الجمهورية العربية السورية

وزارة التعليم العالي والبحث العلمي

الجامعة الوطنية الخاصة

كلية الهندسة

قسم هندسة الحاسوب

مشروع أعد لنيل درجة البكالوريوس في هندسة الحاسوب بعنوان:

نظام إدارة رياض الأطفال

إعداد الطالبتين:

خديجة العقاد

غفران البغاج

إشراف:

د.م. هاني الخطيب

2025-2024

المحتويات

1 الفصل الأول: المقدمة 1

1-1 نطاق المشروع 2

2-1 أهمية المشروع 3

3-1 أهداف المشروع 4

4-1 نطاق المشروع وحدوده 5

5-1 الفرضيات والتحديات 6

1-5-1 الفرضيات 6

2-5-1 التحديات 7

6-1 منهجية العمل والتنفيذ 7

7-1 الهيكلية 8

2 الفصل الثاني: الدراسات السابقة والمفاهيم النظرية 10

1-2 مقدمة الفصل 11

2-2 أنظمة إدارة رياض الأطفال 11

1-2-2 تعريف أنظمة رياض الأطفال 11

2-2-2 الوظائف العامة لأنظمة إدارة رياض الأطفال 12

3-2-2 أمثلة لأنظمة إدارة رياض الأطفال 14

4-2-2 الفجوات والتحديات في الأنظمة الحالية 20

3-2 التقنيات المستخدمة في بناء النظام 21

1-3-2 إطار عمل Flutter 21

2-3-2 إطار العمل Laravel وPHP لتطوير الواجهات الخلفية 26

3-3-2 إطار العمل Bootstrap 34

3 الفصل الثالث: التحليل وتصميم النظام 43

1-3 متطلبات النظام 44

1-1-3 المتطلبات الوظيفية 44

2-1-3 المتطلبات غير الوظيفية 45

2-3 تحليل المستخدمين والأدوار 46

3-3 تحليل الحالات والاستخدامات (Use Case Analysis) 47

4-3 مخطط ERD 50

5-3 تصميم قواعد البيانات 51

4 الفصل الرابع: التنفيذ 54

1-4 واجهة لوحة تحكم المدير (Web Dashboard Interface) 55

4-1-1 واجهة تسجيل الدخول 55

4-1-2 لوحة التحكم الرئيسية 55

4-1-3 الحضور والغياب 56

4-1-4 إدارة الأطفال 57

4-1-5 إدارة الفصول 58

4-1-6 إدارة أولياء الأمور 59

4-1-7 إدارة الوجبات 59

4-1-8 إدارة الإعلانات 60

4-1-9 إدارة الفعاليات 61

4-1-10 المصادر التعليمية 61

4-1-11 إدارة الوسائط 62

4-1-12 الرسائل الواردة 63

4-1-13 ملاحظات أولياء الأمور 63

4-2 واجهات التطبيق 64

4-3 النتائج 75

5 الفصل الخامس: الخاتمة والتطويرات المستقبلية 76

6 المراجع 84

فهرس الأشكال

الشكل 1-2: مثال على أنظمة إدارة رياض الأطفال. 12

الشكل 2-2: شعار نظام Brightwheel 14

الشكل 3-2: واجهة الموقع والتطبيق. 15

الشكل 4-2: نظام إدارة حضور ونشاطات الأطفال في الحضانة والمدرسة باستخدام تطبيق رقمي متكامل. 17

الشكل 5-2: شعار نظام Illumine 18

الشكل 6-2: واجهة نظام Illumine: تتبع الحضور، تحديثات الوجبات، الموافقات على الإجازات، والمدفوعات المالية 19

الشكل 7-2: شعار إطار العمل Flutter 21

الشكل 8-2: شعار لغة البرمجة Dart. 22

الشكل 9-2: الطبقات المعمارية لإطار العمل Flutter 23

الشكل 10-2: تطبيق حجز غرف الاجتماعات - شركةNIX 26

الشكل 11-2: شعار إطار العمل Laravel. 27

الشكل 12-2: بنية MVC 28

الشكل 13-2: RESTful API. 31

الشكل 14-2: Eloquent ORM. 32

الشكل 15-2: شعار لغة البرمجة PHP. 33

الشكل 2-16: شعار Apache HTTP Server 33

الشكل 2-17: شعار إطار العمل Bootstrap..... 35

الشكل 2-18: بعض المكونات التي يوفرها إطار العمل Bootstrap..... 38

الشكل 3-1: Use Case Analysis Diagram. 48

الشكل 3-2: مخطط ERD..... 50

الشكل 3-3: مخطط قاعدة البيانات..... 51

الشكل 4-1: واجهة تسجيل الدخول..... 55

الشكل 4-2: لوحة التحكم الرئيسية..... 56

الشكل 4-3: واجهة الحضور والغياب..... 57

الشكل 4-4: واجهة إدارة الأطفال..... 58

الشكل 4-5: واجهة إدارة الفصول الدراسية..... 58

الشكل 4-6: واجهة إدارة أولياء الأمور. 59

الشكل 4-7: واجهة إدارة الوجبات اليومية..... 60

الشكل 4-8: واجهة إدارة الإعلانات..... 60

الشكل 4-9: إدارة الفعاليات والرحلات..... 61

الشكل 4-10: واجهة إدارة المصادر التعليمية..... 62

الشكل 4-11: واجهة إدارة الوسائط. 62

الشكل 4-12: واجهة صندوق الرسائل..... 63

الشكل 13-4: واجهة ملاحظات أولياء الامور. 64

الشكل 14-4: القائمة الرئيسية. 66

الشكل 15-4: الفعاليات القادمة. 67

الشكل 16-4: جدول الأنشطة الأسبوعي. 68

الشكل 17-4: مؤشرات الطالب الصحية. 69

الشكل 18-4: الحساب الشخصي لولي الأمر. 70

الشكل 19-4: صندوق الوارد. 71

الشكل 20-4: الفصول الدراسية. 72

الشكل 21-4: الصفحة الرئيسية. 73

الشكل 22-4: معرض الألعاب للأطفال. 74

ملخص المشروع

تم في هذا المشروع تصميم وتطوير نظام متكامل لإدارة رياض الأطفال، يهدف إلى تعزيز الكفاءة التشغيلية وتحسين آليات التواصل بين إدارة الروضة وأولياء الأمور. اعتمد النظام على بنية تقنية حديثة تستخدم إطار عمل Flutter لتطوير واجهة أمامية تفاعلية وموحدة للأجهزة المحمولة، بينما تم بناء الوظائف الخلفية باستخدام إطار عمل Laravel المستند إلى لغة PHP، مما وفر بنية تحتية قوية لإدارة البيانات والعمليات. تم تصميم النظام ليخدم ثلاثة أدوار رئيسية: ولي الأمر، الذي يمكنه متابعة تقدم طفله الأكاديمي والصحي، والاطلاع على الأنشطة والفعاليات، والتواصل المباشر مع الإدارة؛ الطفل، الذي يتمحور حوله النظام من خلال سجلات الحضور، والتقارير الصحية، والملاحظات السلوكية؛ والإدارة، التي تمتلك الأدوات اللازمة لإدارة بيانات الأطفال، الفصول الدراسية، الإعلانات، والجداول اليومية.

شملت المساهمات الأساسية للمشروع بناء واجهة برمجة تطبيقات (API) شاملة وموثقة وفق معايير OpenAPI (YAML)، مما يضمن قابلية التوسع والتكامل المستقبلي. كما تم تطوير مجموعة من الواجهات الرسومية في Flutter التي تغطي كافة جوانب النظام، بدءًا من تسجيل الدخول والمصادقة، وصولًا إلى عرض الجداول الزمنية، والتقارير الصحية، والوجبات اليومية، والوسائط المتعددة، والأنشطة التعليمية التفاعلية. نتج عن هذا العمل تطبيق متجاوب وسهل الاستخدام يوفر منصة مركزية للمعلومات، ويقلل من الاعتماد على الوسائل التقليدية للتواصل، ويعزز من الشفافية والمشاركة الفعالة لأولياء الأمور في التجربة التعليمية لأطفالهم، مما يساهم في تحسين جودة الرعاية والتعليم المقدم في مرحلة رياض الأطفال.

