية منقولة لمعالجة هذه المشكلات، حيث تُعد تقنية التعرف على الوجه من أكثر هذه التقنيات شيوعاً وانتشاراً في الأنظمة التعليمية الحديثة. ويعود ذلك إلى ما توفره من سهولة في الاستخدام، ودقة عالية في التحقق من الهوية، وعدم الحاجة إلى تفاعل مباشر من المستخدم أو استخدام أدوات إضافية. وقد بينت دراسات حديثة أن أنظمة الحضور المعتمدة على التعرف على الوجه تسهم بشكل كبير في تقليل حالات التلاعب وتحسين موثوقية سجلات الحضور [5][6]. إلا أن هذه الدراسات تشير أيضاً إلى أن نجاح هذه الأنظمة يتطلب تصميم بنية برمجية منظمة ومتكلمة مع بنية شبكة آمنة، تضمن حماية البيانات الحساسة ومنع الوصول غير المصرح به، وهو ما يشكل الأساس العلمي لهذا المشروع [7][8].

1.2 المشكلة العلمية

تركز العديد من الدراسات على تطوير أنظمة حضور تعتمد على التعرف على الوجه من الناحية البرمجية، مع التركيز على دقة الخوارزميات وسرعة المعالجة [4]. إلا أن جزءاً كبيراً من هذه الأنظمة يتم نشره ضمن بيئات شبكية غير مدرورة، حيث يتم كشف خوادم التطبيقات أو قواعد البيانات مباشرة للمستخدمين، مما يزيد من المخاطر الأمنية ويجعل النظام عرضة للهجمات أو إساعه الاستخدام [5].

تتمثل المشكلة العلمية في الحاجة إلى تصميم نظام حضور إلكتروني لا يقتصر على دقة التعرف على الوجه فقط، بل يعتمد على بنية برمجية منتظمة ومتكلمة مع بنية شبكة آمنة تضمن الفصل بين المستخدمين والخدمات الحساسة، وتدعم مبدأ أقل صلاحية، وتتوفر قابلية التتبع والتدقيق [6].

1.3 الهدف

يهدف هذا المشروع إلى تصميم وتنفيذ بنية شبكة برمجية آمنة لنظام تفقد الحضور الإلكتروني في الجامعات، بما يحقق تكاملاً فعالاً بين الجوانب البرمجية والشبكية للنظام. ويسعى المشروع إلى بناء نظام حضور إلكتروني يعتمد على تقنية التعرف على الوجه للحد من حالات التلاعب والاحتلال وضمان موثوقية تسجيل الحضور . كما يهدف إلى تصميم بنية برمجية خدمية (Service-Oriented Architecture) تعتمد على واجهات برمجية (APIs) لفصل واجهة المستخدم عن منطق النظام، بما يسهم في تحسين القابلية للتوسع وسهولة الصيانة والتطوير المستقبلي .

ومن الناحية الشبكية، يهدف المشروع إلى تصميم بنية متعددة الطبقات (Three-Tier Architecture) تضمن عزل المستخدمين عن خوادم التطبيقات وقواعد البيانات الحساسة، وتعزز مستوى الأمان العام للنظام . ويشمل ذلك استخدام منطقة منزوعة السلاح (DMZ) كنقطة دخول وحيدة للنظام بهدف حماية الموارد الداخلية ومنع الوصول المباشر إليها . كما يسعى المشروع إلى تطبيق سياسات التحكم بالوصول باستخدام قوائم التحكم (Access Control Lists) وفق مبدأ أقل صلاحية (Least Privilege) للحد من سطح الهجوم وتقليل المخاطر الأمنية . إضافة إلى ذلك، يهدف النظام إلى دعم آليات التسجيل المركزي للأحداث

(Centralized Logging) بما يعزز قابلية المراقبة والتدقيق وتتبع الأنشطة، ويساعد في الكشف المبكر عن السلوكيات غير الطبيعية وتحليل الحوادث الأمنية المحتملة .

1.4 أهمية المشروع

تكمّن أهمية هذا المشروع في كونه يجمع بين ثلاثة مجالات رئيسية: الذكاء الاصطناعي، هندسة البرمجيات، وهندسة الشبكات. فقد أظهرت دراسات حديثة أن أنظمة التعرف على الوجه في التعليم تسهم في تحسين دقة وموثوقية تسجيل الحضور [13]، إلا أن غياب التصميم الشبكي الآمن يُعد من الأسباب الرئيسية لفشل هذه الأنظمة عند نشرها فعليًا [14].

كما يقدم المشروع نموذجًا أكاديميًّا وتطبيقيًّا يوضح أن تحقيق الأمان لا يعتمد فقط على استخدام أدوات أمنية متقدمة، بل يمكن تحقيقه من خلال التصميم الصحيح والتكامل بين البنية البرمجية والبنية الشبكية، وهو ما يعزز مفهوم الأمان القائم على المعمارية .[15](Architecture-Based Security)

1.5 المراجعة الأدبية

1.5.1 تطور أنظمة تفقد الحضور في المؤسسات التعليمية

شهدت أنظمة تفقد الحضور في المؤسسات التعليمية تحولاً ملحوظاً من الأساليب التقليدية اليدوية إلى الأنظمة الإلكترونية، وذلك بهدف تحسين دقة البيانات وتقليل الاعتماد على العامل البشري. فالطرق التقليدية مثل التوقيع اليدوي أو النداء الشفهي تستهلك وقتاً كبيراً من المحاضرات، كما أنها عرضة للأخطاء البشرية وحالات التلاعُب والانتهال، خاصة في الجامعات ذات الأعداد الكبيرة من الطلبة. وقد أظهرت الدراسات أن هذه الأساليب تؤدي إلى زيادة العبء الإداري على أعضاء الهيئة التدريسية، إضافة إلى صعوبة حفظ سجلات الحضور واسترجاعها وتحليلها بشكل فعال، مما دفع المؤسسات التعليمية إلى البحث عن حلول تقنية أكثر كفاءة لإدارة الحضور [1][2].

1.5.2 أنظمة الحضور المعتمدة على التعرف على الوجه

مع التقدم في مجال الذكاء الاصطناعي ومعالجة الصور، برزت تقنيات القياسات الحيوية كحلول فعالة لمعالجة مشكلات أنظمة الحضور التقليدية، وتُعد تقنية التعرف على الوجه من أكثر هذه التقنيات استخداماً في البيانات التعليمية. وبعود ذلك إلى قدرتها على التحقق من هوية الأفراد دون الحاجة إلى أدوات إضافية أو تفاعل مباشر من المستخدم، مما يقلل من حالات التلاعُب والانتهال. وقد بينت دراسات عديدة أن أنظمة الحضور المعتمدة على التعرف على الوجه تسهم في تحسين موثوقية تسجيل الحضور ورفع دقتها،

الفصل الأول

خصوصاً في القاعات الدراسية الذكية، إلا أنها أشارت أيضاً إلى وجود تحديات تتعلق بالإضاءة وزوايا التصوير وجودة الكاميرات. [5][4][3]

1.5.3 البنية البرمجية لأنظمة الحضور الذكية

تؤكد الأدبيات المتعلقة بـهندسة البرمجيات على أهمية تصميم بنية برمجية واضحة ومنظمة عند تطوير الأنظمة الذكية، لما لذلك من دور في تحسين القابلية للصيانة والتتوسيع وضمان استقرار النظام. وتشير الدراسات إلى أن اعتماد معماريات برمجية خدمية تعتمد على واجهات برمجية (APIs) يسهم في فصل واجهة المستخدم عن منطق الأعمال، ويتتيح سهولة التكامل مع أنظمة أخرى مستقبلاً. كما أن هذا النوع من البنى البرمجية يساعد في تطبيق ضوابط أمنية بشكل أفضل، خاصة عند التعامل مع بيانات حساسة مثل البيانات البيومترية، وهو ما يجعل البنية البرمجية عنصراً أساسياً في نجاح أنظمة الحضور الإلكترونية. [6][7].

1.5.4 البنية الشبكية وأمن أنظمة الحضور الإلكترونية

تشير الأدبيات في مجال أمن المعلومات إلى أن نجاح الأنظمة الإلكترونية لا يعتمد على الجانب البرمجي فقط، بل يتطلب أيضاً تصميم بنية شبكة آمنة تضمن حماية الموارد والخدمات الحساسة. ويعود التقسيم الشبكي وعزل المكونات الداخلية من المبادئ الأساسية للحد من المخاطر الأمنية، حيث تسهم البنى متعددة الطبقات في تقليل سطح الهجوم ومنع الوصول غير المصرح به. كما تؤكد الدراسات على أهمية استخدام مناطق منزوعة السلاح (DMZ) كنقطة دخول وحيدة للأنظمة، وتطبيق سياسات التحكم بالوصول وفق مبدأ أقل صلاحية، إضافة إلى دور التسجيل المركزي للأحداث في تعزيز المراقبة والتدقيق والاستجابة للحوادث الأمنية. [8][9][10].

جدول 1.1: أهم مقالات الدراسة في أنظمة تفقد الحضور الإلكترونية

SL.No.	Pub Year	Author(s)	Title	Pub Title
1	2024	Ashish Jadhav; Dhwaniket .Kamble; Santosh B ;Rathod; Sumita Kumar Mohammad Dalwai	Attendance management system using face recognition	IAES Int. Journal of Artificial Intelligence
2	2024	:K. Lavanya; N. Akalyaa .S. Archana; J Pushpakala	Face Recognition-Based Attendance System	Shanlax International Journal of Arts, Science & Humanities &
3	2025	Siti Aisyah	Development of a Smart Attendance System Using Face Recognition Technology	ICISTech Journal
4	2024	;Kamal G. Oladele Edidiong J. Iseh; Jamal .O. Oladele; Pawan R	AI-Based Face Recognition Attendance System	International Journal of Innovative Research in Multidisciplinary

الفصل الاول

		Bhaladhare		Professional Studies (IJIRMPS)
5	2023	Phul Babu Jha; Arjun ;Basnet; Biraj Pokhrel Bishnu Pokhrel; Gopal Kumar Thakur; Surya Chhetri	An Automated Attendance System Using Facial Detection and Recognition Technology	Apex Journal of Business and Management
6	2025	;Divyanshu Bhardwaj Nikita; Vernika; Aarush Gupta; Tanya Chauhan	Smart Attendance System Using Face Recognition with OpenCV	Journal of Mobile ,Computing & Communication Mobile Networks
7	2024	S. A. Feroze	The Facial Recognition Technology in Academic Attendance	International Journal of Innovative Technology and Management
8	2022	.J. Patel et al	Facial Recognition Attendance System with Anti-Spoofing	Multimedia Tools and Applications
9	2023	.R. Kumar et al	Privacy-Aware Face Recognition Attendance System	IEEE Access
10	2023	.M. Alqahtani et al	AI-Based Attendance System for Higher Education	Education and Information Technologies
11	2024	.Y. Zhang et al	Robust Face Recognition in Uncontrolled Environments	Computer Vision and Image Understanding
12	2024	.S. Rose et al	Zero Trust Architecture for Distributed Web Applications	ACM Computing Surveys

1.6 التحديات

تشير الدراسات الحديثة إلى أن أنظمة تفقد الحضور الإلكتروني المعتمدة على تقنية التعرف على الوجه تواجه مجموعة من التحديات التقنية والعملية عند تطبيقها في البيئات التعليمية الواقعية. من أبرز هذه التحديات تغير ظروف الإضاءة داخل القاعات الدراسية، حيث أوضحت بعض الدراسات أن ضعف الإضاءة أو عدم تجانسها يؤدي إلى انخفاض جودة الصور الملتقطة، مما يؤثر سلباً على دقة نماذج التعرف على الوجه، خاصة في البيئات غير المضبوطة. كما تُعد زوايا التصوير المختلفة وحركة الطلبة أثناء الدخول أو الجلوس من العوامل التي تزيد من صعوبة استخراج الخصائص البيومترية بدقة .

1.7 التطبيقات العملية

خلال السنوات الأخيرة، ولا سيما منذ عام 2020، شهدت أنظمة تفقد الحضور الإلكتروني المعتمدة على تقنية التعرف على الوجه تطوراً ملحوظاً في التطبيقات العملية داخل المؤسسات التعليمية، حيث ركزت العديد من الدراسات على تحويل النماذج النظرية إلى أنظمة قابلة للتطبيق في البيئات الجامعية الواقعية. وقد أظهرت هذه التطبيقات قدرة عالية على أتمتة عملية تسجيل الحضور بدقة وكفاءة، مع تقليل الاعتماد على الأساليب التقليدية التي تعاني من التلاعيب والأخطاء البشرية.

في هذا السياق، طُبقت أنظمة الحضور المعتمدة على التعرف على الوجه في القاعات الدراسية الذكية، حيث يتم التقاط صور الطلبة تلقائياً عند دخولهم القاعة أو أثناء المحاضرة، ثم معالجتها باستخدام نماذج تعلم عميق لاستخراج الخصائص البيومترية ومقاربتها مع قواعد البيانات المخزنة مسبقاً. وأسهم هذا الأسلوب في تسريع عملية تسجيل الحضور، وتقليل الوقت المهدور، ورفع موثوقية السجلات الأكademية.

كما امتدت التطبيقات العملية لهذه الأنظمة لتشمل البيانات الجامعية واسعة النطاق، مثل الامتحانات الإلكترونية والتحقق من هوية الطلبة، إضافة إلى دمجها مع أنظمة إدارة التعلم والمنصات التعليمية. وأظهرت الدراسات أن الجمع بين تقنيات معالجة الصور، ونماذج التعرف المتقدمة، والبني البرمجية الآمنة، يساهم في توفير حلول عملية قابلة للتوسيع، مع الحفاظ على خصوصية البيانات وحمايتها، مما يجعل هذه الأنظمة مناسبة للتطبيق الفعلي في مؤسسات التعليم العالي.