الكلمات المفتاحية: نظام إدارة، رياض الأطفال، Flutter، Laravel، واجهة برمجة تطبيقات (API)، تطبيقات محمولة، تواصل أولياء الأمور.

Project Abstract

This project involes the design and development of an integrated management system for kindergartens, aiming to enhance operational efficiency and improve communication mechanisms between kindergarten administration and parents. The system is built on a modern technical infrastructure, using the Flutter framework to develop an interactive and unified mobile front-end, while the back-end was developed using the Laravel framework based on PHP, providing a robust foundation for data and process management. The system is designed to serve three main roles: the parent, who can monitor their child's academic and health progress, view activities and events, and communicate directly with the administration; the child, around whom the system is centered through attendance records, health reports, and behavioral notes; and the administration, which has tools to manage children’s data, classrooms, announcements, and daily schedules.

The core contributions of the project include building a comprehensive and well-documented Application Programming Interface (API) based on OpenAPI (YAML) standards, ensuring scalability and future integration. Additionally, a complete set of graphical interfaces was developed using Flutter, covering all aspects of the system—from login and authentication to displaying schedules, health reports, daily meals, multimedia content, and interactive learning activities. The resulting product is a responsive and user-

friendly application that offers a centralized information platform, reduces reliance on traditional communication methods, and enhances transparency and active parental involvement in the educational experience of their children, thereby contributing to improved quality of care and education in early childhood settings.

Keywords: Management System, Kindergarten, Flutter, Laravel, Application Programming Interface (API), Mobile Applications, Parental Communication.

1

الفصل الأول: المقدمة

1-1 نطاق المشروع

تُعد رياض الأطفال من المؤسسات التربوية الحيوية في مرحلة تنشئة الطفل، حيث تُسهم في تنمية الجوانب الذهنية والحركية والسلوكية خلال سنواته المبكرة. وتواجه هذه المؤسسات تحديات كبيرة نتيجة الاعتماد على الأساليب التقليدية في الإدارة، مثل السجلات الورقية، الأمر الذي يؤدي إلى صعوبات في الوصول إلى المعلومات الدقيقة، وتأخير التواصل مع أولياء الأمور، فضلاً عن احتمالية فقدان البيانات، مما يقضي على كفاءة الأداء المؤسسي [1]. أضف إلى ذلك أن التوسع الملحوظ في الإقبال على رياض الأطفال زاد من تعقيد العمليات الإدارية، مما يُلزم بضرورة البحث عن حلول تقنية يمكنها تبسيط هذه المهام وتحسين جودة الخدمات المقدمة.

وقد أكدت الدراسات الأكاديمية مؤخرًا على أن التحول الرقمي في رياض الأطفال يمثل خطوة أساسية نحو تحسين الكفاءة التشغيلية وتيسير التواصل الفوري والشامل بين إدارة الروضة والأسر. وأشارت إحدى الأبحاث في مصر إلى تصور تطبيقي يمكن تنفيذه لتحويل البيئة الإدارية في هذه المؤسسات نحو الرقمنة [1]، كما أبرزت دراسات في السعودية ضرورة إدماج مفاهيم التقنية والمواطنة الرقمية ضمن مناهج رياض الأطفال لتعزيز قدرات المعلمات والأسرة على التعامل مع التكنولوجيا [2].

في ضوء ما تقدم، تتضح أهمية إنشاء نظام إدارة رقمي متكامل لرياض الأطفال، يجمع بين قدرات تتبع الحضور، التقارير الصحية والسلوكية، ونقل المعلومات بين الجهة الإدارية وأولياء الأمور في لحظة حدوثها. ومن هنا جاءت فكرة مشروع "نظام إدارة رياض الأطفال" الذي يهدف إلى بناء بيئة

تقنية متقدمة تجمع بين واجهة أمامية مرنة ومعتمدة على Flutter، وخلفية قوية مبنية باستخدام Laravel/PHP.

2-1 أهمية المشروع

في ظل غياب نظام رقمي متخصص لإدارة العمليات اليومية في رياض الأطفال، تظهر عدة إشكالات تحدّ من كفاءة الأداء وجودة التفاعل بين الأطراف المعنية. تعاني الإدارات من صعوبة إيصال التعميمات والمستجدات إلى أولياء الأمور بشكل سريع وفعّال، ويستلزم تتبع الحالة الصحية والغذائية للأطفال جهداً يدويًا قد يسفر عن ضياع بيانات أو تأخر في التحديث. كما يفتقر أولياء الأمور إلى وسيلة آمنة لمتابعة أنشطة أطفالهم اليومية، سواء عبر الصور والفيديوهات أو التقارير التربوية، مما يقلل من مشاركتهم الفاعلة في المسار التعليمي.

تتجلى أهمية إيجاد حل رقمي في قدرته على معالجة هذه التحديات من خلال توفير بيئة متكاملة، تمكّن إدارة الروضة من أداء مهامها بكفاءة عالية، وتمنح أولياء الأمور أدوات تفاعلية لمتابعة نمو أطفالهم في الجوانب الأكاديمية والصحية. كما أن النظام يعزز الشفافية ويعزز مستوى الثقة والتعاون بين الإدارة والأسر، ويعمل على تحسين جودة الرعاية والخدمات التربوية. هذا يتوافق مع نتائج دراسات تؤكد أن استخدام تقنيات التواصل الرقمي بين المعلمين وأولياء الأمور يُسهم بشكل ملحوظ في زيادة مشاركة الأسرة وتعزيز الشفافية [3].

أكثر من ذلك، يلمّ المشروع بتصميم أدوات تصنيف المحتوى التعليمي حسب الفئة العمرية والموضوع، ويزوّد بواجهات خاصة للفعاليات والرحلات تتيح تنظيم المشاركة وتوثيقها، مما يجعل

المشروع إضافة نوعية إلى المنظومة التعليمية لرياض الأطفال، ويعكس الحاجة الملحة إلى تبني الحلول الرقمية الذكية داخل المؤسسات التربوية [4].

3-1 أهداف المشروع

انطلاقًا من الحاجة الملحّة إلى نظام رقمي يُعنى بتنظيم شؤون رياض الأطفال ويسهم في تحسين جودة الخدمات التربوية والإدارية المقدمة، تم تحديد مجموعة من الأهداف التي يسعى المشروع إلى تحقيقها. وقد رُوعي في صياغة هذه الأهداف أن تكون واقعية، قابلة للتنفيذ، ومباشرة في تلبية احتياجات المستخدمين من مختلف الفئات: الإدارة، أولياء الأمور، وحتى الأطفال بشكل غير مباشر.

تتمثل الأهداف الأساسية للمشروع فيما يلي:

- تطوير نظام رقمي متكامل لإدارة رياض الأطفال باستخدام بنية تقنية حديثة تعتمد على إطار Flutter لتطوير واجهة أمامية متجاوبة، وإطار Laravel لبناء واجهة خلفية قوية لإدارة البيانات والمعالجة المنطقية.
- تصميم واجهات استخدام مخصصة لكل من أولياء الأمور، إدارة الروضة، والطفل (بشكل غير مباشر)، تتيح لكل فئة تنفيذ المهام الخاصة بها بفعالية وسهولة.
- بناء واجهة برمجة تطبيقات (API) موثقة وفق معايير OpenAPI (YAML)، بما يضمن قابلية التوسع والتكامل المستقبلي مع أنظمة أو تطبيقات أخرى.

- إتاحة التواصل الفوري بين الإدارة وأولياء الأمور من خلال إرسال الإشعارات والتعليمات والتحديثات حول الأنشطة والبرامج التعليمية.
- تنظيم سجلات الأطفال الصحية والتعليمية، مثل تتبع الحضور، وتسجيل الملاحظات السلوكية، ومواعيد التطعيم، ومتابعة النظام الغذائي اليومي.
- توفير إمكانية مشاركة الوسائط المتعددة (صور، فيديوهات) الخاصة بالأنشطة اليومية للأطفال في بيئة رقمية آمنة، تخضع لإشراف الإدارة.
- تصنيف المحتوى التعليمي والتربوي حسب الفئة العمرية والموضوع، مع توفير مكتبة فيديو تعليمية تفاعلية تعزز التعلم غير المباشر.
- إدارة الفعاليات والرحلات الإلكترونية من خلال قسم خاص يمكّن أولياء الأمور من الاطلاع على التفاصيل، التسجيل، والموافقة عبر النظام.

1-4 نطاق المشروع وحدوده

بناءً على الأهداف المحددة للمشروع، تم وضع نطاق دقيق يحدد ما سيتم تضمينه ضمن إطار العمل، لضمان تركيز الجهود على الجوانب الأساسية التي تحقق الفائدة المرجوة وتتناسب مع الموارد والمدة الزمنية المتاحة. يركّز نطاق المشروع على تطوير نظام رقمي متكامل لإدارة رياض الأطفال، يُنفّذ على هيئة تطبيق مخصص لأجهزة Android باستخدام إطار Flutter، مرتبط بواجهة خلفية تعتمد على Laravel لإدارة البيانات والعمليات. يشمل النظام خدمات أساسية تستهدف ثلاث فئات رئيسية: إدارة الروضة، أولياء الأمور، والأطفال بشكل غير مباشر، مع تصميم واجهات

استخدام مخصصة وصلاحيات محددة لكل فئة. كما يتضمن المشروع تطوير قاعدة بيانات مركزية تدار من خلال لوحة تحكم خاصة بالإدارة، تتيح متابعة الأنشطة اليومية، مشاركة الوسائط، عرض وجبات الطعام، تتبع الحالة الصحية للأطفال، وجدولة التطعيمات، بالإضافة إلى مكتبة مصنفة من الفيديوهات التعليمية حسب العمر والموضوع. يُمكّن النظام أولياء الأمور من إنشاء حسابات خاصة لمتابعة أطفالهم ضمن بيئة آمنة. في المقابل، لا يشمل نطاق المشروع في هذه المرحلة دعم أنظمة iOS، أو دمج تقنيات الذكاء الاصطناعي، أو أدوات تحليل البيانات، أو الدفع الإلكتروني، مع التركيز على اللغة العربية فقط. وسيتم تنفيذ التطبيق ضمن بيئة اختبارية محلية لضمان فعاليته قبل الإطلاق.

5-1 الفرضيات والتحديات

1-5-1 الفرضيات

- توفر بيئة تقنية تدعم تشغيل التطبيق على أجهزة Android.
- استعداد المستخدمين (الإدارة وأولياء الأمور) لتبنّي الحل الرقمي والتفاعل معه.
- قدرة إدارة الروضة على تزويد النظام بالبيانات بشكل منتظم (كجداول الطعام، الأنشطة، والتقارير الصحية).
- امتلاك فريق التطوير المعرفة الكافية بأدوات Flutter وLaravel، وقدرتهم على تنفيذ المتطلبات التقنية.

1-5-2 التحديات

- تحقيق توازن بين سهولة الاستخدام ومتطلبات الأمان، خصوصًا عند التعامل مع صور وفيديوهات الأطفال.
- صعوبات محتملة في ربط الواجهة الأمامية بالواجهة الخلفية وضمان سلاسة تدفق البيانات.
- تصميم قاعدة بيانات مرنة وقابلة للتوسع لاستيعاب تعدد الوحدات والوظائف.
- احتمالية ظهور أخطاء منطقية أو تضارب في الصلاحيات أثناء مرحلة الاختبار.
- محدودية الوقت والموارد البشرية، مما يتطلب إدارة دقيقة للوقت وتوزيع فعّال للمهام.

1-6 منهجية العمل والتنفيذ

تم تنفيذ المشروع وفق منهجية تعتمد على تقسيم العمل إلى مراحل متتابعة، تبدأ بـ مرحلة التخطيط الأولي التي تضمنت تحديد فكرة النظام وأهدافه العامة، وتحليل الإمكانيات التقنية والبشرية المتاحة. ثم جاءت مرحلة التحليل، حيث تم جمع متطلبات المستخدمين (الإدارة، أولياء الأمور) وتحديد الوظائف الأساسية التي يجب أن يوفرها النظام، مع تصميم قاعدة البيانات وهيكل الواجهات. بعد ذلك، نُفذت مرحلة التنفيذ، التي تم خلالها تطوير التطبيق باستخدام Flutter للواجهة الأمامية المخصصة لأجهزة Android، وربطه بواجهة خلفية تعتمد على Laravel. وأخيرًا، تم الانتقال إلى مرحلة الاختبار والتجريب، حيث جرى اختبار جميع وظائف النظام والتأكد من توافقها مع احتياجات المستخدمين، مع معالجة أي أخطاء أو مشاكل ظهرت أثناء التجربة لضمان الاستقرار وسهولة الاستخدام.

7-1 الهيكلية

يتألف هذا البحث من خمسة فصول رئيسية، تم تنظيمها بشكل منهجي لتغطية مختلف جوانب المشروع بدءًا من الخلفية النظرية ووصولًا إلى التنفيذ والتقييم:

1. **الفصل الأول: المقدمة**

يتناول هذا الفصل عرضًا عامًا لمشكلة المشروع وخلفيته، بالإضافة إلى تحديد أهدافه، نطاقه، الفرضيات والتحديات المتوقعة، ومنهجية التنفيذ، ويختتم بعرض هيكلية البحث المعتمدة.

2. **الفصل الثاني: الدراسات السابقة والإطار النظري**

يُستعرض في هذا الفصل تعريف أنظمة إدارة رياض الأطفال ووظائفها الأساسية، بالإضافة إلى تحليل أمثلة عالمية لأنظمة مشابهة مثل Brightwheel وKinderpedia وIllumine، مع تسليط الضوء على الفجوات والتحديات في الأنظمة الحالية. كما يتم عرض التقنيات المستخدمة في بناء النظام مثل Flutter وLaravel وBootstrap، وأسباب اختيارها.

3. **الفصل الثالث: تحليل النظام وتصميمه**

يُركّز هذا الفصل على تحديد المتطلبات الوظيفية وغير الوظيفية للنظام، وتحليل الأدوار المختلفة للمستخدمين، وتصميم حالات الاستخدام (Use Cases)، بالإضافة إلى عرض مخطط الكيانات والعلاقات (ERD) وتصميم قاعدة البيانات.

4. الفصل الرابع: تنفيذ النظام

يُعرض في هذا الفصل الجوانب العملية لتطوير النظام، بما يشمل الواجهات الرسومية الخاصة بلوحة تحكم الإدارة وتطبيق أولياء الأمور، بالإضافة إلى شرح لآليات العمل وتدفق البيانات ضمن النظام.

5. الفصل الخامس: الخاتمة والتطويرات المستقبلية

يتضمن هذا الفصل ملخصًا لأبرز النتائج المتحققة من المشروع، والتحديات التي تمت مواجهتها، إلى جانب طرح اقتراحات لتطوير النظام مستقبلًا بما يتلاءم مع احتياجات رياض الأطفال المتزايدة.