الفصل الأول

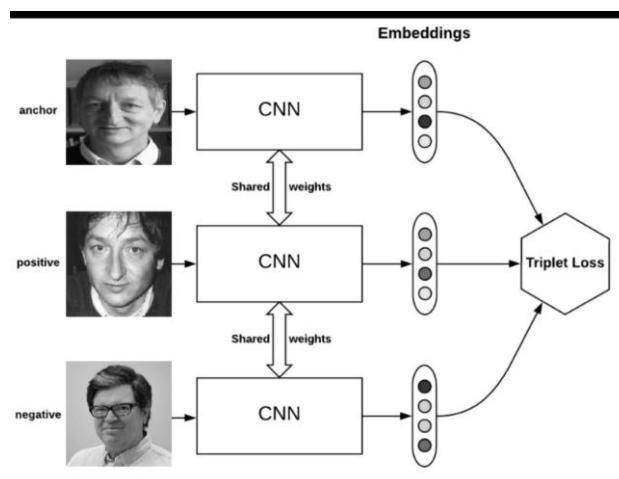
2.1 الأساس النظري

يهدف هذا الفصل إلى تقديم الإطار النظري والمفاهيمي الذي يقوم عليه تصميم وتنفيذ نظام تفقد الحضور الإلكتروني المقترن، وذلك من خلال استعراض مجموعة من المفاهيم الأساسية في مجالات التعرف على الوجه، وهندسة البرمجيات، وأمن الشبكات. ويُعد هذا الفصل أساساً علمياً لفهم البنية البرمجية والبنية الشبكية للنظام قبل الانتقال إلى الجوانب التطبيقية والتنفيذية في الفصول اللاحقة. كما يركّز الفصل على توضيح العلاقة التكاملية بين التقنيات البرمجية والمفاهيم الشبكية والأمنية، ودورها في تحقيق نظام حضور إلكتروني يتمتع بالدقة، والاعتمادية، وحماية البيانات الحساسة.

2.2 التعرف على الوجه (Face Recognition)

يُعد التعرف على الوجه أحد تقنيات القياسات الحيوية التي تعتمد على الخصائص الفизيائية الفريدة لوجه الإنسان للتحقق من الهوية. وتقوم هذه التقنية على تحليل ملامح الوجه واستخراج مجموعة من الخصائص الرقمية التي تمثل بصمة فريدة لكل شخص، ثم مقارنتها مع بيانات مخزنة مسبقاً في قاعدة بيانات.

وقد أسمى التطور الكبير في نماذج التعلم العميق في تحسين دقة أنظمة التعرف على الوجه بشكل ملحوظ، خاصة في التعامل مع التغيرات في الإضاءة، وتعبيرات الوجه، وزوايا التصوير. ويُعد هذا النوع من التقنيات مناسباً لأنظمة تفقد الحضور في الجامعات، نظراً لكونه غير تلامسي ولا يتطلب تفاعلاً مباشراً من المستخدم.

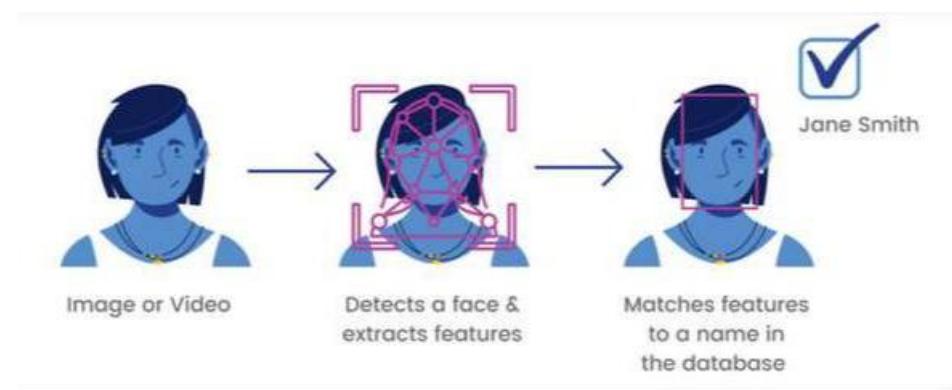


الشكل (1.1): المخطط العام لأآلية عمل نظام التعرف على الوجه

2.3 كشف الوجه والتعرف عليه (Face Detection vs Face Recognition)

تمر أنظمة التعرف على الوجه بمرحلتين أساسيتين تختلفان من حيث الوظيفة والدور. تتمثل المرحلة الأولى في كشف الوجه، وتهدف إلى تحديد موقع الوجه داخل الصورة أو الفيديو دون التعرف على هوية الشخص. أما المرحلة الثانية فهي التعرف على الوجه، والتي تركز على تحديد هوية الشخص من خلال مقارنة الخصائص المستخرجة مع قاعدة البيانات.

ويُعد الفصل بين هاتين المرحلتين أمراً ضرورياً في تصميم الأنظمة الذكية، حيث يسمح باستخدام خوارزميات متخصصة لكل مرحلة، مما يؤدي إلى تحسين الكفاءة العامة للنظام وزيادة دقتها.



الشكل (1.2): الفرق بين كشف الوجه والتعرف على الوجه

2.4 الأنظمة المعتمدة على واجهات البرمجة (API-Based Systems)

تشير الأنظمة المعتمدة على واجهات البرمجة (APIs) إلى نمط معماري يسمح بتبادل البيانات والخدمات بين مكونات النظام المختلفة من خلال واجهات محددة وواضحة. وبُسّهم هذا الأسلوب في فصل واجهة المستخدم عن منطق الأعمال، مما يعزز قابلية الصيانة والتطوير.

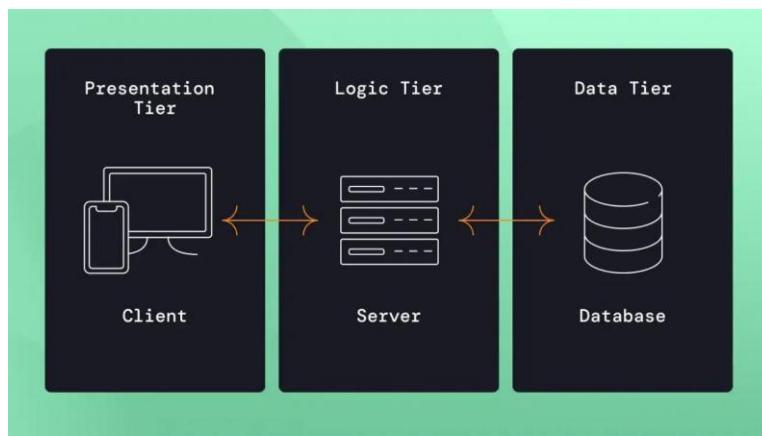
في نظام تفقد الحضور الإلكتروني، تُستخدم واجهات البرمجة لربط واجهة المستخدم بخدمات التعرف على الوجه وقواعد البيانات وأنظمة المصادقة، مما يحقق بنية مرنة وقابلة للتتوسيع.

2.5 البنية متعددة الطبقات (Three-Tier Architecture)

تعتمد البنية متعددة الطبقات على تقسيم النظام إلى ثلاثة طبقات رئيسية:

طبقة العرض، وطبقة التطبيق، وطبقة البيانات. ويساعد هذا التقسيم على تنظيم النظام بشكل أفضل، وتقليل الاعتماد المباشر بين مكوناته، وتعزيز مستوى الأمان.

وتحت هذه البنية مناسبة لأنظمة الحضور الجامعية التي تتطلب إدارة عدد كبير من المستخدمين، مع ضمان حماية قواعد البيانات من الوصول المباشر.

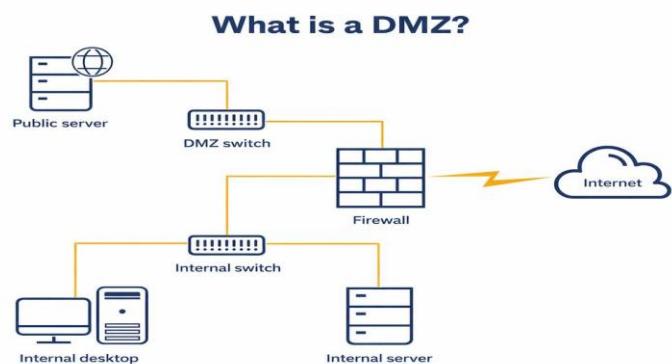


(الشكل(1.3): البنية متعددة الطبقات (Three-Tier Architecture)

2.6 منطقة العزل الشبكي(Demilitarized Zone – DMZ)

تشير منطقة العزل الشبكي (DMZ) إلى طبقة وسيطة تُستخدم لعزل الخوادم التي تعامل مع المستخدمين الخارجيين عن الشبكة الداخلية. وتحتَّمَّ DMZ من المفاهيم الأساسية في تصميم الشبكات الآمنة، حيث تقلل من مخاطر الوصول غير المصرح به إلى الموارد الحساسة.

في نظام الحضور الإلكتروني، تُستخدم DMZ كنقطة دخول وحيدة لواجهات البرمجة والخدمات، مما يعزز مستوى الحماية ويحد من انتشار الهجمات داخل الشبكة.



الشكل (1.4): نموذج بنية شبكة باستخدام DMZ

الفصل الثاني

يوثق هذا الفصل المعمارية البرمجية التي تم تطبيقها في مشروع "نظام الحضور الجامعي المعتمد على التعرف على الوجه". يرتكز الفصل على كيفية تقسيم النظام إلى وحدات مستقلة (Modules)، وكيفية تفاعل الواجهات الأمامية مع واجهات REST API في Django، وكيف يتم تشغيل محرك التعرف على الوجه (ArcFace) عبر ONNX (API Embedding Face) لإنتاج تمثيل عدي للوجه (Embedding Face) وربطه بـ هوية الطالب. كما يوضح الفصل الآليات لتحقيق من الصلاحيات داخل القواعد (Accounting) والآلية لحفظ سجلات التدقيق (Audit Logs) لدعم مبدأ المحاسبة (Authorization) وإمكانية تتبع الأحداث.

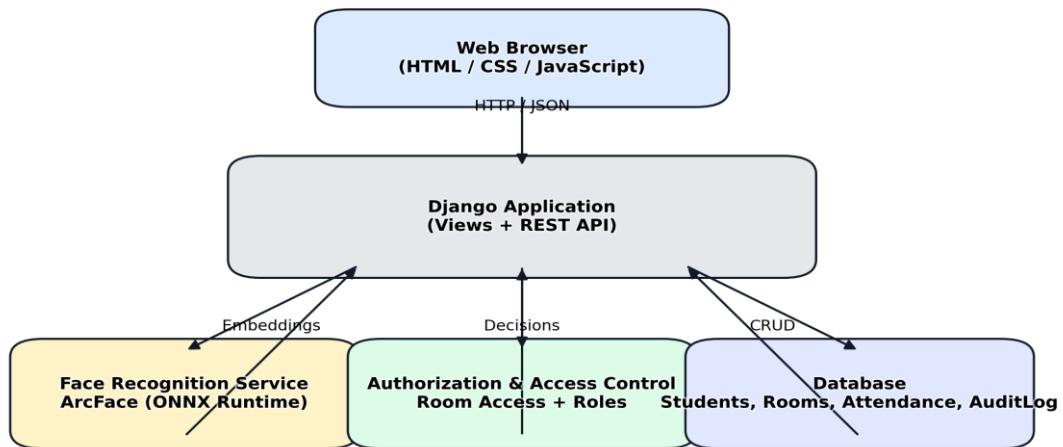
3.1 نظرة عامة على الفصل

يوفر النظام مسار عمل متكامل لإدارة الحضور داخل الجامعة. يبدأ المسار من تسجيل طالب جديد (Enrollment) عبر التقاط صورة/إطار من الكاميرا، ثم استخراج الـ Embedding وتخزينه في قاعدة البيانات. بعد ذلك، عند تنفيذ عملية التحقق داخل قاعة محددة (Live Scan)، يقوم النظام باستخراج الـ Embedding من الصورة الحالية ومقارنته مع القوالب المسجلة للطلاب، ثم يطبق سياسة الصلاحيات الخاصة بالقاعة (هل هذا الطالب مسموح له بالدخول لهذه القاعة/الجامعة؟). في النهاية يتم إنشاء سجل حضور منظم (Attendance Record) وإضافة سجل تدقيق (Audit Log) يتضمن نتيجة العملية ومعلوماتها.

من الناحية الهندسية، تم تصميم النظام كبنية معيارية (Modular Design) بحيث تكون كل وظيفة رئيسية معزولة قدر الإمكان. هذا التقسيم يسهل تطوير النظام واختباره، ويجعل توسيعه لاحقاً أكثر أماناً ووضوحاً. كما تم تضمين عناصر "المصادقة + التفويض + المحاسبة" (AAA) من خلال التحقق من هوية المستخدم عند الحاجة، تقييم صلاحية الطالب لدخول قاعة معينة (Authorization)، وتسجيل كل العمليات الأمنية المهمة مثل تسجيل الطلاب، محاولات التتحقق، وأي محاولة وصول غير مصرح بها.

3.2 معمارية النظام

يتبع النموذج الأولي معمارية طبقية تفصل بين: طبقة العرض (Web Pages)، وطبقة منطق الأعمال (Django)، وخدمات التعرف على الوجه، ووحدة التحكم بالوصول. يتم تبادل البيانات بين المتصفح وتطبيق HTTP/JSON، بينما يتم استدعاء محرك ArcFace داخلياً لاستخراج الـ Embedding. يوضح الشكل 3.1 نظرة شاملة على المكونات الأساسية ومسارات الاتصال بينها.



الشكل(2.1) : نظرة عامة على بنية النظام

تم تطوير النظام باستخدام Django REST Framework مع Python لواجهات الـ API. أثناء العرض التجريبي يتم تشغيل الخادم محلياً عبر `python manage.py runserver`، ويقوم المتصفح باستدعاء الصفحات ومسارات الـ API لجلب البيانات وتحديث لوحة التحكم (Dashboard) بشكل دوري.

3.3 التقنيات والأدوات المستخدمة

يعرض هذا القسم التقنيات الأساسية التي اعتمد عليها التنفيذ. يشمل ذلك: Django/DRF لإنشاء واجهات الويب وواجهات الـ API، قاعدة بيانات SQLite (في النسخة التجريبية)، مكتبات تشغيل النماذج عبر ONNX Runtime، ومعيار ArcFace لاستخراج تمثيل الوجه. كما تم استخدام HTML/CSS/JavaScript في الواجهات، وبالأخص `getUserMedia` للوصول إلى الكاميرا ضمن المتصفح.