2

الفصل الثاني: الدراسات السابقة والمفاهيم النظرية

2-1 مقدمة الفصل

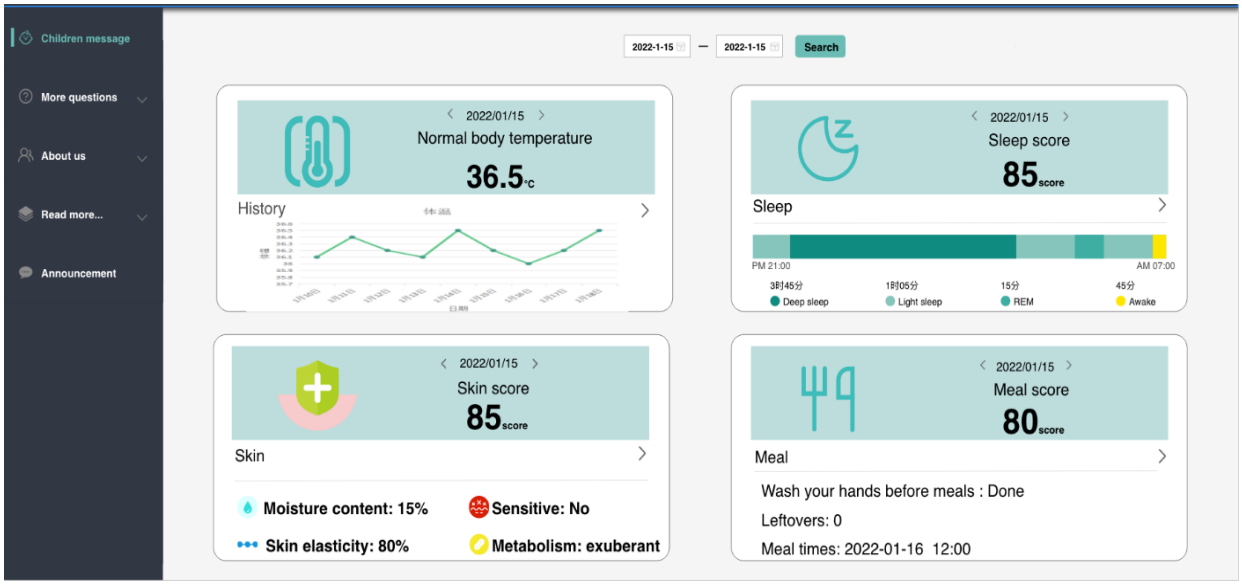
في هذا الفصل نستعرض الإطار النظري الذي يشكل الأساس لفهم وتصميم نظام إدارة رياض الأطفال الرقمي المقترح، حيث نتناول تعريف أنظمة إدارة رياض الأطفال ووظائفها، والتقنيات البرمجية المستخدمة في تطوير النظام مثل Flutter وLaravel، بالإضافة إلى الوظائف الرئيسية التي يقدمها النظام، وأهمية المحتوى التعليمي والأمن والخصوصية في تطبيقات الأطفال. كما نناقش الدراسات السابقة ذات الصلة لتسليط الضوء على الفجوات التي يهدف مشروعنا إلى معالجتها، مما يمهد الطريق للانتقال إلى المرحلة العملية في الفصول التالية.

2-2 أنظمة إدارة رياض الأطفال

2-2-1 تعريف أنظمة رياض الأطفال

أنظمة إدارة رياض الأطفال هي برمجيات متخصصة تهدف إلى تنظيم وتسهيل العمليات الإدارية والتربوية داخل مؤسسات التعليم المبكر. تعمل هذه الأنظمة على جمع وتنظيم البيانات المتعلقة بالأطفال، الموظفين، والبيئة التعليمية بشكل شامل ومنسق، مما يسهم في تحسين جودة الرعاية والتعليم المقدمين داخل الروضة.

يتمثل الهدف الأساسي من هذه الأنظمة في تعزيز كفاءة العمل الإداري، تسهيل التواصل بين روضة الأطفال وأولياء الأمور، وضمان متابعة دقيقة لتطور الأطفال من النواحي الصحية والتعليمية والسلوكية، مما يساعد في توفير بيئة تعليمية آمنة ومحفزة.



الشكل 1-2: مثال على أنظمة إدارة رياض الأطفال.

تشير الدراسات إلى أن اعتماد هذه الأنظمة الرقمية يعزز من جودة الخدمة ويقلل من الأخطاء الإدارية، مما يعود بالفائدة على جميع الأطراف المعنية في العملية التعليمية [5].

2-2-2 الوظائف العامة لأنظمة إدارة رياض الأطفال

تتضمن الوظائف العامة لأنظمة إدارة رياض الأطفال عدة مهام رئيسية تهدف إلى تحسين جودة الإدارة والخدمات المقدمة [5]، وفيما يلي تفصيل لكل وظيفة وأهميتها:

1. **تسجيل الأطفال:** تُعتبر أول خطوة في النظام، حيث يتم إدخال بيانات الطفل الأساسية مثل الاسم، تاريخ الميلاد، معلومات الوالدين، والحالة الصحية. يضمن التسجيل الدقيق سهولة الوصول إلى البيانات وتنظيمها لاحقًا في النظام، مما يسهل المتابعة والإدارة.

2. **متابعة الحضور والانصراف:** تتيح هذه الوظيفة تسجيل دخول الأطفال وخروجهم يوميًا بدقة، مع القدرة على رصد الغيابات والتأخيرات. تؤدي هذه الميزة إلى تحسين الأمان وضبط الحضور مما ينعكس إيجابًا على التخطيط اليومي للروضة.
3. **التواصل مع أولياء الأمور:** تشمل إرسال التنبيهات والإشعارات بشأن الأحداث اليومية، النشاطات، الحالة الصحية للأطفال، أو أي مستجدات أخرى. هذا التواصل يعزز الشفافية ويقوي العلاقة بين الروضة والأسرة، مما يرفع من رضا الأهالي ويساهم في دعم الطفل.
4. **إدارة النشاطات والفعاليات:** توفر هذه الوظيفة إمكانية تنظيم وتخطيط الأنشطة التعليمية والترفيهية، تحديد مواعيدها، وتسجيل مشاركة الأطفال فيها. يساعد هذا في ضمان تنوع الأنشطة وتحقيق أهداف التنمية الشاملة للأطفال.
5. **التقارير الصحية:** تسمح برصد وتوثيق الحالة الصحية للأطفال، مواعيد التطعيم، متابعة الأمراض أو الحساسيات. هذه الوظيفة ضرورية لضمان سلامة الأطفال واتخاذ الإجراءات المناسبة في الوقت المناسب.
6. **إدارة الحسابات المالية:** تشمل تسجيل الرسوم الدراسية، المدفوعات، والفواتير، مما يساعد في تنظيم الشؤون المالية للروضة بدقة وشفافية.