الفصل الثاني

الجدول 2.1 : مجموعه التقنيات (تنفيذ النموذج الأولي)

Layer	Technology	Purpose
Backend Web Framework	Django	Routing, views, templates, ORM, admin panel
API Layer	Django REST Framework	JSON endpoints, authentication mechanisms, request validation
ML Inference Runtime	ONNX Runtime	Execute ArcFace embedding model efficiently on CPU/GPU
Face Embedding Model	ArcFace	Generate discriminative face embeddings for matching
Database	Relational DB (SQLite/PostgreSQL)	Persistence for students, rooms, attendance, audit logs
Frontend	HTML/CSS/JavaScript	Enrollment page, live scan view, dashboard updates
Administration	Django Admin	Create rooms, manage enrollments, inspect logs, debug data

3.4 مكونات النظام (تفكيك الوحدات)

من أجل قابلية الصيانة والتنظيم، قسم النظام إلى وحدات واضحة، بحيث تتكامل هذه الوحدات عبر واجهات محددة. الهدف من هذا التقسيم هو تقليل الترابط (Coupling) وزيادة التماสق (Cohesion) بحيث يمكن تطوير كل جزء دون كسر بقية الأجزاء. يوضح الجدول 3.2 نظرة سريعة على هذه الوحدات ومسؤولياتها.

- المصادقة وملفات المستخدم (User Profiles & Authentication): إدارة حسابات المستخدمين (مثل Admin/Teacher)، تسجيل الدخول، وإدارة بيانات الملف الشخصي الأساسية، مع إمكانية الاعتماد على جلسات الويب أو Tokens عند استخدام واجهات API.
- إدارة الطلاب (Student Management): حفظ هوية الطالب (الرقم الجامعي والاسم) وربطها ب قالب الوجه المخزن (Embedding Vector) الذي يمثل بصمة الوجه الرقمية.
- إدارة القاعات والصلاحيات (Access Management & Room): تعريف القاعات (Rooms) وسياسات الوصول المرتبطة بها، أي تحديد الطالب المسموح لهم بالدخول إلى قاعة معينة (ويمكن توسيعها لتشمل جلسات مقرر محددة).
- محرك الحضور (Attendance Engine): تنسيق خطوات التحقق: التقاط الإطار من الكاميرا، استخراج الـ Embedding، المقارنة مع القوالب المسجلة، دمج نتيجة التعرف مع سياق القاعة، ثم كتابة سجل حضور في قاعدة البيانات.
- التدقيق والمحاسبة (Accounting & Auditing): تسجيل الأحداث ذات الحساسية الأمنية مثل محاولات تسجيل طالب، محاولات تحقق ناجحة/فشلية، ومحاولات وصول غير مصرح بها. هذه السجلات تدعم تتبع "من قام بماذا، ومتى، ومن أي عنوان IP، وما هي النتيجة".
- الواجهة الأمامية ولوحة التحكم (Frontend Dashboard): توفير صفحات سهلة الاستخدام للعرض التجريبي: صفحة تسجيل طالب، صفحة التتحقق المباشر (Live Scan)، ولوحة Dashboard لعرض سجلات الحضور وتحديثها دورياً.

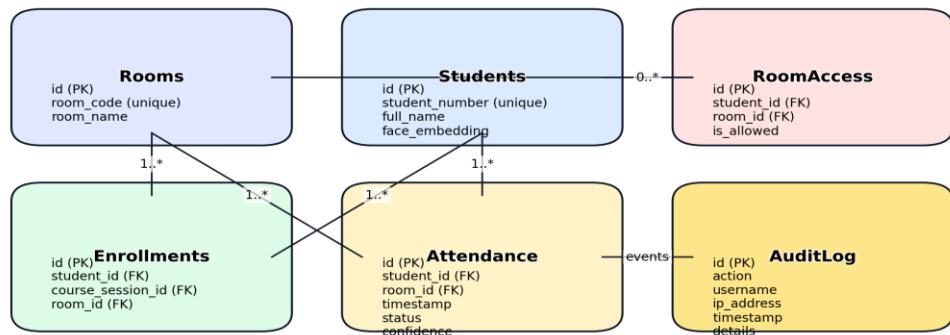
الفصل الثاني

الجدول (2.2) : المكونات الرئيسية ومسؤولياتها في Django

Component	Typical File(s)	Key Responsibilities
Models	models.py	Define Students, Rooms, Enrollments, RoomAccess, Attendance, AuditLog
Views (Pages)	views.py + templates	Render dashboard, enrollment page, live scan page
Views (APIs)	views.py / api views	Enrollment API, verification API, attendance list API
Serializers/Validators	serializers.py	Validate inputs (student data, images, parameters)
Face Service	face_service/*	Load ONNX model, extract embeddings, cosine similarity
Authorization	authorization.py	Decide if student is allowed in target room/session
Auditing	accounting.py	Write structured audit events for accountability

3.5 تصميم قاعدة البيانات

يعتمد النظام على مخطط علائقى (Relational Schema) يربط بين: الطلاب، القاعات، جلسات المقررات (Course Sessions)، وسجلات الحضور والتدقيق. يهدف هذا التصميم إلى ضمان التكامل المرجعى (Integrity) وإتاحة الاستعلامات المهمة، مثل: عرض حضور طالب ضمن فترة زمنية، أو عرض جميع عمليات الدخول المحاولة لقاعة معينة. يوضح الشكل 2.2 العلاقات الأساسية بين الجداول.



الشكل (2.2) : تصميم قاعدة البيانات وعلاقات الجداول

يحتفظ جدول **Students** بمعلومات الطالب الأساسية وفأله الوجه (**face_embedding**). بينما يعرف جدول **Rooms** القاعات وأковادها (مثل LAB-101). ويستخدم جدول **Enrollments/RoomAccess** لربط الطالب بالقاعات أو الجلسات المسموح بها، أما جدول **Attendance** فيحفظ نتيجة كل عملية تحقق (الحالة، الوقت، القاعة، ونسبة الثقة). أخيراً، يسجل جدول **AuditLog** أحداث النظام لأغراض التتبع والمحاسبة.

الجدول(2.3) : الجداول الأساسية والحقول الرئيسية

Table	Key Fields	Notes
Students	student_number, full_name, face_embedding	Face embedding stored as a vector/blob/encoded list
Rooms	room_code, room_name	Used to tag events by physical location
Enrollments	student_id, room_id, course_session_id	Maps student to room/session context
RoomAccess	student_id, room_id, is_allowed	Explicit allow/deny rules (can override defaults)
Attendance	student_id, room_id, timestamp, status, confidence	Status {IN,OUT,FORBIDDEN}
AuditLog	action, username, ip_address, timestamp, details	Immutable security trail

3.6 خدمة التعرف على الوجه

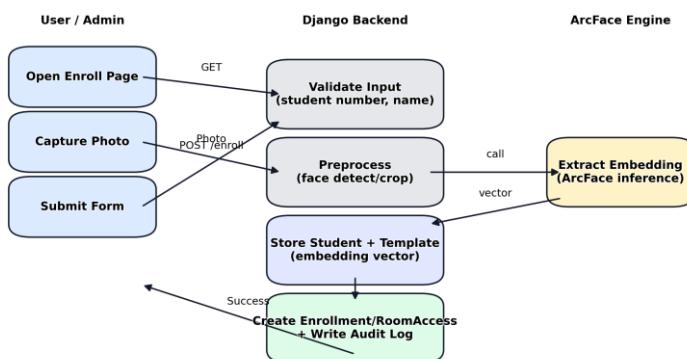
تعتمد عملية التعرف على الوجه على استخراج تمثيل عددي ثابت الطول (Embedding) لكل وجه. في مرحلة التسجيل، تُلقط صورة واضحة لوجه الطالب ثم تمر عبر النموذج ArcFace (بصيغة ONNX) لإنتاج Embedding. يتم تخزين هذا المتجه في قاعدة البيانات وربطه بمعرفة الطالب. هذا الأسلوب يسمح بالمقارنة لاحقاً عبر مقارنة المتجهات بدل الاعتماد على الصورة الخام.

في مرحلة التحقق، يتم استخراج Embedding من الإطار الحالي ثم حساب مقياس التشابه (Similarity) مع قواليب الطلاب المسجلة. إذا تجاوزت أعلى درجة تشابه عتبة (Threshold) محددة، يعتبر الطالب مُعْرِفًا (Recognized). بعد التعرف، تُطبق سياسة الوصول: إذا كان الطالب ضمن قائمة المسموح لهم لقاعة/جامعة معينة، تُسجل الحالة IN، وإذا كان معروفاً لكنه غير مخول للقاعة تُسجل الحالة FORBIDDEN. أما إذا لم يتم الوصول لعتبة التعرف فتعامل الحالة كـ Unknown أو OUT وفق سياسة النظام.

من أجل موثوقية النتائج، يفرض النظام قيوداً أثناء التسجيل مثل: وجود وجه واحد فقط في الإطار، إضافة مناسبة، وعدم وجود اهتزاز كبير. كما تتم معالجة الحالات الاستثنائية مثل فشل التقاط الصورة أو عدم وجود وجه واضح، مع تسجيل الحدث في AuditLog لإظهار سبب الفشل.

3.7 عملية تسجيل الطالب (Enrollment)

تُنفذ عملية التسجيل عبر صفحة Enrollment في الواجهة. عند إدخال رقم الطالب والاسم، يطلب المتصفح صلاحية الوصول للكاميرا ثم يلتقط صورة عند الضغط على Take Photo. ترسل الصورة إلى Django API، الذي يستدعي محرك ArcFace لاستخراج الـ Embedding Students ويخزنه في جدول Enrollment. يوضح الشكل 2.3 تسلسل الخطوات من المتصفح وحتى تخزين القالب.



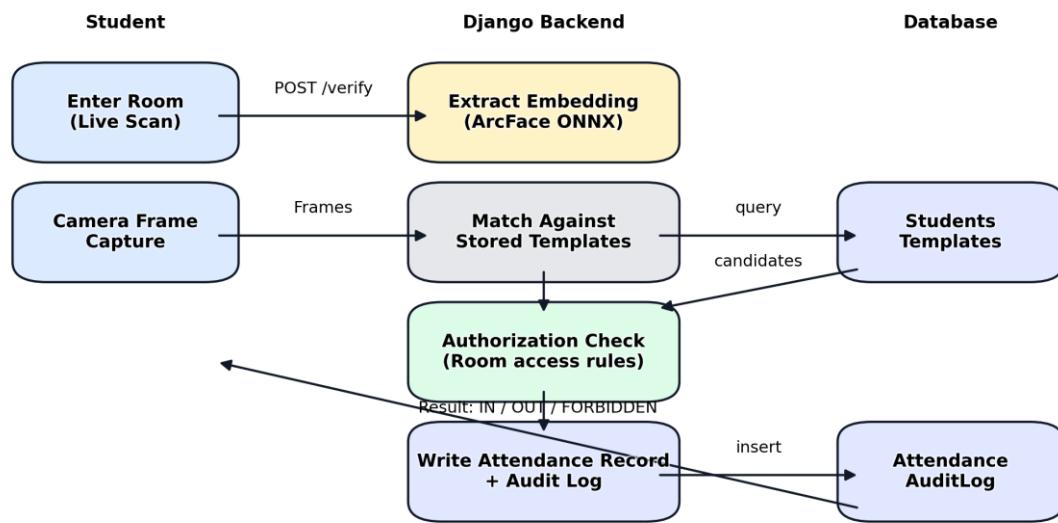
الشكل(2.3) : تسلسل عملية التسجيل

لربط الطالب بقاعة محددة (التجربة التفريضي) ، يتم إنشاء قيد في Enrollment أو RoomAccess يحدد أن هذا الطالب مسموح له بالدخول إلى قاعة معينة مثل (LAB-101) Lab 1. يمكن تنفيذ ذلك عبر لوحة Django Admin . يمكن تنفيذ ذلك عبر لوحة Django Admin . يمكن تنفيذ ذلك عبر لوحة Django Admin .

3.8 التحقق من الحضور (Verification)

عملية التتحقق (Verification) تتم من خلال صفحة Live Scan المرتبطة بقيمة القاعة الحالية (Room). يتم التقاط إطار (Frame) من الكاميرا بشكل دوري وإرساله إلى مسار API مخصص للتحقق. يستخرج (Code) . يتم التقاط إطار (Frame) من الكاميرا بشكل دوري وإرساله إلى مسار API مخصص للتحقق. يستخرج (Code) . يتم التقاط إطار (Frame) من الكاميرا بشكل دوري وإرساله إلى مسار API مخصص للتحقق. يستخرج (Code) .

الخادم الـ Embedding للإطار ثم يبحث عن أفضل تطابق مع الطلاب المسجلين. بعد ذلك يطبق سياسة الوصول بناءً على القاعة: (مسموح / غير مسموح) ثم يعيد نتيجة بصيغة JSON ليتم عرضها في الواجهة. يوضح الشكل 2.4 هذا التسلسل.

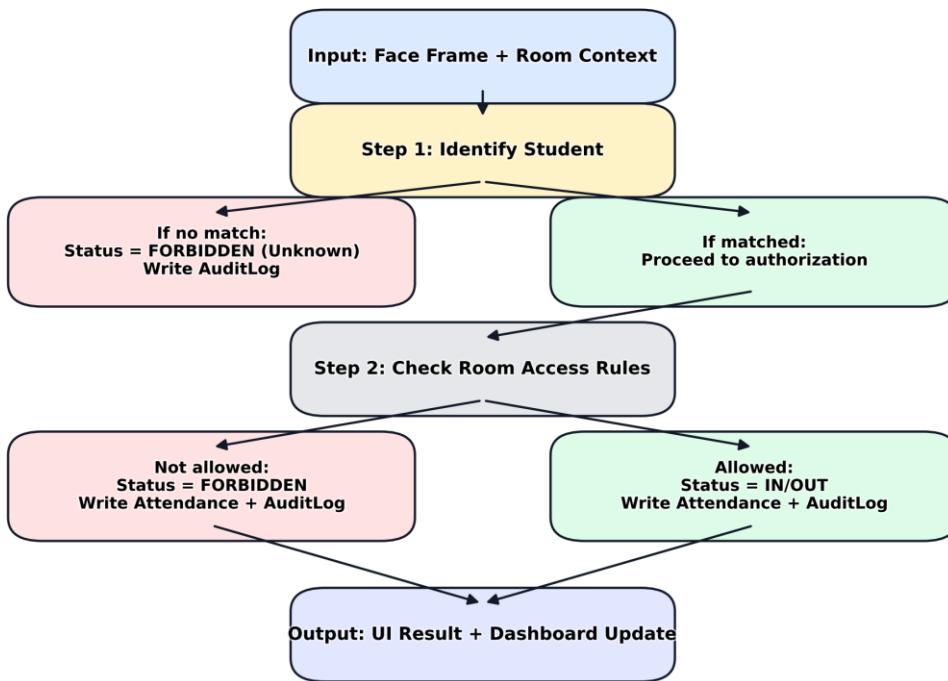


الشكل (2.4) : تسلسل التحقق من الحضور

بعد اتخاذ القرار، ينشئ النظام سجل حضور جديد يحتوي على: الطالب (إن وجد)، القاعة، الوقت، الحالة (IN/OUT/FORBIDDEN) وقيمة الثقة (Confidence). يتم تحديث لوحة Dashboard بشكل دوري عبر استدعاء مسار API لعرض السجل اليومي وتلخيص النتائج.