2-2-3 أمثلة لأنظمة إدارة رياض الأطفال

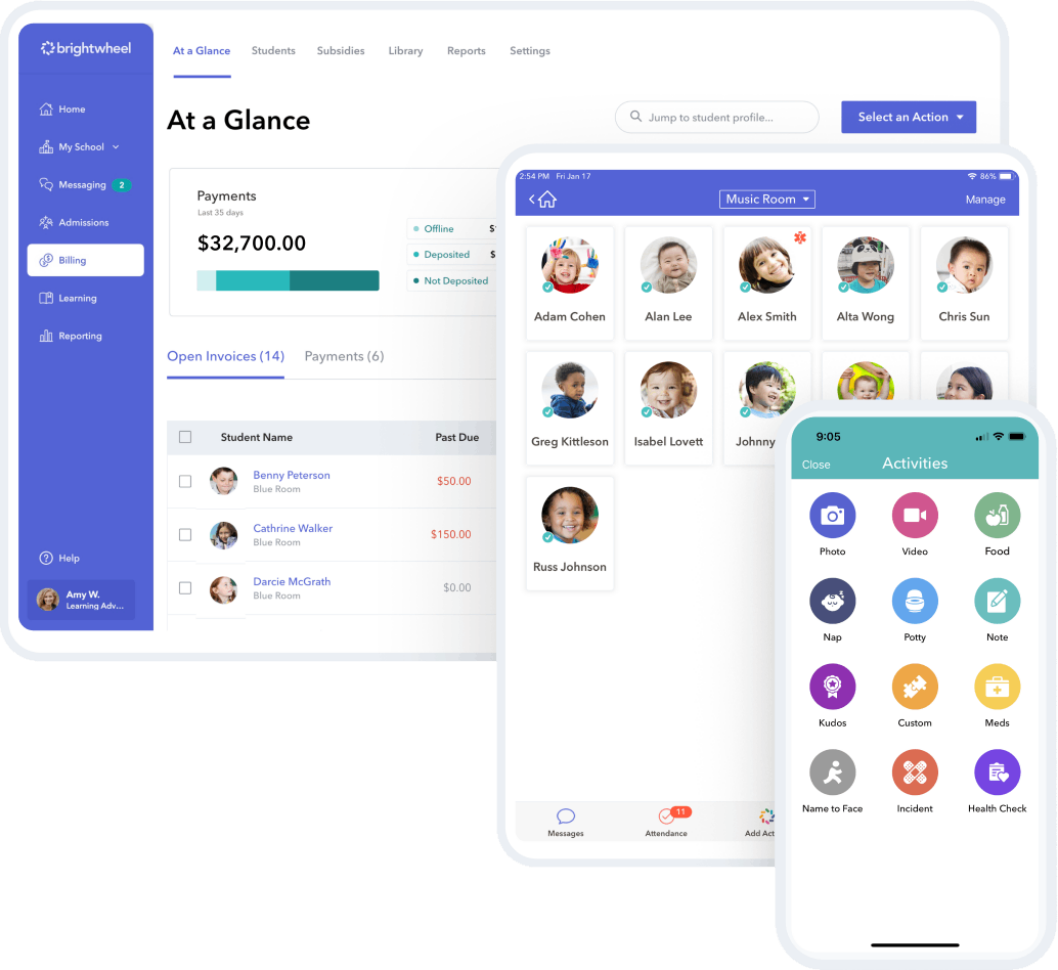
2-2-3-1 نظام Brightwheel

نظام Brightwheel يُعتبر من الأنظمة الرقمية الرائدة في مجال إدارة رياض الأطفال ومراكز التعليم المبكر، ويتميز بتكامله وسهولة استخدامه، مما يجعله خيارًا شائعًا بين المؤسسات التعليمية في الولايات المتحدة ودول أخرى. يهدف هذا النظام إلى تسهيل العمليات اليومية داخل الروضة، من خلال أدوات تقنية تتيح للمعلمين والإداريين تسجيل الحضور وتتبع سلوك الأطفال وإرسال تقارير يومية إلى أولياء الأمور، فضلًا عن إدارة الجداول الزمنية والمدفوعات والوثائق الصحية. ويُوفر Brightwheel بيئة تفاعلية تسهّل التواصل الفوري بين الروضة والأسرة، مما يعزز الثقة ويوفر شفافية في متابعة تقدم الأطفال [6].

الشكل 2-2: شعار نظام Brightwheel.

يعتمد Brightwheel على واجهات استخدام منظمه وعمليه، حيث تعد واجهه "Today Feed" من أبرزها، وهي تعرض تسلسلًا زمنيًا لنشاط الطفل خلال اليوم، متضمنة أوقات الوجبات، فترات النوم، الصور، الملاحظات، وغيرها، وهي واجهة أساسية يعتمد عليها أولياء الأمور للاطلاع على تفاصيل يوم طفلهم لحظة بلحظة. كذلك يضم النظام واجهات مهمة أخرى مثل لوحة الحضور

"Attendance Dashboard" التي تمكّن المعلمين من تسجيل الدخول والخروج للأطفال بسهولة، وواجهة "Billing" التي تُستخدم لإدارة الفواتير والدفع الإلكتروني.



الشكل 3-2: واجهة الموقع والتطبيق.

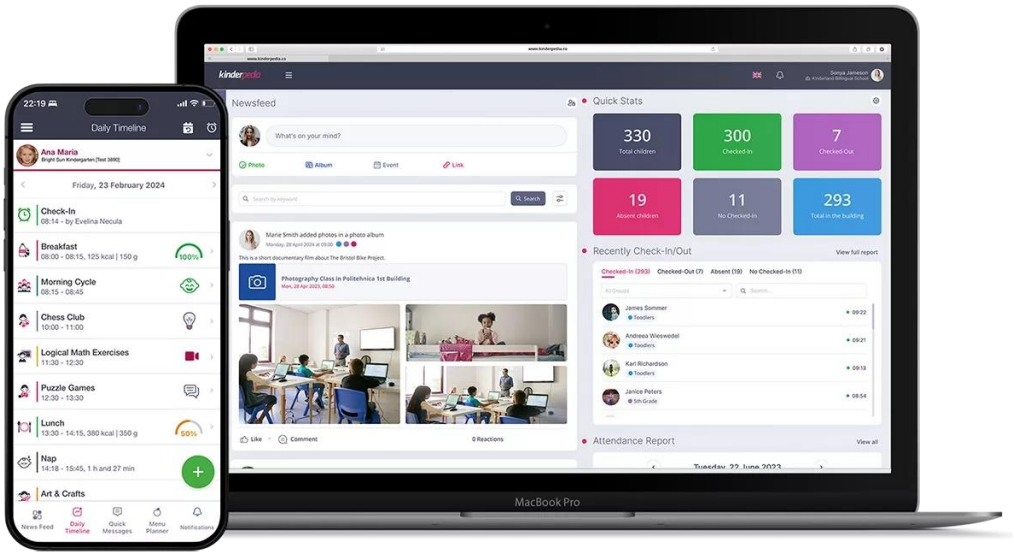
من نقاط قوة Brightwheel قدرته على دمج الوظائف التربوية والإدارية في تطبيق موحد، مع دعم كامل لتطبيقات الهواتف الذكية (iOS وAndroid)، مما يتيح الوصول إلى النظام من أي مكان وبشكل لحظي. كما يُوفر النظام تقارير دورية قابلة للتخصيص، وإشعارات تلقائية، وميزات أمنية متقدمة لحماية بيانات الأطفال. أما أبرز نقاط ضعفه، فتتمثل في أن بعض الوظائف مثل التحليلات المتقدمة والتخصيص الكامل متاحة فقط في النسخة المدفوعة، بالإضافة إلى غياب الدعم المباشر

للغة العربية، ما يجعل استخدامه محدودًا في بعض السياقات المحلية. ورغم هذه التحديات، يظل Brightwheel من أقوى الحلول التقنية المخصصة لإدارة بيئات التعليم المبكر بفعالية وكفاءة.

2-3-2-2 نظام Kinderpedia

نظام Kinderpedia يُعد من الحلول الرقمية المتقدمة في مجال إدارة مؤسسات الطفولة المبكرة والمدارس الابتدائية، وقد تم تطويره لتقديم تجربة تعليمية وإدارية شاملة تسهّل التواصل بين المعلمين والإداريين من جهة، وأولياء الأمور من جهة أخرى، مع التركيز على تبسيط العمليات اليومية داخل الروضة. يتيح هذا النظام أدوات متنوعة لإدارة جداول الأطفال، تتبع الحضور، مشاركة الموارد التعليمية، وتوفير قنوات تواصل فوري تُمكن أولياء الأمور من متابعة تطور أبنائهم بشكل منتظم. تعتمد Kinderpedia على بيئة سحابية تتيح الوصول إلى النظام من خلال تطبيقات الهاتف المحمول أو متصفح الإنترنت، ما يجعل التفاعل مع النظام مرنًا وسهلًا لجميع الأطراف.