3.9 التفويض والتحكم بالوصول (Authorization)

يطبق النظام مبدأ التحكم بالوصول عبر قواعد واضحة تربط هوية الطالب بسياق القاعة/الجلسة. بمعنى آخر، التعرف على الوجه وحده لا يكفي لإثبات السماح بالدخول؛ بل يجب أن يكون الطالب "مخولاً" لهذه القاعة وفق البيانات المخزنة. لهذا السبب تم فصل منطق التعرف (Recognition) عن منطق الصلاحيات (Authorization) لضمان اتساق القرار وسهولة تدقيقه.



الشكل (2.5) : مخطط تدفق قرار التفويض

عند التحقق، إذا تم التعرف على الطالب ثم تم إيجاد قيد RoomAccess يسمح له بالدخول، تكون النتيجة IN. أما إذا تم التعرف على الطالب لكن لا يوجد سماح لهذه القاعة، تكون النتيجة FORBIDDEN وتحسّن كمحاولة غير مصرح بها. هذا الفصل بين "التعرف" و"التفويض" يحمي النظام من سيناريوهات مثل تعرف صحيح ولكن في قاعة خاطئة.

3.10 التدقيق والمحاسبة (Accounting & Auditing)

يمثل التدقيق (Auditing) جزءاً أساسياً في النظام، لأنّه يوفر سجلاً زمنياً لجميع الأحداث المهمة. ويستخدم هذا السجل أثناء العرض لإثبات ما تم تنفيذه عملياً: إنشاء طالب جديد، محاولات تحقق ناجحة، ومحاولات مرفوضة.

يتضمن سجل التدقيق عادة حقولاً مثل: نوع العملية (action)، اسم المستخدم أو الطالب، عنوان IP، الطابع الزمني (timestamp)، وتفاصيل إضافية (details) مثل رقم القاعة أو سبب الفشل. يساعد ذلك في التحليل الجنائي (Forensics) واستكشاف الأخطاء وتحسين الضبط.

3.11 واجهات النظام ومسارات الصفحات وواجهات API

يعتمد النظام على تنظيم واضح لمسارات الصفحات (Pages) ومسارات الـ API داخل auth_app/urls.py، مع استخدام أسماء مسارات (...=name) لتمكين الدوال {url %} داخل القوالب. خلال التطوير ظهرت مشكلة NoReverseMatch بسبب الإشارة إلى اسم مسار غير موجود (مثل 'face')، وتم حلها بتوحيد أسماء المسارات في urls.py وتحديث القوالب لاستخدام الأسماء الصحيحة (مثلاً: 'attendance', 'enroll', 'verify').

الجدول (2.4) : أمثلة على نقاط النهاية المستخدمة في النموذج الأولي

Category	Endpoint (example)	Method	Purpose
Dashboard	/	GET	Main dashboard view
Enrollment Page	/enroll-page/	GET	UI for registering a new student
Live Scan Page	/attendance/	GET	UI for camera-based verification
Enroll API	/auth/enroll-face/	POST	Submit photo + student metadata to create template
Verify API	/auth/verify-face/	POST	Submit photo/frame to identify student and record status
Attendance List API	/auth/attendance/	GET	Return recent attendance records as JSON (filterable)
Admin	/admin/	GET	Administrative management interface

3.12 اعتبارات الأمان والخصوصية

على الرغم من أن النموذج الحالي تجريبي، إلا أن تصميمه أخذ بعين الاعتبار مجموعة من ضوابط الأمان الأساسية. يشمل ذلك حماية مسارات الإدارية، فرض صلاحيات الوصول للعمليات الحساسة، وتسجيل كل الأنشطة المهمة. يستعرض هذا القسم أهم اعتبارات الأمانة والخصوصية التي تم الالتزام بها أثناء التنفيذ.

الفصل الثاني

- **المصادقة (Authentication):** تقييد لوحة الإدارة Django Admin للمستخدمين الموثوقين فقط، وإمكانية توسيع النظام لاحقاً لاستخدام Token/JWT لمسارات الـ API عند الحاجة.
- **التفويض (Authorization):** تطبيق قواعد وصول دقيقة لكل قاعة عبر RoomAccess/Enrollments، بحيث لا تقبل عملية دخول إلا إذا كانت مطابقة لهوية الطالب ومطابقة لسياسة القاعة.
- **التدقيق (Auditing):** تخزين AuditLog لكل حدث حساس مع بيانات السياق (IP/وقت/نتيجة)، مما يسهل المراجعة واكتشاف السلوك غير الطبيعي.
- **سلامة البيانات (Data Integrity):** الاعتماد على علاقات Foreign Keys وضمان التكامل المرجعي بين الطالب والقاعات وسجلات الحضور.
- **مبدأ أقل الصلاحيات (Least Privilege):** فصل الأدوار بين المستخدمين (Admins/Teachers) والطلاب، وحصر التعديلات المباشرة في البيانات الحساسة ضمن لوحة الإدارة فقط.

الجدول(2.5) : ضوابط الأمان المطبقة أو التي تم النظر فيها

Control	Where Implemented	Notes
Authentication	Django/DRF	Admin/operator must log in; tokens may protect APIs
Room-based authorization	authorization module	Denies access if student not permitted in room
Audit logging	accounting module	Logs enrollment, verification, configuration changes
Thresholding	face service	Rejects low-confidence matches to reduce false accepts
Separation of concerns	architecture	Keeps ML logic separate from web and database logic

3.13 سيناريو العرض العملي (Demo)

لتنفيذ عرض عملي واضح، يمكن اتباع السيناريو التالي: (1) إنشاء قاعة داخل النظام (Lab) إن لم تكن موجودة. (2) تسجيل طالب جديد عبر صفحة Enrollment والتقاط صورة واضحة. (3) ربط الطالب بالقاعة عبر إضافة (4) فتح صفحة Live Scan وتحديد القاعة 1 Lab ثم تنفيذ التحقيق. (5) ملاحظة ظهور السجل في Dashboard مع الحالة IN والثقة. (6) لإثبات التفويض، يمكن إزالة RoomAccess ثم إعادة التحقق لنحصل على FORBIDDEN، مع وجود سجل تدقيق يثبت محاولة دخول غير مصرح بها.

توفر لوحة Admin Django وسيلة فعالة لإظهار البيانات المخزنة أثناء العرض. يمكن من خلالها استعراض الطالب المُنشأ (Students)، التحقق من وجود الـ face_embedding محفوظاً، مراجعة قيود RoomAccess أو Audit logs وAttendance Enrollments التي تحدد السماح، وأخيراً مراجعة جداول Audit logs.

3.14 القيود والتحسينات المستقبلية

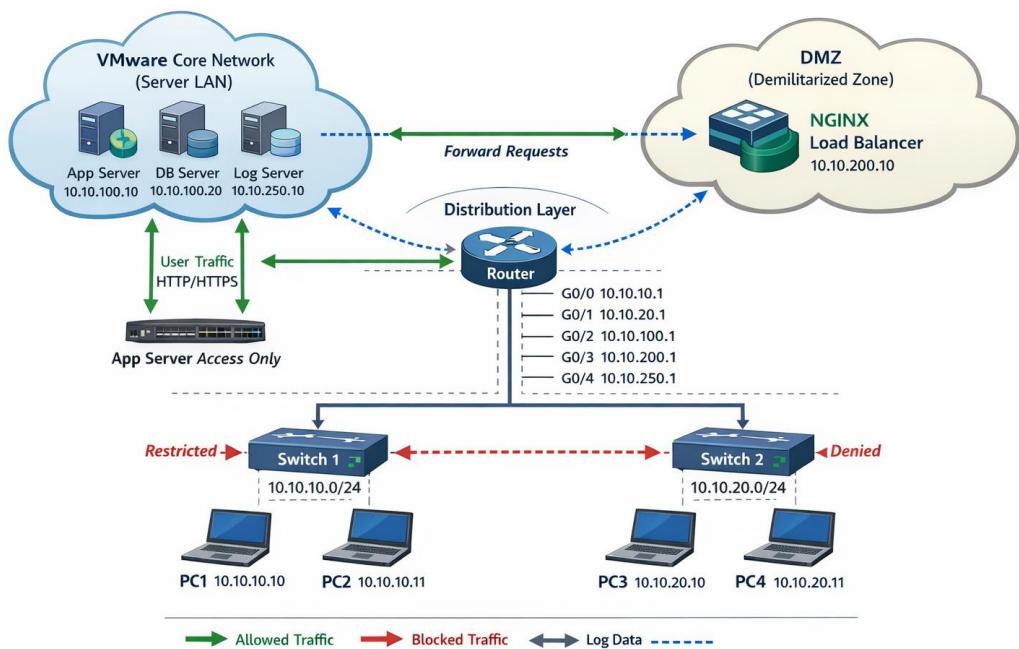
يمثل هذا النموذج نسخة أولية (Prototype) ترکز على إثبات الفكرة. من التحسينات المقترحة مستقبلاً: إضافة كشف الحيوية (Liveness Detection) لمنع محاولات الخداع بالصور، استخدام قاعدة بيانات أقوى مثل PostgreSQL، تحسين إدارة الجلسات والتكامل مع جدول المحاضرات الرسمي، تشفير قوالب الوجه أو تخزينها بشكل آمن (مثلاً تشفير على مستوى الحقل)، وإضافة أدوات مراقبة (Monitoring) وتقارير إحصائية متقدمة.

3.15 الخلاصة

قدم هذا الفصل شرحاً تفصيلياً للبنية البرمجية لنظام الحضور، بدءاً من المعمارية العامة، مروراً بتقسيم المكونات وتصميم قاعدة البيانات، وصولاً إلى آليات تسجيل الطالب والتحقق من الحضور والتحكم بالوصول وسجلات التدقيق. تشير النتائج العملية من لوحة التحكم وسجلات القاعدة إلى نجاح دمج التعرف على الوجه مع سياسات التفويض، مما يوفر نظام حضور أكثر انضباطاً وقابلية للتتبع مقارنة بالحلول التقليدية.

الفصل الثالث

في هذا الفصل سيتم عرض بنية الشبكة المحلية ، و المكونات الشبكية وأدوارها ، التقسيم الشبكي المعتمد ، منهجية التنفيذ والربط والعنونة .



الشكل (3.1) : الشبكة

3.1 تصميم الشبكة

تم تصميم شبكة المشروع وفق مفهوم التقسيم الشبكي الظبيقي Layered Network Segmentation بهدف تحقيق عزل منطقي بين أنواع الحركة المختلفة داخل النظام. يعكس هذا التصميم أسلوباً شائعاً في بناء أنظمة الويب ضمن بيئات المؤسسات، حيث يتم فصل شبكة المستخدمين عن منطقة الخدمات المعروضة (DMZ)، وفصل خوادم التطبيق عن طبقة العرض، وإضافة شبكة مستقلة للإدارة والمراقبة. يحقق هذا النهج هدفين رئисيين:

- (1) رفع مستوى الأمان عبر تقليل مساحة التعرض للهجمات والحد من الحركة الجانبية،
- (2) تبسيط التحكم بالوصول عبر سياسات واضحة قابلة للقياس والاختبار. [41][40][22][16]

3.2 المكونات الشبكية وأدوارها

اعتمد التنفيذ على العناصر التالية:

- موجّه رئيسي (R1): يقوم بوظيفة التوجيه بين الشبكات المنفصلة (Inter-Subnet Routing) ويشكّل نقطة تطبيق سياسات التحكم بالوصول (ACL) بين المناطق.
- مبدلات افتراضية (Switches) داخل GNS3: تُستخدم لتمثيل شبكات الطبقات المختلفة وربط الأجهزة الافتراضية ضمن كل منطقة.
- GNS3 داخل Cloud Nodes: لربط شبكات GNS3 مع شبكات VMware الافتراضية (VMnet) بحيث تصبح الأجهزة الافتراضية (VMs) جزءاً فعلياً من طوبولوجيا الشبكة.
- أجهزة افتراضية (VMs): تمثل خوادم النظام (خادم التطبيق، خادم الواجهة الأمامية، خادم السجلات)، مع فصل كل دور ضمن منطقة شبكة مناسبة.
- 1. أجهزة اختبار (VPCS): استخدمت كأطراف تمثل مستخدمين لإجراء اختبارات الاتصال والتحقق من التوجيه والسياسات.[44]

3.3 التقسيم الشبكي المعتمد (Segmentation Model)

تم تقسيم الشبكة إلى مناطق وظيفية مستقلة، بحيث تدار حركة المرور بينها عبر الموجّه والسياسات:

1. شبكة المستخدمين: تمثل أجهزة المستخدمين .
2. منطقة DMZ: تُخصص لطبقة العرض NGINX لأنها تتعامل مع اتصالات واردة من المستخدمين.
3. شبكة الخوادم الداخلية (Server LAN): تُخصص لخادم التطبيق (Django) وتعامل كمنطقة أكثر حساسية لأنها تحتوي منطق النظام والبيانات.
4. شبكة الإدارة/السجلات (Logs/MGMT): تُخصص للإدارة والتجميع المركزي للسجلات، ما يسهل المتابعة والتدقيق ويوفر قناة إدارة منفصلة عن المستخدمين.

يمكن هذا التقسيم من تطبيق سياسات دقيقة (على مستوى "من يسمح لمن؟") ويجعل التحقق من سلوك الشبكة قابلاً للفياس عبر اختبارات محددة.[22][40][41]

3.4 منهجية التنفيذ والربط داخل VMware و GNS3

تم تنفيذ الشبكة باستخدام **GNS3** لمحاكاة عناصر الشبكة (موجة/مبدلات)، وربطها مع **VMware** لتشغيل خوادم حقيقة ضمن آلات افتراضية. جرى استخدام **Cloud Nodes** لتمثل نقاط الربط بين عالم المحاكاة (**GNS3**) وعالم الخوادم (**VMware**) عبر واجهات **VMnet**. يتيح هذا الأسلوب اختباراً عملياً قريباً من الواقع، لأن خوادم المشروع تعمل كنظم تشغيل حقيقة (**Ubuntu Server**) وتتصل بالشبكات الافتراضية كما لو كانت ضمن بنية مادية.

ولضمان نجاح الربط، تم التحقق من:

1. أن كل خادم مرتبط بالشبكة الصحيحة ضمن **VMware**.

2. أن **Cloud Nodes** في **GNS3** مضافة إلى واجهات **VMnet** المناسبة.

3. أن المبدلات تربط الأجهزة ضمن نفس المنطقة.

4. أن الموجه يوفر التوجيه بين المناطق عند الحاجة وفق السياسات. [44][22]

3.5 العنونة Addressing

تم إعداد خطة عنونة دقيقة لتغطية جميع مكونات النظام وضمان وضوح مسارات الاتصال بين الطبقات المختلفة. صُممَت الخطة بحيث تعكس الفصل الوظيفي بين المناطق، مع تجنب أي تعارض أو تداخل في نطاقات الشبكات. [40][42]

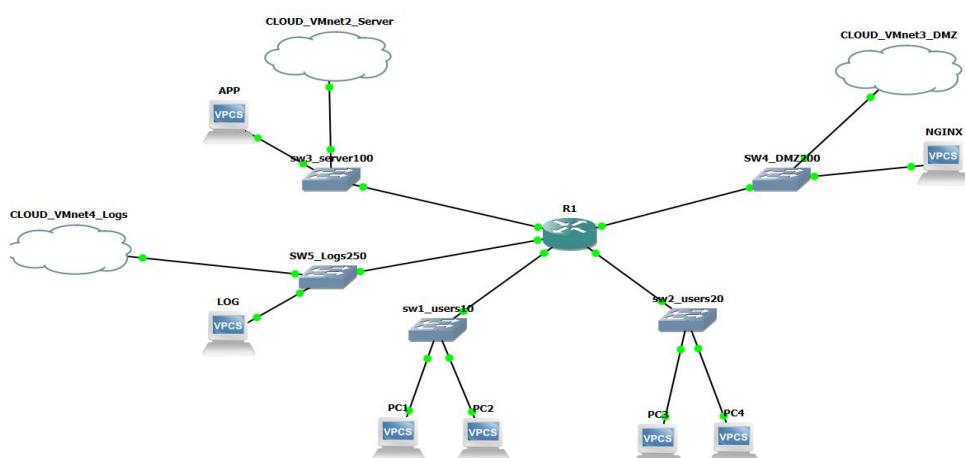
3.5.1 منهجية تصميم خطة العنونة

اعتمدت الخطة على مبدأ "شبكة فرعية لكل طبقة" (**Subnet per Tier**) لتبسيط الإدارة والضبط. تم اختيار عنونة خاصة (**Private IPs**) من الفئة A و C بحيث تكون متوافقة مع بيئات المحاكاة، وفي الوقت نفسه تعبر عن الطوبولوجيا الشبكية المقترنة.

كما تم تخصيص عنوان 1. في كل شبكة ليكون عنوان واجهة الراوتر (**Gateway**)، والعنوان 10. عادة لخادم المنطقة (**Server/VM**)، بحيث تبقى العنونة سهلة التتبع في اختبارات ping والتحقق. [42]

3.5.2 الشبكات المعتمدة

المنطقة الشبكية	الهدف	الشبكة الفرعية	البوابة	عنوان الخادم	ملاحظات
Server LAN	تشغيل خادم Django	10.10.100.0/24	10.10.100.1	10.10.100.10	شبكة داخلية محمية
DMZ (Demilitarized Zone)	استضافة خادم الواجهة الأمامية (NGINX)	10.10.200.0/24	10.10.200.1	10.10.200.10	تستقبل طلبات المستخدمين
Logs/MGMT	إدارة النظام وتجميع السجلات المركزية	10.10.250.0/24	10.10.250.1	10.10.250.10	مخصصة حسراً للإدارة والتدقير
User Network (A)	مستخدمين داخلين (طلاب/موظفي)	10.10.10.0/24	10.10.10.1	حسب الأجهزة	وصولهم فقط إلى DMZ
User Network (B)	مستخدمين خارجيين أو زوار	10.10.20.0/24	10.10.20.1	حسب الأجهزة	معزولة عن باقي الطبقات



الشكل 3.2 : الشبكة باستخدام GNS3

3.5.3 المنطق في توزيع العناوين

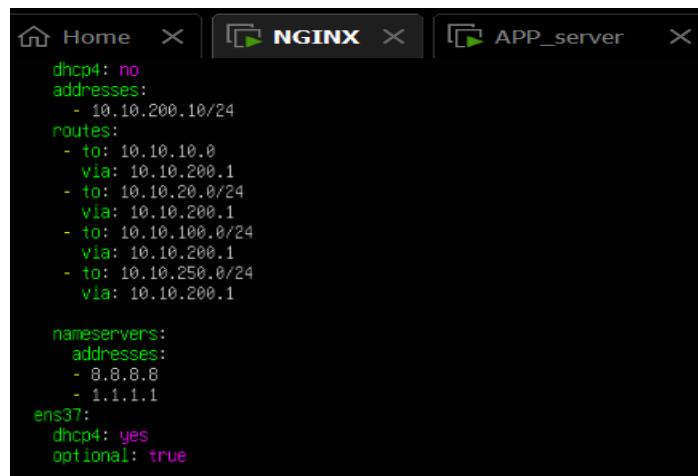
اختيار هذا النمط من العناوين $10.10.0.0/16$ لم يكن عشوائياً، بل تم تقسيم الشبكات ضمن نطاق موحد لضمان سهولة التوجيه والتمييز البصري بين الطبقات.[42]

3.5.4 ضبط العنونة على الأجهزة الافتراضية

تم تطبيق العناوين على كل آلة افتراضية داخل ملفات إعداد الشبكة (`/etc/netplan/...`) بحيث تكون دائمة حتى بعد إعادة التشغيل. تتضمن كل خادم واجهتين:

- الأولى (`ens33`): للشبكة الداخلية التي تخص دوره (Server LAN, DMZ, MGMT).
- الثانية (`ens37`): مخصصة لنظام NAT في VMware لتحديث الحزم أو تحميل المكتبات من الإنترن트 أثناء التطوير (ثم عزلها لاحقاً عند انتهاء مرحلة التنفيذ والتطوير).

تم أيضاً تحديد المسارات الافتراضية (`routes`) ضمن كل خادم بحيث توجه الشبكات الداخلية نحو البوابة الصحيحة (واجهة الراوتر ضمن شبكتها). أما على مستوى GNS3، فقد تم ربط كل شبكة بمبدل (Switch) خاص بها، متصل بدوره بواجهة الراوتر المخصصة، بما يضمن أن كل شبكة مستقلة تماماً ولا يحدث أي تداخل بين البوابات.[44]



```
Home      X | NGINX     X | APP_server X
dhcpc4: no
addresses:
- 10.10.200.10/24
routes:
- to: 10.10.10.0
  via: 10.10.200.1
- to: 10.10.20.0/24
  via: 10.10.200.1
- to: 10.10.100.0/24
  via: 10.10.200.1
- to: 10.10.250.0/24
  via: 10.10.200.1
nameservers:
addresses:
- 8.8.8.8
- 1.1.1.1
ens37:
dhcpc4: yes
optional: true
```

الشكل 3.3 : اعدادات nginx داخل netplan (مثال)

3.5.5 التحقق من ثبات العنونة بعد إعادة التشغيل

تم إجراء اختبارات متعددة للتأكد من أن العنونة لا تتغير بعد إيقاف الأجهزة أو إعادة تشغيلها، وذلك بتنفيذ أوامر ip a و ip 2 بعد كل إعادة تشغيل، ومقارنة النتائج مع القيم الأصلية المحددة في خطة العنونة.

جميع الاختبارات أثبتت أن العنونة أصبحت ثابتة ودائمة بفضل الاعتماد على netplan بدلاً من الأوامر المؤقتة.

3.6 التتحقق من الاتصال (Connectivity Verification) واختبارات

التشغيل الشبكي

بعد تثبيت خطة العنونة وربط مكونات الطوبولوجيا، تم إجراء سلسلة اختبارات تتحقق للتأكد من أن الشبكة تعمل كما هو مخطط لها ، وأن الاتصال بين الطبقات يتم حسراً عبر الموجه، وبما يسمح لاحقاً بوصول المستخدمين إلى خدمة النظام (تسجيل الحضور) عبر المسار الشبكي الصحيح.[22][40]

3.6.1 التتحقق من حالة واجهات الموجه (Router Interface Verification)

- للتأكد أن واجهات الراوتر تعمل فعلياً (Up/Up) وأن كل شبكة موصولة بالواجهة الصحيحة. حيث تم تنفيذ الأمر التالي على R1:

show ip interface brief

- النتيجة : ظهور واجهات الراوتر الخاصة بكل منطقة بحالة up/up وبعنوانينها المحددة ضمن خطة العنونة (بوابة لكل شبكة). وبالتالي تأكيد جاهزية الراوتر للتوجيه بين الشبكات وأن كل منطقة أصبحت "متصلة" به .(Connected Networks)

R1#show ip interface brief	Interface	IP-Address	OK?	Method	Status
	rotocol				
	FastEthernet0/0	10.10.10.1	YES	NVRAM	up
	p				
	FastEthernet0/1	10.10.20.1	YES	NVRAM	up
	p				
	FastEthernet1/0	10.10.100.1	YES	NVRAM	up
	p				
	FastEthernet2/0	10.10.200.1	YES	NVRAM	up
	p				
	FastEthernet3/0	10.10.250.1	YES	NVRAM	up
	p				
R1#	□				

الشكل 3.4 : واجهات الراوتر up

3.6.2 التحقق من الاتصال داخل كل شبكة إلى البوابة (Gateway Reachability)

- الهدف: التأكيد من أن كل جهاز ضمن منطقته يستطيع الوصول إلى البوابة الافتراضية الخاصة به، وهو الشرط الأول قبل اختبار أي اتصال بين الشبكات.
- الاختبارات: تم تنفيذ اختبار ping من كل VM نحو بوابتها:
 - من APP VM (Server LAN) نحو 10.10.100.1
 - من NGINX VM (DMZ) نحو 10.10.200.1
 - من LOG VM (Logs/MGMT) نحو 10.10.250.1
- النتيجة : نجاح ping مع نسبة فقد 0%. وهذا يثبت أن طبقة الربط (Layer 2) ضمن كل شبكة سليمة وأن الجهاز يرى الراوتر كـ Gateway .

```

Home X NGINX X APP_server X
admin_sudoer@ubuntudatabaseserver:~$ ping -c 4 10.10.200.1
PING 10.10.200.1 (10.10.200.1) 56(84) bytes of data.
64 bytes from 10.10.200.1: icmp_seq=1 ttl=128 time=0.597 ms
64 bytes from 10.10.200.1: icmp_seq=2 ttl=128 time=0.474 ms
64 bytes from 10.10.200.1: icmp_seq=3 ttl=128 time=0.547 ms
64 bytes from 10.10.200.1: icmp_seq=4 ttl=128 time=0.676 ms

--- 10.10.200.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3105ms
rtt min/avg/max/mdev = 0.474/0.573/0.676/0.073 ms
admin_sudoer@ubuntudatabaseserver:~$
```

الشكل 3.5 : تنفيذ اختبار ping من nginx نحو بوابته 10.10.200.1 (مثلاً)

3.6.3 التتحقق من جداول ARP على الراوتر (L2/L3 Validation via ARP)

- الهدف: التتحقق أن الراوتر “يرى” الأجهزة فعلياً على المستوى الظبي (MAC)، وليس فقط أن عناوينه مصبوطة.

الاختبار: بعد تنفيذ ping من الأجهزة للأجهزة الـ Gateway على R1:

`show arp`

النتيجة: ظهور entries للأجهزة (مثل APP/NGINX/LOG أو أجهزة الاختبار) ضمن الشبكات المعنية.

لتأكيد أن الاتصال داخل الشبكة واقعي [40]. (VM ↔ Switch/Cloud ↔ Router)

Protocol	Address	Age (min)	Hardware Addr	Type	Interface
Internet	10.10.10.1	-	c401.51e4.0000	ARPA	FastEthernet0/0
Internet	10.10.20.1	-	c401.51e4.0001	ARPA	FastEthernet0/1
Internet	10.10.100.1	-	c401.51e4.0010	ARPA	FastEthernet1/0
Internet	10.10.200.1	-	c401.51e4.0020	ARPA	FastEthernet2/0
Internet	10.10.250.1	-	c401.51e4.0030	ARPA	FastEthernet3/0

الفصل الثالث

3.6.4 اختبار التوجيه بين الشبكات (Inter-Subnet Routing)

- الهدف: إثبات أن الراوتر يقوم بالتوجيه بين المناطق وفق التصميم، وأن الوصول بين الطبقات ممكن (قبل تطبيق سياسات المنع/السماح النهائية).
- الاختبارات: تم تنفيذ ping بين الطبقات، مثل:
 - من APP VM إلى LOG VM
 - من NGINX VM إلى LOG VM
 - من APP VM إلى NGINX VM
- النتيجة: نجاح الاتصال بين الشبكات لأن سياسات "سامحية" أو لم يتم تشديد ACL بعد. وهذا يثبت جاهزية البنية متعددة الشبكات للتكامل بين طبقات النظام، وهو شرط أساسي لوصول الخدمة للمستخدم النهائي.