تتميز واجهات النظام بتصميم واضح وتنظيم دقيق للوظائف، حيث تُعد واجهة "Dashboard" من أهمها، إذ تعرض ملخصًا فوريًا لكافة المعلومات المتعلقة بالصفوف، الأنشطة، الرسائل، والحضور، مما يساعد المعلمين والإدارة على اتخاذ قرارات سريعة ودقيقة. كما تتضمن واجهات أخرى مهمة مثل "Daily Reports" التي تسجل الملاحظات اليومية المتعلقة بتغذية الطفل، نومه، مزاجه، ومشاركته في الأنشطة، وترسل هذه البيانات تلقائيًا إلى الوالدين، مما يعزز عنصر الشفافية. بالإضافة إلى ذلك، يحتوي النظام على واجهة خاصة بـ "Lesson Planning" والتي تُستخدم لتخطيط الدروس وتوزيع المحتوى التعليمي بشكل منظم حسب المراحل العمرية.



الشكل 4-2: نظام إدارة حضور ونشاطات الأطفال في الحضانة والمدرسة باستخدام تطبيق رقمي متكامل.

من أبرز مزايا Kinderpedia أنه يدمج بين الجوانب الإدارية والتعليمية في واجهة واحدة، ويوفر إمكانية الترجمة للغات متعددة، من ضمنها اللغة العربية، مما يجعله مناسبًا أكثر من غيره في السياقات التعليمية في العالم العربي. كما يدعم النظام أدوات التحليل التربوي ومراقبة تقدم الأطفال من خلال الرسوم البيانية والتقارير المخصصة، إضافة إلى إمكانيات الدفع الإلكتروني وتتبع الفواتير. أما التحديات التي قد تواجه المستخدمين فتتمثل في أن بعض الواجهات قد تتطلب تدريبًا أوليًا لفهم آليات استخدامها بكفاءة، خاصة للمعلمين غير المتمرسين تقنيًا، فضلًا عن أن بعض الميزات تكون مدفوعة في الخطط المتقدمة. ومع ذلك، فإن Kinderpedia يقدم نموذجًا متطورًا في دعم التحول الرقمي لرياض الأطفال، ويساهم بشكل فعّال في رفع جودة الخدمات التعليمية والإدارية المقدمة [7].

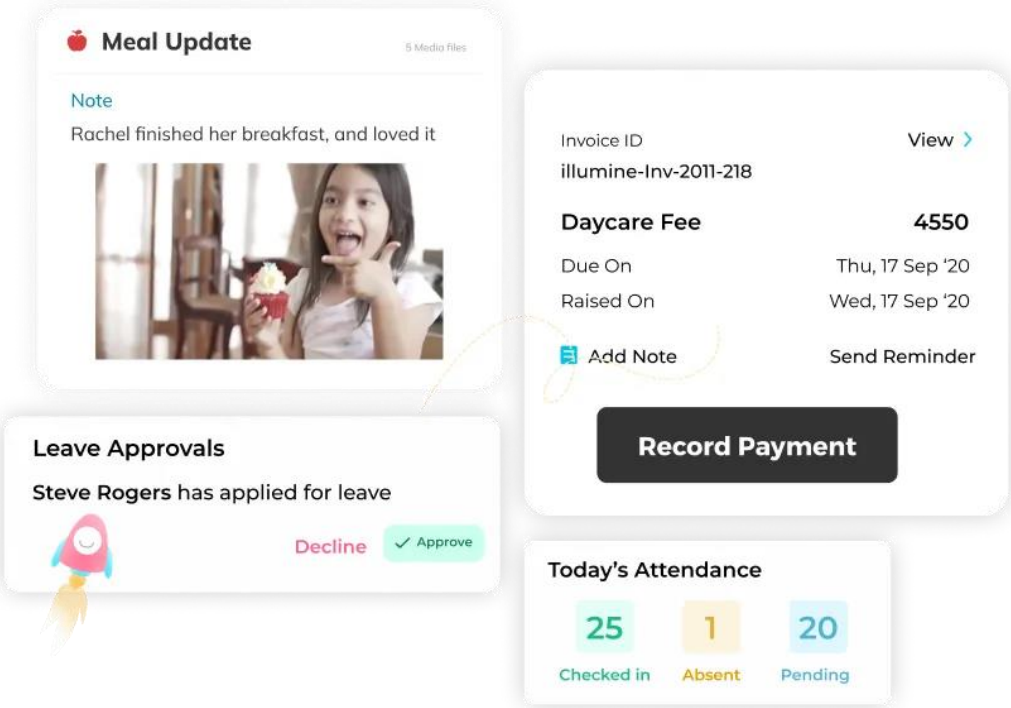
2-2-3-3 تطبيق Illumine

يُعد تطبيق Illumine من التطبيقات الحديثة المتخصصة في إدارة الحضانات ورياض الأطفال، ويقدم حلاً رقميًا متكاملاً يدعم مختلف الأنشطة اليومية داخل المؤسسة التعليمية المبكرة. يعتمد التطبيق على تبسيط العمل الإداري وتعزيز التواصل مع أولياء الأمور من خلال واجهات ذكية تدعم مشاركة المعلومات في الوقت الفعلي، بما في ذلك تقارير النشاط اليومي، صور الأطفال أثناء التعلم واللعب، وحالة النوم والطعام والصحة. كما يوفر Illumine نظامًا لتتبع الحضور والانصراف، وجدولة النشاطات، وإرسال الإشعارات الفورية، مما يساهم في تحسين الكفاءة التشغيلية وتقليل العبء الورقي على الطاقم التربوي.

الشكل 5-2: شعار نظام Illumine.

واحدة من ابرز ميزات هذا التطبيق هي قابليته للتخصيص ليتلاءم مع احتياجات كل مؤسسة على حدة، سواء من حيث تخطيط الفصول الدراسية أو إدارة الموظفين أو إعداد التقارير الدورية، ويشمل ذلك نظامًا خاصًا لإدارة الفواتير والمدفوعات الرقمية بما يعزز الاستدامة المالية للروضة. كما يتميز Illumine بقدرته على دعم المحتوى متعدد الوسائط، مثل الفيديوهات والصور، والتي تُستخدم لاطلاع الأهل على تقدم الطفل ومشاركته في الأنشطة التعليمية. أضف إلى ذلك أن

التطبيق يدعم الوصول من خلال الهاتف المحمول (iOS وAndroid) أو عبر الإنترنت، مما يتيح مرونة في الاستخدام من قبل مختلف الأطراف.



الشكل 6-2: واجهة نظام Illumine: تتبع الحضور، تحديثات الوجبات، الموافقات على الإجازات، والمدفوعات المالية.

من الناحية التقنية، يعتمد Illumine على بنية سحابية آمنة تضمن حماية بيانات الأطفال والموظفين، كما يتيح تصنيف المستخدمين وفقًا لصلاحيات محددة، مما يضمن خصوصية الوصول للمعلومات. ومع أن التطبيق يُصنف من بين الحلول الفعالة في السوق، إلا أن بعض التحديات قد تتعلق بتعقيد الواجهة الأولية بالنسبة للمستخدمين الجدد أو المؤسسات التي تنتقل لأول مرة من إدارة تقليدية إلى رقمية. ورغم ذلك، فإن التطوير المستمر لهذا التطبيق يجعله من بين الخيارات الواعدة التي تستهدف رفع جودة التعليم المبكر من خلال حلول ذكية وشاملة [8].

4-2-2 الفجوات والتحديات في الأنظمة الحالية

رغم التقدم الكبير الذي أظهرته الأنظمة العالمية مثل Brightwheel وKinderpedia وIllumine في إدارة رياض الأطفال، إلا أن هناك بعض النقاط التي ما زالت تحتاج إلى تحسين. على سبيل المثال، لا تسمح بعض هذه الأنظمة بتخصيص الخدمات بما يناسب احتياجات رياض الأطفال في كل بلد، خاصة في الدول الناطقة بالعربية. كما أن بعض التطبيقات لا توفر محتوى تعليميًا تفاعليًا كافيًا، أو يكون من الصعب استخدامه من قبل بعض أولياء الأمور أو العاملين في الروضة بسبب التصميم المعقد للواجهات.