3.6.5 التحقق من مسار الخدمة (Service Path Validation)

- الهدف: ربط نتائج الشبكة بالهدف الوظيفي النهائي: وصول المستخدم للنظام، تم تعريف "مسار الخدمة" على الشكل التالي:
 - المستخدم يصل إلى خادم الواجهة الأمامية في DMZ (NGINX).
 - NGINX يمرر الطلب إلى خادم التطبيق داخل الشبكة الداخلية (APP/Django).
 - التطبيق يرد بنتيجة الواجهة (صفحة تسجيل الدخول/واجهة الحضور).
- اختبارات تحقق المسار (على مستوى الشبكة):
 - التحقق أن المستخدم يستطيع الوصول إلى عنوان DMZ (اختبار HTTP/Connectivity).
 - التتحقق أن NGINX يستطيع الوصول إلى APP على منفذ الخدمة (8000) في وضع runserver.
- النتائج:
 - وصول المستخدم إلى DMZ يعمل دائماً ضمن السياسة المسموح بها.
 - وصول APP إلى NGINX مسموح فقط لأنه يمثل قناة الخدمة، بينما يُمنع الوصول المباشر من المستخدم إلى APP عند تطبيق ACL النهائي (Strict).
- عند نجاح مسار الخدمة، يصبح النظام جاهزاً وظيفياً ليتمكن الطالب من الوصول إلى صفحة النظام وإتمام عملية تسجيل الحضور عبر واجهة الويب، حيث أن الشبكة أصبحت تدعم تدفق الطلبات بشكل صحيح. . [3][40]

3.7 سياسات التحكم بالوصول (ACL) وآلية التطبيق والتحقق

بعد التأكيد من جاهزية الربط الشبكي والتوجيه بين الشبكات، تم الانتقال إلى مرحلة ضبط سياسات التحكم بالوصول على الموجه R1. تهدف هذه المرحلة إلى تحويل الشبكة من "شبكة تعمل وظيفياً" إلى "شبكة تعمل وظيفياً وبضوابط أمنية"، بحيث يتم تنظيم حركة المرور بين المناطق وفق مبدأ أقل صلاحية (Least Privilege) وبما يضمن أن مسار الخدمة الوحيد المسموح هو المسار الذي يخدم وظيفة النظام الأساسية: وصول الطالب إلى واجهة النظام ثم تنفيذ عملية تسجيل الحضور وبقية العمليات . . [40][41]

3.7.1 أهداف سياسة ACL

تم تصميم الـ ACL لتحقيق الأهداف التالية:

- حماية الخوادم الداخلية عبر منع الوصول المباشر من شبكات المستخدمين إلى شبكة الخوادم الداخلية (Server LAN).
- اعتماد DMZ كنقطة دخول وحيدة بحيث يكون خادم الواجهة الأمامية (NGINX) هو الوجهة الوحيدة للمستخدمين.
- السماح بمسار خدمة محدد ومراقب بين DMZ وخدمي التطبيق (Django) فقط، لأن هذا الاتصال يمثل وظيفة البروكسي/الواجهة الأمامية.
- عزل شبكة الإدارة والسجلات ومنع وصول المستخدمين إليها، مع السماح بما يلزم للإدارة (SSH) وجمع السجلات (Syslog).
- تجنب الانقطاعات أثناء البناء عبر تطبيق السياسة على مراحلتين (Strict ثم Permissive)، لضمان التشخيص السريع ثم الوصول إلى تشديد نهائي جاهز للعرض والتسلیم. [41]

3.7.2 منطق السياسة (Policy Logic) حسب المناطق

اعتمدت السياسة على تحديد "من يسمح له بالوصول لمن؟" وفق الأدوار وليس وفق الأجهزة فقط:

(A) شبكات المستخدمين (Users Networks):

- مسموح: الوصول إلى خادم NGINX في DMZ عبر HTTPS (وأو HTTP عند توفره).
- غير مسموح: الوصول المباشر إلى خادم التطبيق في Server LAN أو إلى شبكة Logs/MGMT أو أي منفذ إداري.

الفصل الثالث

- السبب: المستخدم النهائي لا يحتاج للوصول إلى طبقة التطبيق أو الإدارة، والسماح بذلك يزيد سطح الهجوم ويخالف مبدأ أقل صلاحية.

ب) (NGINX خادم DMZ)

- مسموح: الوصول إلى خادم التطبيق داخل Server LAN على منفذ الخدمة المستخدم في العرض (8000) عند استخدام (Django runserver).
- غير مسموح: فتح اتصالات غير ضرورية إلى باقي الشبكات أو استقبال اتصالات إدارية من المستخدمين.
- السبب : DMZ هي طبقة وسietة، ويجب أن تكون اتصالاتها "محودة" بحيث لا تصبح جسراً مفتوحاً بين المستخدمين والشبكات الحساسة.

ج) (Django Server LAN)

- مسموح: استقبال الطلبات القادمة من NGINX فقط (مسار الخدمة).
- غير مسموح: استقبال طلبات مباشرة من المستخدمين.
- السبب: خادم التطبيق يحتوي منطق النظام وبياناته، لذا يجب أن يكون غير مكشف للمستخدمين.

د) Logs/MGMT

- مسموح: وصول الإدارة عبر SSH إلى الخوادم (APP وNGINX) من شبكة الإدارة فقط، والسماح باستقبال Syslog من الخوادم.
- غير مسموح: وصول المستخدمين إلى شبكة الإدارة أو إلى خدماتها.
- السبب: فصل قناة الإدارة هو عنصر احترافي في التصاميم المؤسسية ويقلل مخاطر هجمات الإدارة ومحاولات التلاعب بالسجلات.[43]

3.7.3 منهجية التطبيق المرحلية (Strict ثم Permissive)

- المرحلة الأولى: سياسة سماحية للاختبار (Permissive Policy)
 - للتحقق من أن جميع العناوين والتوجيه والربط يعمل قبل تطبيق القيود. وبالتالي السماح بحركة المرور الأساسية بين الشبكات لأغراض الاختبار (مثل ping curl). وتسهيل اكتشاف المشاكل المتعلقة بالـ Cloud/VMnet routing أو netplan routing
 - اختبارات التحقق في هذه المرحلة:
 - نجاح ping من كل VM إلى بوابته.
 - نجاح ping بين الخوادم (NGINX ↔ APP، LOG ↔ APP، LOG ↔ NGINX).
 - نجاح الوصول إلى خدمة Django مباشرة على 10.10.100.10:8000 من داخل الشبكات الإدارية/DMZ لأغراض التأكيد.
 - نتيجة هذه المرحلة:

الفصل الثالث

الشبكة مثبتة وظيفياً، وأي خلل يظهر هنا غالباً يكون خللاً في الربط/العنونة وليس في السياسة.

المرحلة الثانية: سياسة التشديد النهائيه (Strict Policy)

- من أجل تطبيق الضوابط الأمنية النهائية بحيث تصبح حركة المرور مطابقة لمسار الخدمة المطلوب فقط.[40]
السياسات:

1. السماح للمستخدمين بالوصول إلى DMZ فقط (HTTP/HTTPS).
2. منع المستخدمين من الوصول إلى Server LAN وLogs/MGMT بشكل مباشر.
3. السماح فقط لـ NGINX بالوصول إلى Django على منفذ الخدمة 8000.
4. حصر SSH من شبكة Logs/MGMT فقط.
5. السماح بإرسال Syslog من الخوادم إلى LOG VM فقط.
6. منع أي اتصالات أخرى افتراضياً (Default Deny).

```
ip access-list extended USERS_IN •  
remark Allow web to DMZ only •  
permit tcp 10.10.10.0 0.0.0.255 host 10.10.200.10 eq 80 •  
permit tcp 10.10.10.0 0.0.0.255 host 10.10.200.10 eq 443 •  
deny ip 10.10.10.0 0.0.0.255 10.10.100.0 0.0.0.255 •  
deny ip 10.10.10.0 0.0.0.255 10.10.250.0 0.0.0.255 •  
permit icmp 10.10.10.0 0.0.0.255 any echo •  
deny ip any any log ip access-list extended DMZ_TO_APP permit tcp host  
10.10.200.10 host 10.10.100.10 eq 8000 •  
deny ip any 10.10.100.0 0.0.0.255 log •  
permit ip any any •
```

3.7.4 اختبارات التحقق النهائيه المرتبطة

تمت صياغة الاختبارات بحيث تتطابق مباشرة مع هدف المشروع النهائي: تمكين الطالب من تسجيل الحضور عبر النظام دون فتح منافذ أو مسارات غير لازمة.

- اختبار (1): وصول المستخدم إلى واجهة النظام عبر DMZ
- الإجراء: من جهاز مستخدم (أو VPCS يمثل مستخدماً) يتم طلب صفحة النظام عبر عنوان DMZ.

الفصل الثالث

- النتيجة: نجاح الوصول إلى 10.10.200.10 (واجهة NGINX).

- الدلالة: المستخدم يستطيع الوصول إلى مدخل النظام الوحيد.

- اختبار (2): منع الوصول المباشر للتطبيق من المستخدمين

- الإجراء: محاولة الوصول إلى 10.10.100.10:8000 من نفس جهاز المستخدم.

.(Timeout/Denied) فشل الوصول النتيجة:

الدالة: لا يمكن تجاوز DMZ، ما يمنع كشف خادم Django للمستخدمين.

- اختبار (3): نجاح مسار الخدمة \rightarrow APP \rightarrow DMZ

الإجراء: يقوم NGINX بتمرير الطلبات إلى Django على VM APP.

النتيجة: ظهر صفحات النظام المستخدم عبر NGINX بشكل طبيعي، ما يعني أن الاتصال

يُعمل 10.10.100.10:8000 → 10.10.200.10

الدلالـة: المسار الـوحيد المسموح للخدمة يـعمل كما خـطـط له.

- اختبار (4): الإدارة الآمنة من شبكة Logs/MGMT

الإجراء: تنفيذ اتصال SSH من LOG VM إلى APP/NGINX

النتيجة : نجاح SSH من شبكة الإدارة فقط، وفشله من شبكات المستخدمين.

الدالة: قناة الادارة مخصوصة ولا تشكل سطح هجوم من المستخدمين.

- اختيار (5): (اختياري) التحقق من إرسال السجلات إلى

الآخر اع: توليد سجل سط من APP أو NGINX (مثل أمر logger).

النتيجة: وصول السجل إلى LOG VM وتخزينه ضمن ملفات السجلات.

الدلالة: دعم التدقيق (Auditing) والمراقبة.[43]

3.7.5 ربط السياسة بالنتيجة الوظيفية النهائية (تسجيل الحضور)

تتضمن سياسةـ ACLـ أن المستخدم النهائي لا يحتاج إلا إلى مسار واحد للوصول إلى النظام: Users → DMZ. بذلك، يقوم NGINX ضمن DMZ بكمال الطلب عبر المسار المسموح Server LAN → DMZ للوصول إلى ذلك. عند نجاح هذا المسار، يستطيع الطالب فتح النظام، تسجيل الدخول إن وجد، ثم تتنفيذ عملية تسجيل الحضور Django. عبر واجهة الويب، بينما تبقى طبقة التطبيق والسجلات والإدارة معزولة عن الوصول المباشر من المستخدمين. وبذلك تتحقق السياسةـ توافقـ بين الوظيفة والأمان: الخدمة تعمل للمستخدم، لكن الوصول غير الضروري محظوظ ومرافق.

3. التحقق النهائي من الهدف الوظيفي: تمكين الطالب من تسجيل الحضور

عبر مسار شبكي آمن

تمثل هذه المرحلة "نقطة التسلیم" للقسم الشبکي في المشروع، إذ لا يقتصر التتحقق على نجاح الاتصال التقني (ping والتجویه) فحسب، بل يمتد لإثبات أن الشبکة وسياسات التحكم بالوصول (ACL) تدعم الهدف الوظيفي النهائي للمشروع: تمكين الطالب من الوصول إلى النظام وتنفيذ عملية تسجيل الحضور عبر مسار صحيح ومقصود، دون فتح مسارات غير لازمة إلى الطبقات الحساسة.

3.8.1 تعريف سيناريو الاستخدام النهائي (End-to-End Scenario)

تم تعريف سيناريو التتحقق النهائي بحيث يحاكي الاستخدام الحقيقي للنظام:

1. الطالب (ضمن شبكة المستخدمين) يحاول فتح النظام عبر متصفح/طلب HTTP.
2. الطلب يجب أن يصل إلى نقطة الدخول المعلنة فقط (خادم NGINX في DMZ).
3. خادم NGINX يقوم بإعادة توجيه الطلبات (Reverse Proxy) إلى خادم التطبيق Django داخل الشبکة الداخلية.
4. خادم Django يعالج الطلبات ويعرض صفحات النظام، ثم يتم تنفيذ إجراء "تسجيل الحضور".
5. يتم التأكيد أن العملية تمت بنجاح وأن النظام استجاب ضمن القيود الأمنية المحددة.

هذا السيناريو يضمن أن التتحقق لا يختبر مكوناً واحداً بمعزل، بل يختبر التكامل بين الشبکة والطبقات الخدمية.

3.8.2 التحضير لاختبار النهائي (Prerequisites)

قبل تنفيذ التتحقق النهائي، تم التأكيد من جاهزية العناصر التالية:

أولاً - جاهزية خدمة التطبيق (APP VM):
تشغيل Django باستخدام:
`python manage.py runserver 0.0.0.0:8000`
نجاح الترحيلات:
`python manage.py migrate`
وجود قاعدة بيانات SQLite داخل APP VM (مثلًا db.sqlite3) بما يثبت أن النظام قادر على تخزين بيانات
الحضور محلياً.

الفصل الثالث

ثانياً - جاهزية طبقة العرض (NGINX VM):

تشغيل خدمة NGINX والتأكد من سلامة الإعداد:

systemctl restart nginx -t

ضبط Reverse Proxy ليحول طلبات المستخدمين إلى:

(DMZ) (من منظور NGINX داخل http://10.10.100.10:8000)

ثالثاً - جاهزية الشبكة والسياسات (R1 + ACL):

التأكد من أن واجهات الراوتر up/up.

التأكد من تطبيق ACL على الواجهات الصحيحة (in/out) حسب سياسة التشديد النهائية.

التأكد من أن شبكات المستخدمين لا تملك مسار مباشر إلى Server LAN إلا عبر DMZ وبحسب المسموح.

3.8.3 تصميم مسار الخدمة (Service Path) كمطلوب تحقق

تم اعتبار "مسار الخدمة" شرطاً رسمياً في التحقق النهائي، وتم تعريفه من منظور الشبكة كالتالي:

المسار المسموح للمستخدم:

HTTP/HTTPS Users → DMZ (NGINX) فقط.

المسار المسموح للخدمة داخلياً:

.(runserver TCP/8000 على منفذ الخدمة DMZ (NGINX) → Server LAN (Django)

المسارات غير المسموحة:

Users → Logs/MGMT، و Users → Logs/MGMT، وأي منافذ إدارية من المستخدمين.
هذا التعريف يجعل نجاح الاختبار النهائي ليس "وصول صفحة فقط"، بل "وصول صفحة عبر المسار الصحيح فقط".

3.8.4 خطوات التحقق العملي ونتائجها المتوقعة (Tests & Expected Results)

الاختبار (T1): وصول الطالب إلى بوابة النظام عبر DMZ

الهدف: إثبات أن الطالب يستطيع الوصول إلى نقطة الدخول الوحيدة للنظام.

الإجراء: من جهاز يمثل الطالب (VM أو VPCS) ضمن شبكة المستخدمين يتم محاولة الوصول إلى عنوان DMZ:

فتح http://10.10.200.10/ عبر المتصفح أو تنفيذ:

curl <http://10.10.200.10>

النتيجة المتوقعة:

الفصل الثالث

الحصول على استجابة HTTP (مثل 200/302) وظهور صفحة النظام (تسجيل دخول/واجهة).
الدالة: طبقة DMZ تعمل كنقطة دخول رسمية والطالب يستطيع بدء استخدام النظام.

الاختبار (T2): التأكد من أن الطالب لا يستطيع تجاوز DMZ والوصول مباشرة إلى Django

الهدف: إثبات أن Server LAN غير مكشوف للمستخدمين.
الإجراء: من نفس جهاز الطالب يتم محاولة الوصول مباشرة إلى خادم Django:

: http://10.10.100.10:8000 أو

curl http://10.10.100.10:8000

النتيجة المتوقعة:

فشل الاتصال (Timeout/Denied) وعدم ظهور صفحة التطبيق مباشرة.
الدالة: تحقق مبدأ Least Privilege وتنمنع الوصول غير المصرح به للطبقة الحساسة.

الاختبار (T3): نجاح قناة الخدمة الوحيدة APP → DMZ

الهدف: إثبات أن NGINX قادر على الوصول إلى Django داخلياً وفق السياسة.
الإجراء: من داخل VM NGINX يتم اختبار الوصول إلى Django:

curl http://10.10.100.10:8000

ثم يتم إعادة اختبار فتح الصفحة من جهاز الطالب عبر NGINX

/http://10.10.200.10

النتيجة المتوقعة:

نجاح الوصول من APP إلى NGINX.
نجاح عرض صفحات النظام للطالب عبر NGINX.
الدالة: الخدمة تعمل عبر المسار الآمن المسموح فقط، ولا يوجد كسر للسياسة الأمنية.

الاختبار (T4): تنفيذ عملية تسجيل الحضور والتحقق من أثرها (Functional Impact)

الهدف: ربط الشبكة بنتيجة وظيفية ملموسة (تسجيل الحضور).

الإجراء: من حساب طالب/مستخدم ضمن النظام يتم تنفيذ عملية "Attendance / Check-in" من الواجهة.
(إذا كان النظام يستخدم تسجيل دخول، يتم تسجيل الدخول أولاً، ثم تنفيذ العملية.)

النتيجة المتوقعة:

الفصل الثالث

ظهور رسالة نجاح أو تحذير في الواجهة يدل على أن الحضور تم تسجيله.

إمكانية رؤية سجل حضور جديد ضمن الواجهة أو ضمن لوحة الإدارة (إن كانت مفعلة).
الدلالـة: الشبكة لم تكـفـي بـإيـصالـ صـفـحةـ، بل دعـمـتـ تنـفـيـذـ عمـلـيـةـ حـاسـسـةـ تـنـطـلـبـ اـتـصـالـاـ وـاسـتـجـابـةـ صـحـيـحةـ منـ خـادـمـ التـطـبـيقـ وـقـاعـدـةـ الـبـيـانـاتـ.

الاختبار (T5): التحقق من حفظ البيانات (Persistence) داخل SQLite [46][47]

الهدف: إثبات أن بيانات الحضور لا تتضمن وأنها مخزنة فعليـاـ.

الإجراء: بعد تنفيذ تسجيل الحضور:

إعادة تحميل الصفحة (Refresh) والتحقق أن السجل موجود.
(اختياري) إعادة تشغيل خدمة runserver ثم التحقق مجدداً.

النتـيـجةـ المتـوقـعةـ:

بقاء السجل موجوداً بعد إعادة التحميل، ما يؤكد أن البيانات محفوظة في قاعدة البيانات.
الدلالـةـ: التـكـامـلـ بـيـنـ وـاجـهـةـ الـمـسـتـخـدـمـ وـالـتـطـبـيقـ وـالتـخـرـيـنـ يـعـملـ بـشـكـلـ صـحـيـحـ.

3.8.5 تفسير النتائج وربطها بأهداف الأمن والوظيفة

نجاح الاختبارات السابقة يثبت أن تصميم الشبكة و ACL حقق الأهداف التالية بالتزامن:

وظيفياً: الطالب يستطيع الوصول للنظام وتنفيذ عملية تسجيل الحضور بنجاح.

أمنياً: الوصول يتم عبر DMZ فقط، وخدمـةـ التطبيقـ غيرـ مـكـشـوفـ للمـسـتـخـدـمـينـ، وـقـنـواتـ الإـدـارـةـ مـحـصـوـرـةـ، وـمـسـارـ الخـدـمـةـ مـحـدـودـ وـوـاصـحـ.

تشغيلياً: التحقق موثق بالأوامر والنتائج، ما يجعل تقييمه قابلاً للمراجعة من اللجنة دون الاعتماد على الانطباع.

3.8.6 نتائج القسم الشبكي (Network Section Outcome)

بناءً على التحقق النهائي، يمكن اعتبار البنية الشبكية "جاهزة للاستخدام ضمن نطاق المشروع" لأنها:

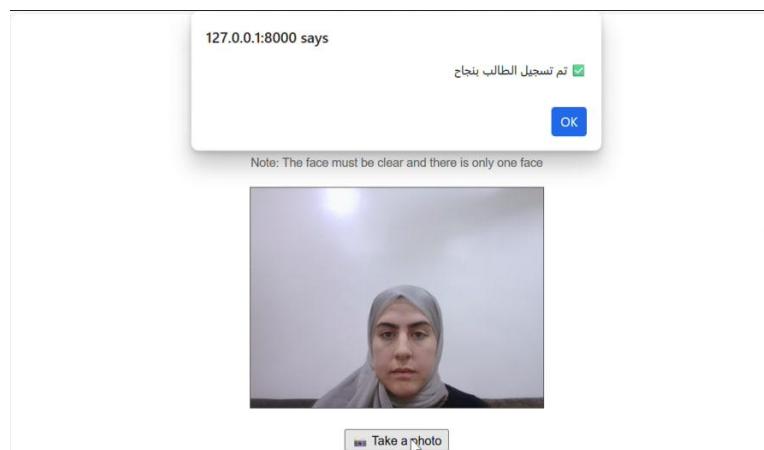
1. تحقق العزل الشبكي الطبيعي.
2. تحقق وصول الخدمة المستخدم النهائي عبر المسار الصحيح.
3. تدعم إثباتات واضحة (Ping/HTTP/ACL Counters).
4. ترتبط مباشرةً بهدف المشروع النهائي (تسجيل حضور).

الفصل الرابع

يهدف هذا الفصل إلى عرض وتحليل النتائج المتحققة من تنفيذ نظام تفهيم الحضور الإلكتروني المعتمد على التعرف على الوجه، والمدعوم ببنية برمجية وشبكية آمنة. ويركز الفصل على تقييم مدى نجاح التصميم المقترن في تحقيق أهداف المشروع، وذلك من خلال تحليل نتائج التشغيل العملي، والاختبارات الوظيفية، والاختبارات الأمنية التي أجريت على النظام ضمن بيئه محاكاة تمثل مؤسسة تعليمية جامعية.

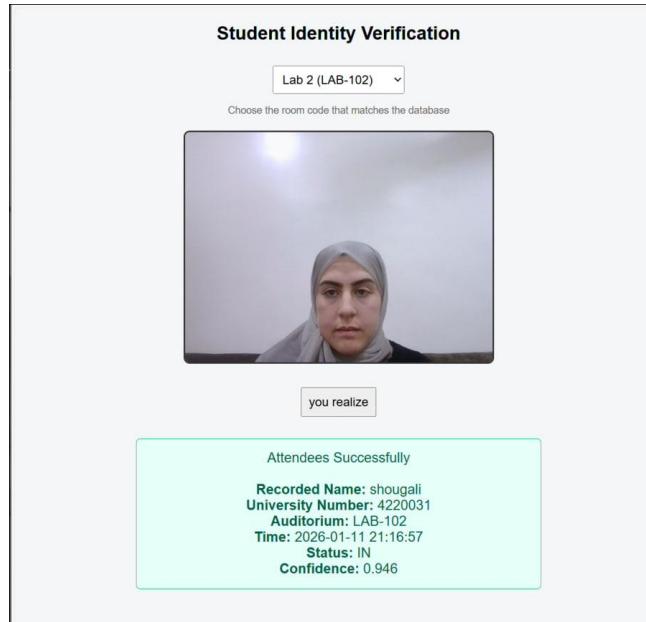
4.1 نتائج البنية البرمجية للنظام

أظهرت نتائج التنفيذ أن البنية البرمجية للنظام نجحت في تحقيق الوظائف الأساسية المطلوبة بكفاءة واستقرار. فقد تمكّن النظام من تنفيذ عمليات تسجيل الطلبة (Face Enrollment) واستخراج بصمة الوجه الرقمية (Face Embedding) باستخدام نموذج ArcFace، وربطها بهوية الطالب داخل قاعدة البيانات بشكل صحيح. كما أثبتت آلية التحقق (Verification) قدرتها على التعرف على الطلبة المسجلين بدقة مناسبة، وتسجيل الحضور تلقائياً دون تدخل بشري.

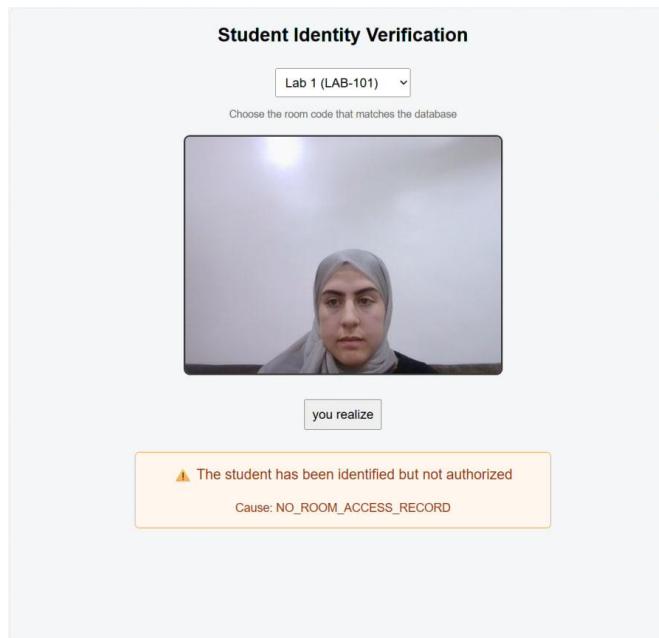


الشكل (4.1): نتائج التعرف على الوجه وتسجيل الحضور بنجاح

الفصل الرابع



الشكل (4.2): نتائج عملية التحقق (Verification) وتسجيل الحضور مع قيمة الثقة



الشكل (4.3): نتائج عملية التتحقق (Verification) مع فشل التفویض (Authorization Failure)

Attendance Record						
Trust	Status	The Hall	University Number	Student	Time	
0.95	FORBIDDEN	Lab 1	4220031	shougali	12/01/2026, 00:17:10	
0.95	IN	Lab 2	4220031	shougali	12/01/2026, 00:16:57	
0.85	FORBIDDEN	Lab 2	4223311	husain ali	11/01/2026, 21:08:44	
0.87	IN	Lab 1	4223311	husain ali	11/01/2026, 21:04:11	

(الشكل (4.4): مثال على مخرجات نظام الحضور الإلكتروني (IN / FORBIDDEN)

كما بينت نتائج التشغيل أن الفصل بين منطق التعرف على الوجه ومنطق التفويض (Authorization) أسهم في تحسين موثوقية القرار النهائي، حيث لم يقتصر النظام على التعرف على هوية الطالب فقط، بل تحقق أيضًا من مدى أحقيته بالدخول إلى القاعة المحددة. وقد انعكس ذلك في تسجيل حالات متعددة مثل (OUT, FORBIDDEN, IN) بشكل واضح داخل لوحة التحكم.

4.2 نتائج البنية الشبكية والأمنية

أظهرت نتائج الاختبارات الشبكية أن البنية المعتمدة وفرت مستوى جيداً من العزل بين مكونات النظام المختلفة. فقد نجح استخدام البنية متعددة الطبقات ومنطقة العزل الشبكي (DMZ) في منع الوصول المباشر من شبكات المستخدمين إلى خوادم التطبيق وقواعد البيانات الداخلية، مما قلل من سطح الهجوم المحتمل.

كما أثبت تطبيق سياسات التحكم بالوصول باستخدام قوائم ACL فاعليته في حصر حركة المرور بالمسارات الضرورية فقط، حيث سُمح للمستخدمين بالوصول إلى واجهة النظام عبر الخادم الوسيط (Reverse Proxy) دون تمكين أي اتصال

الفصل الرابع

مباشر بالحوادم الداخلية. وأظهرت الاختبارات أن أي محاولة وصول غير مصرح بها يتم حجبها وتسجيلها، مما يعزز الجانب الأمني للنظام.

4.3 نتائج التسجيل المركزي للأحداث والتدقيق

بيّنت نتائج التطبيق أن آلية التسجيل المركزي للأحداث (Centralized Logging) أسهمت في تعزيز قابلية المراقبة والتتبع. فقد تم تسجيل جميع العمليات الحساسة، مثل تسجيل الطلبة، ومحاولات التحقق الناجحة والفاشلة، ومحاولات الوصول غير المصرح بها، ضمن سجلات تدقيق واضحة تحتوي على معلومات السياق الزمني وعنوان الشبكة والنتيجة النهائية.

وقد أتاح ذلك إمكانية تتبع "من قام بماذا ومتى"، وهو ما يدعم مبدأ المحاسبة (Accountability) ويساعد في تحليل الأخطاء والحوادث الأمنية المحتملة، ويُعد مؤشراً مهماً على نضج التصميم الأمني للنظام.

4.4 مقارنة النتائج مع أهداف المشروع

عند مقارنة النتائج المتحققة مع الأهداف المحددة في الفصل الأول، يتضح أن النظام نجح في تحقيق معظم أهداف المشروع. فقد تحقق هدف أتمته عملية تفقد الحضور باستخدام التعرف على الوجه، والحد من التلاعب والانتحال، إضافة إلى بناء بنية برمجية مرنّة وقابلة للتتوسيع. كما تحققت الأهداف الأمنية المتمثلة في عزل المكونات الحساسة، وتطبيق مبدأ أقل صلاحية، ودعم المراقبة والتدقيق.

4.5 التحديات التي ظهرت أثناء التنفيذ

على الرغم من نجاح النظام في تحقيق أهدافه الأساسية، أظهرت النتائج بعض التحديات العملية، مثل تأثير دقة التعرف على الوجه في حالات الإضاءة غير المناسبة، وال الحاجة إلى ضبط عتبة الثقة (Threshold) بعناية لقليل حالات الرفض أو القبول الخاطئ. كما تطلب إعداد سياسات ACL وضبط الشبكة مراحل اختبار تدريجية لتفادي أخطاء الضبط التي قد تؤثر على استمرارية الخدمة.