بالإضافة إلى ذلك، لا تتيح بعض هذه الأنظمة ربطًا مرنًا بين الواجهة التي يراها المستخدم (مثل تطبيق الهاتف) والنظام الداخلي الذي يدير البيانات، مما يجعل عملية التحديث أو التوسع صعبة في بعض الحالات. كما أن الأنظمة الحالية غالبًا ما تركز على الوظائف الإدارية أكثر من تركيزها على دعم العملية التعليمية نفسها.

انطلاقًا من هذه الفجوات، جاء مشروعنا ليعالج هذه التحديات من خلال تصميم نظام شامل وسهل الاستخدام يخدم إدارة الروضة، أولياء الأمور، والأطفال. استخدمنا أدوات حديثة مثل Flutter لتوفير تطبيق متجاوب يعمل بسلاسة على الهواتف، وLaravel لبناء خلفية قوية تدير البيانات والوظائف المختلفة. كما أن النظام يدعم اللغة العربية بالكامل ويعكس احتياجات رياض الأطفال في بيئتنا المحلية، وهو ما يميّزه عن الأنظمة المستوردة.

ومع ذلك، ما يزال بالإمكان تطوير النظام ليصبح أقوى، مثل إضافة تقنيات ذكية لتحليل سلوك الطفل، أو ربطه بخدمات إضافية مثل الدفع الإلكتروني أو أنظمة الصحة المدرسية، وأيضًا تصميم تقارير ذكية تساعد الإدارة على تحسين أدائها اعتمادًا على بيانات دقيقة.

2-3 التقنيات المستخدمة في بناء النظام

2-3-1 إطار عمل Flutter

تم تصميم Flutter ليكون إطار عمل متكامل يتيح تطوير تطبيقات أصلية تعمل على أنظمة تشغيل متعددة من خلال قاعدة شيفرة واحدة. ويُعد هذا الإطار أداة فعالة لإنشاء تطبيقات متوافقة مع Android وiOS وواجهات الويب وسطح المكتب بدون الحاجة إلى إعادة كتابة الأكواد لكل نظام بشكل منفصل، ما يسهم في توفير الوقت والموارد [9].

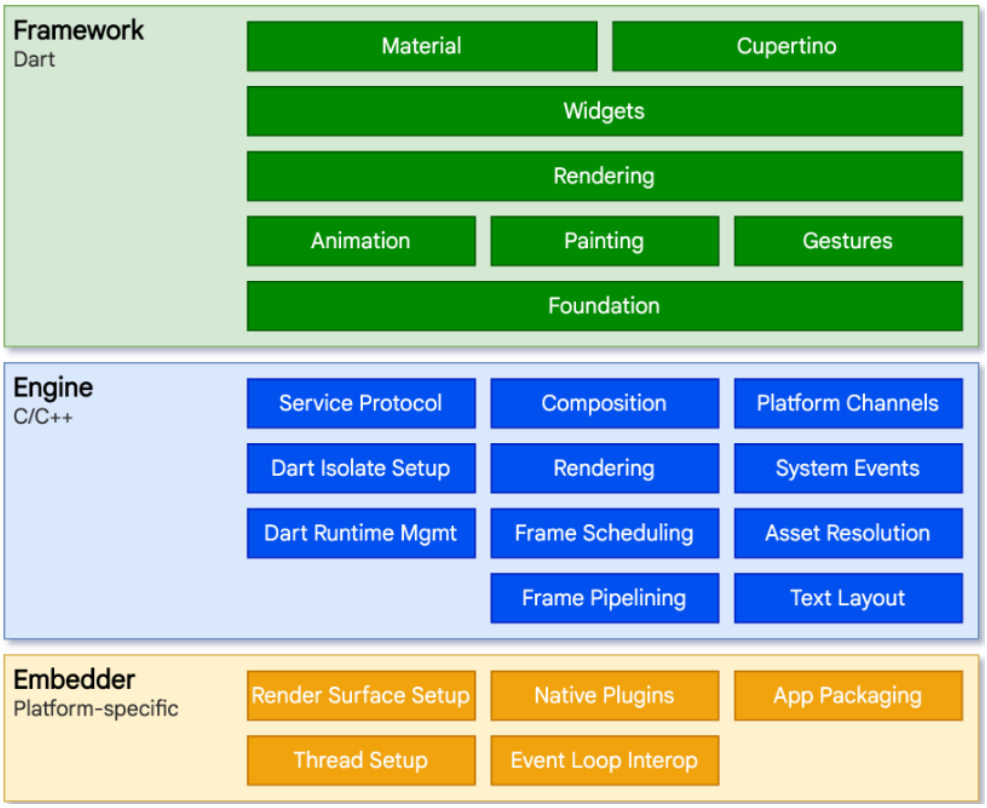
الشكل 2-7: شعار إطار العمل Flutter.

تعتمد Flutter على لغة البرمجة Dart، التي طوّرتها Google خصيصًا لتعزيز تجربة تطوير الواجهات المتعددة المنصات. وتتميز Dart بأنها مصممة لتكون لغة كائنية بسيطة ومرنة، مع أداء عالٍ متفوق، إذ تعتمد على تجميع التعليمات البرمجية إلى كود آلة مباشر (AOT) وبالتالي تدعم عرضًا سريعًا وموثوقًا لعناصر الشاشة. كما تتسم Dart بكونها لغة مفتوحة المصدر وتدعم ميزات

مثل الأمان من الأخطاء المتعلقة بالقيم الفارغة (null safety)، وهو ما يعزز من استقرار التطبيقات ويقلل من زمن اختبارها [9].

الشكل 8-2: شعار لغة البرمجة Dart.

يُعتبر اعتماد Flutter على بنية واضحة تتألف من ثلاث مكونات رئيسية—إطار العمل المكتوب بلغة Dart، ومحرك العرض (Flutter Engine)، والمكوّن الذي يدمجه في نظام التشغيل—عاملًا أساسيًا في تقديم واجهات مستخدم متجانسة وأداء ثابت عبر الأنظمة. وتعتمد Flutter نموذج "الواجهة التصريحية" في التصميم، مما يلزم المطورين بتحديد النتيجة النهائية المرئية أولًا ثم بناء العناصر داخليًا بناءً عليها، وهو ما يعزز وضوح الكود وتنظيمه. ويعد هذا النهج سببًا في انتشار Flutter واستخدامها في مشاريع بارزة مثل Google Pay وGoogle Classroom، بالإضافة إلى اعتمادها من شركات كبرى في التجارة الإلكترونية مثل Alibaba وeBay. وعلى المستوى المجتمعي، فإن Flutter تحظى بشهرة واسعة لكل من مطوري Android وiOS، وقد شهدت مؤخرًا نموًا كبيرًا في الاستقبال والدعم عبر GitHub وStack Overflow، وهو ما يؤكد قدرتها على تقديم بيئة تطويرية متكاملة وموثوقة.



الشكل 9-2: الطبقات المعمارية لإطار العمل Flutter.

2-3-1-1 مميزات Flutter وأسباب اختياره

- **دعم واجهات المستخدم الوصفية (Declarative UI):** يسمح Flutter ببناء واجهات مستخدم ديناميكية وسلسة باستخدام أسلوب وصفي، مما يسهّل عملية إنشاء الواجهات المعقدة مع المحافظة على وضوح الكود وتنظيمه.
- **الاستقلالية عن المنصة (Platform Independence):** يتميز Flutter بعدم اعتماده على مكونات واجهة المستخدم الخاصة بالنظام (native widgets)، إذ يستخدم محركه الرسومي الخاص لرسم كل العناصر، مما يضمن توحيد التجربة البصرية عبر مختلف أنظمة التشغيل.