الخاتمة

في ظل التطور المتسارع في التقنيات الرقمية والاعتماد المتزايد على الأنظمة الإلكترونية في المؤسسات التعليمية، برزت الحاجة إلى تطوير حلول تقنية موثوقة تسهم في تحسين الكفاءة الإدارية وضمان دقة العمليات الأكademية. وقد تناول هذا العمل تصميم دراسة نظام تفقد حضور إلكتروني يعتمد على تقنية التعرف على الوجه، بوصفها إحدى تقنيات القياسات الحيوية القادرة على معالجة أوجه القصور المرتبطة بالأساليب التقليدية المعتمدة في تسجيل الحضور.

وقد ركّز العمل على دمج الجوانب النظرية والتطبيقية المتعلقة بالبنية البرمجية والبنية الشبكية الآمنة، بما يسهم في بناء نظام يتمتع بدرجة مناسبة من الدقة والاعتمادية. كما أظهرت النتائج أن اعتماد مفاهيم مثل البنية متعددة الطبقات، ومنطقة العزل الشبكي، والتحكم بالوصول، يسهم في تعزيز مستوى الأمان وحماية البيانات الحساسة.

وفي ضوء ما تم التوصل إليه، يمكن اعتبار النظام المقترن خطوة داعمة نحو تطوير أنظمة تفقد حضور إلكترونية أكثر كفاءة وقابلية للتطبيق في المؤسسات التعليمية، مع إمكانية تطويره مستقبلاً لتحسين الأداء وتوسيع نطاق الاستخدام بما يتناشئ مع متطلبات التحول الرقمي في التعليم.

قائمة المصطلحات

- البنية متعددة الطبقات (Three-Tier Architecture)
- التسجيل المركزي للأحداث (Centralized Logging)
- التشفير (Encryption)
- التشفير أثناء التخزين (Encryption at Rest)
- التشفير أثناء النقل (Encryption in Transit)
- التعلم العميق (Deep Learning)
- التعرف على الوجه (Face Recognition)
- الذكاء الاصطناعي (Artificial Intelligence – AI)
- العميل (Client)
- القياسات الحيوية (Biometrics)
- الجدار النارى (Firewall)
- الخادم (Server)
- خادم التطبيق (Application Server)
- خادم قاعدة البيانات (Database Server)
- كشف الوجه (Face Detection)
- قائمة التحكم بالوصول (ACLs – Access Control Lists)
- قاعدة البيانات (Database)
- منطقة العزل الشبكي (DMZ – Demilitarized Zone)
- نظام تفقد الحضور الإلكتروني (Electronic Attendance System)
- نموذج التعرف (Recognition Model)
- واجهة المستخدم (User Interface – UI)
- واجهة برمجة التطبيقات (API – Application Programming Interface)
- بصمة الوجه (Face Embedding)
- فصل الصالحيات والمهام (Separation of Duties)

قائمة الاختصارات (List of Abbreviations)

- ACL — Access Control List
- AI — Artificial Intelligence
- API — Application Programming Interface
- CNN — Convolutional Neural Network
- CPU — Central Processing Unit
- DB — Database
- DMZ — Demilitarized Zone
- DNS — Domain Name System
- GUI — Graphical User Interface
- HTTP — HyperText Transfer Protocol
- HTTPS — HyperText Transfer Protocol Secure
- IAM — Identity and Access Management
- IP — Internet Protocol
- JWT — JSON Web Token
- LAN — Local Area Network
- ML — Machine Learning
- OS — Operating System
- RAM — Random Access Memory
- RBAC — Role-Based Access Control
- REST — Representational State Transfer
- SSL — Secure Sockets Layer
- TLS — Transport Layer Security
- UI — User Interface
- URL — Uniform Resource Locator
- WAN — Wide Area Network

References

1. Deng, J., Guo, J., Xue, N., & Zafeiriou, S. (2019). ArcFace: Additive angular margin loss for deep face recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 4690–4699).
https://openaccess.thecvf.com/content_CVPR_2019/html/Deng_ArcFace_Additive_Angular_Margin_Loss_for_Deep_Face_Recognition_CVPR_2019_paper.html
2. Grother, P., Ngan, M., & Hanaoka, K. (2020). Face Recognition Vendor Test (FRVT) Part 3: Demographic effects (NISTIR 8280). National Institute of Standards and Technology.
<https://doi.org/10.6028/NIST.IR.8280>
3. Fielding, R., & Reschke, J. (2014). Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing (RFC 7230). RFC Editor. <https://doi.org/10.17487/RFC7230>
4. Rescorla, E. (2018). The Transport Layer Security (TLS) Protocol Version 1.3 (RFC 8446). RFC Editor. <https://doi.org/10.17487/RFC8446>
5. Jones, M., Bradley, J., & Sakimura, N. (2015). JSON Web Token (JWT) (RFC 7519). RFC Editor. <https://doi.org/10.17487/RFC7519>
6. Krawczyk, H., Bellare, M., & Canetti, R. (1997). HMAC: Keyed-Hashing for Message Authentication (RFC 2104). RFC Editor. <https://doi.org/10.17487/RFC2104>

-
7. Eby, P. J. (2010). PEP 3333 – Python Web Server Gateway Interface (WSGI) v1.0.1. Python Enhancement Proposals. <https://peps.python.org/pep-3333/>
8. Django Software Foundation. (n.d.). Django documentation: Settings (Security settings, ALLOWED_HOSTS, CSRF, Sessions, HSTS). Retrieved January 29, 2026, from <https://docs.djangoproject.com/en/5.0/ref/settings/>
9. Django REST framework. (n.d.). Authentication: TokenAuthentication. Retrieved January 29, 2026, from <https://www.django-rest-framework.org/api-guide/authentication/#tokenauthentication>
10. djangorestframework-simplejwt contributors. (n.d.). Simple JWT documentation (JWTAuthentication). Retrieved January 29, 2026, from <https://django-rest-framework-simplejwt.readthedocs.io/en/latest/>
11. Gunicorn Project. (n.d.). Gunicorn documentation: Settings (bind, workers, deployment). Retrieved January 29, 2026, from <https://docs.gunicorn.org/en/stable/settings.html>
12. World Wide Web Consortium (W3C). (n.d.). Media Capture and Streams (getUserMedia). Retrieved January 29, 2026, from <https://www.w3.org/TR/mediacapture-streams/>
13. OpenCV. (n.d.). OpenCV documentation: Color space conversions (cvtColor). Retrieved January 29, 2026, from <https://docs.opencv.org/>
14. ONNX Runtime. (n.d.). ONNX Runtime documentation (Python API / InferenceSession). Retrieved January 29, 2026, from <https://onnxruntime.ai/docs/>
15. Python Cryptographic Authority. (n.d.). cryptography documentation: Fernet (symmetric encryption). Retrieved January 29, 2026, from <https://cryptography.io/en/latest/fernet/>

16. OWASP Foundation. (2021). OWASP Top 10: The Ten Most Critical Web Application Security Risks. <https://owasp.org/Top10/>

17 . Anti-Phishing Working Group. (2025, March 10). Phishing activity trends reports. APWG. Retrieved from <https://apwg.org/trendsreports/>

18 . Del Pozo, I., Iturralde, M., & Restrepo, F. (2018, August). Social engineering: Application of psychology to information security. In 2018 6th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW) (pp. 108–114). IEEE. <https://doi.org/10.1109/FiCloudW.2018.00027>

19 . IBM. (2024, May). What is phishing? IBM. Retrieved from <https://www.ibm.com/think/topics/phishing>

20 .Mishra, S., & Soni, D. (2023). DSmishSMS—A system to detect smishing SMS. Neural Computing and Applications, 35(7), 4975–4992.
<https://doi.org/10.1007/s00521-022-07724-4>

21 . Figueiredo, J., Carvalho, A., Castro, D., Gonçalves, D., & Santos, N. (2024). On the feasibility of fully AI-automated vishing attacks. arXiv preprint arXiv:2409.13793. <https://arxiv.org/abs/2409.13793>

22 . ISACA. (2024). State of cybersecurity 2024. ISACA. Retrieved from <https://www.isaca.org/resources/reports/state-of-cybersecurity-2024>

23 . Frauenstein, E. D., & Flowerday, S. (2020). Susceptibility to phishing on social

network sites: A personality information processing model. *Computers & Security*, 94, 101862. <https://doi.org/10.1016/j.cose.2020.101862>

24. Yang, L., Zhang, J., Wang, X., Li, Z., Li, Z., & He, Y. (2021). An improved ELM-based and data preprocessing integrated approach for phishing detection considering comprehensive features. *Expert Systems with Applications*, 165, 113863.

<https://doi.org/10.1016/j.eswa.2020.113863>

25. Basnet, R. B., Sung, A. H., & Liu, Q. (2011, July). Rule-based phishing attack detection. In *Proceedings of the International Conference on Security and Management (SAM)* (Vol. 1, No. 1, pp. 624–630). CSREA Press.

26. Safi, A., & Singh, S. (2023, February). A systematic literature review on phishing website detection techniques. *Journal of King Saud University – Computer and Information Sciences*, 35(2), 590–611.

<https://doi.org/10.1016/j.jksuci.2020.11.027>

27. Alsariera, Y. A., Alanazi, M. H., Said, Y., & Allan, F. (2024, June). An investigation of AI-based ensemble methods for the detection of phishing attacks. *Engineering, Technology & Applied Science Research*, 14(3), 14266–14274.

<https://doi.org/10.48084/etasr.6895>

7312. Rashid, H., Liaqat, H. B., Sana, M. U., Kiren, T., Karamti, H., & Ashraf, I. (2025, May). Framework for detecting phishing crimes on Twitter using selective features and machine learning. *Computers & Electrical Engineering*, 124, 110363.

<https://doi.org/10.1016/j.compeleceng.2025.110363>

-
28. Asiri, S., Xiao, Y., Alzahrani, S., Li, S., & Li, T. (2023). A survey of intelligent detection designs of HTML URL phishing attacks. *IEEE Access*, 11, 6421–6443.
<https://doi.org/10.1109/ACCESS.2023.3236247>
29. Alzaidah, R., Al-Shaikh, A., Al-Mousa, M. R., Khafajah, H., Samara, G., Alzyoud, M., Al-Shanableh, N., & Almatarneh, S. (2023, July). Website phishing detection using machine learning techniques. *Journal of Statistics Applications & Probability*, 13(1), 119–129. <https://doi.org/10.18576/jsap/130113>
30. Mat Rani, L., Mohd Foozy, C., & Mustafa, S. (2023, January). Feature selection to enhance phishing website detection based on URL using machine learning techniques. *Journal of Soft Computing and Data Mining*, 4(1), 30–41.
31. Albishri, A. A., & Dessouky, M. M. (2024, December). A comparative analysis of machine learning techniques for URL phishing detection. *Engineering, Technology & Applied Science Research*, 14(6), 18495–18501.
<https://doi.org/10.48084/etasr.7267>
32. Sudar, K. M., Rohan, M., & Vignesh, K. (2024, August). Detection of adversarial phishing attack using machine learning techniques. *Sādhanā*, 49(3), 232.
<https://doi.org/10.1007/s12046-024-02474-3>
33. Tamal, M. A., Islam, M. K., Bhuiyan, T., Sattar, A., & Prince, N. U. (2024, July). Unveiling suspicious phishing attacks: Enhancing detection with an optimal feature vectorization algorithm and supervised machine learning. *Frontiers in*

34. Paithane, P. M. (2025, April). URLGuard: A holistic hybrid machine learning approach for phishing detection. International Journal of Information Engineering and Electronic Business, 17(2), 95–110. <https://doi.org/10.5815/ijieeb.2025.02.07>
35. Karim, A., Shahroz, M., Mustofa, K., Belhaouari, S., & Joga, S. (2023). Phishing detection system through hybrid machine learning based on URL. IEEE Access, 11, 36805–36822. <https://doi.org/10.1109/ACCESS.2023.3272715>
36. Aldakheel, E. A., Zakariah, M., Gashgari, G. A., Almarshad, F. A., & Alzahrani, A. I. A. (2023, April). A deep learning-based innovative technique for phishing detection in modern security with uniform resource locators. Sensors, 23(9), 4403. <https://doi.org/10.3390/s23094403>
37. Asiri, S., Xiao, Y., & Li, T. (2023, December). PhishTransformer: A novel approach to detect phishing attacks using URL collection and transformer. Electronics, 13(1), 30. <https://doi.org/10.3390/electronics13010030>
38. Kavya, S., & Sumathi, D. (2025). Multimodal and temporal graph fusion framework for advanced phishing website detection. IEEE Access, 13, 74128–74146. <https://doi.org/10.1109/ACCESS.2025.3506278>
39. Remya, S., Pillai, M., Aparna, B., Subbareddy, S. R., & Cho, Y. (2025). BGL-PhishNet: Phishing website detection using hybrid model—BERT, GNN, and LightGBM. IEEE Access, 13, 47552–47569.

<https://doi.org/10.1109/ACCESS.2025.3378865>

40. NIST. (2009).

SP 800-41 Rev.1: Guidelines on Firewalls and Firewall Policy.

National Institute of Standards and Technology.

<https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-41r1.pdf>

41. Saltzer, J. H., & Schroeder, M. D. (1975).

The Protection of Information in Computer Systems.

Proceedings of the IEEE, 63(9), 1278–1308.

<https://ieeexplore.ieee.org/document/1451869>

42. Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G. J., & Lear, E. (1996).

RFC 1918: Address Allocation for Private Internets.

RFC Editor.

<https://www.rfc-editor.org/rfc/rfc1918>

43. Gerhards, R. (2009).

RFC 5424: The Syslog Protocol.

RFC Editor.

<https://www.rfc-editor.org/rfc/rfc5424>

44. Souppaya, M., Scarfone, K., & Hoffman, P. (2011).

NIST SP 800-125: Guide to Security for Full Virtualization Technologies.

National Institute of Standards and Technology.

<https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-125.pdf>

45. Django Software Foundation. (2025).

Deployment Checklist.

Django Documentation.

<https://docs.djangoproject.com/en/stable/howto/deployment/checklist/>

46. Gaffney, S., et al. (2022).

SQLite: Past, Present, and Future.

Proceedings of the VLDB Endowment, 15(12).

<https://www.vldb.org/pvldb/vol15/p3535-gaffney.pdf>

47. SQLite.org. (2025).

File Locking and Concurrency in SQLite Version 3.

<https://www.sqlite.org/lockingv3.html>