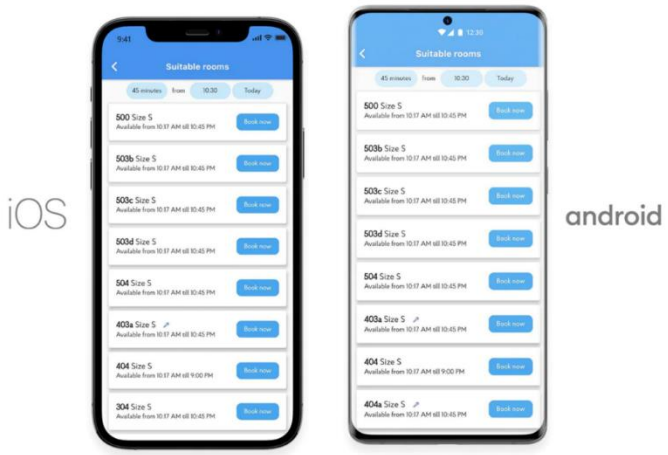
- **أداء قريب من الأداء الأصلي (Native-like Performance):** يُعزى ذلك إلى اعتماد Flutter على محرك رسوميات عالي الكفاءة مكتوب بلغة ++C، ما يُمكن التطبيقات من العمل بسرعة واستجابة عالية على مختلف الأجهزة.
- **هيكلية مرنة تعتمد على الطبقات (Layered Architecture):** تُقسم البنية المعمارية في Flutter إلى طبقات متمايزة مثل إطار العمل، المحرك، والطبقة المدمجة (Embedder)، مما يسهّل تخصيص كل جزء على حدة وتحسينه دون التأثير على الأجزاء الأخرى.
- **تطوير واجهات رسومية مخصصة بالكامل (Custom Rendering):** تتيح البنية الرسومية لمحرك Flutter إمكانية تصميم مكونات واجهة مستخدم متقدمة وغير تقليدية دون الاعتماد على أدوات النظام.
- **تكامل سهل مع النظام المضيف (Host Integration):** يدعم Flutter التكامل السلس مع الأكواد الأصلية (native code) مثل Java أو Kotlin للأندرويد وSwift أو Objective-C لنظام iOS، مما يوفّر مرونة إضافية للمطورين.
- **توفر أدوات تطوير حديثة (Modern Dev Tools):** يتضمن Flutter أدوات قوية مثل Hot Reload التي تتيح للمطورين رؤية التغييرات مباشرة أثناء التطوير، مما يزيد من الإنتاجية ويُسرّع من عملية التجريب والتعديل.
- **مجتمع نشط ودعم مستمر من Google:** يحظى Flutter بدعم تقني متواصل وتحديثات دورية من شركة Google، إلى جانب مجتمع عالمي يوفر موارد تعليمية ومكتبات مفتوحة المصدر.

2-1-3-2 تطبيقات عملية على استخدام إطار العمل Flutter

رغم حداثة ظهور Flutter نسبيًا، إلا أنه أصبح معتمدًا من قبل العديد من المؤسسات والخدمات العالمية البارزة، مثل Google Ads، Alibaba، AppTree، وReflectly، إلى جانب استخدامه من قِبل Nubank، التي تُعد من أكبر البنوك الرقمية في العالم خارج آسيا. ويُعد اختيار Nubank لإطار Flutter مثالًا واضحًا على الفوائد التي يقدمها تطوير التطبيقات الموحدة من حيث الكفاءة وتوفير الوقت، خصوصًا في المؤسسات التي تتعامل مع قاعدة مستخدمين واسعة تتجاوز 20 مليون مستخدم.

أما فيما يخص نطاق استخدامات Flutter، فإنه يُعد مناسبًا لتطوير جميع أنواع تطبيقات الهاتف المحمول دون استثناء، سواء كانت موجهة للشركات الناشئة أو الكبرى. يمكن استثماره في تطوير تطبيقات لمتاجر إلكترونية، وبنوك، ومراكز اتصال، وخدمات التوصيل، وإدارة العمليات الداخلية في المؤسسات. وتبرز قوته بشكل خاص في المشاريع البحثية ومبادرات تطوير النماذج الأولية (R&D)، نظرًا لقدراته على تسريع اختبار الأفكار التجارية وإطلاقها بسرعة.

ومن التجارب التطبيقية التي تؤكد كفاءة Flutter في البيئات المهنية، تجربة فريق شركة NIX في تطوير تطبيق لإدارة حجوزات غرف الاجتماعات. حيث طلب عميل كبير يعمل في قطاع التأمين ويضم حوالي 2000 موظف، تطوير نسخة محمولة من نظامه لإدارة المساحات المكتبية، وذلك كبديل محدث عن النسخة الويب القديمة، وبأقل تكلفة ممكنة.



الشكل 10-2: تطبيق حجز غرف الاجتماعات - شركةNIX .

فاقترح الفريق حلًا عبر Flutter لتقديم تطبيق موحّد يعمل على كل من نظامي iOS وAndroid. وبعد مرور شهرين فقط من بدء التطوير، استلم العميل تطبيقًا متعدد الوظائف يتيح جدولة الاجتماعات وحجز المساحات التشاركية بسهولة وكفاءة كما يظهر في الشكل (2-10)، مما يعكس الإمكانات العالية التي يوفّرها إطار Flutter لتطوير تطبيقات عالية الجودة في فترة زمنية قصيرة [10].

2-3-2 إطار العمل Laravel وPHP لتطوير الواجهات الخلفية

يُعد Laravel إطار عمل حديث بلغة PHP، وهو مخصص لتسهيل بناء مواقع وتطبيقات الويب. تم تصميمه ليساعد المطورين على كتابة التعليمات البرمجية بطريقة منظمة وواضحة، مما يجعل العمل على المشاريع الكبيرة أكثر سهولة وكفاءة. يعتمد Laravel على أسلوب يُسمى MVC (وهو اختصار لـ Model-View-Controller)، والذي يعني تقسيم المشروع إلى ثلاثة أجزاء رئيسية: الأول

يهتم بالبيانات، الثاني يهتم بعرض المحتوى للمستخدم، والثالث ينظم العلاقة بينهما. هذا الأسلوب يساعد في الحفاظ على الترتيب وفصل المهام داخل المشروع [11].

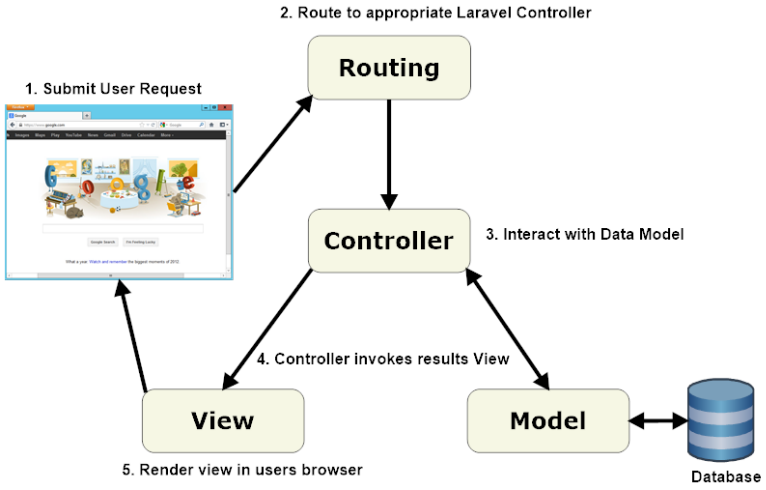
من أهم مميزات Laravel أنه يُوفر أدوات جاهزة تساعد على تنفيذ كثير من المهام الشائعة بسهولة، مثل إنشاء الصفحات، التعامل مع قواعد البيانات، وتنظيم تنقل المستخدم داخل الموقع. إحدى هذه الأدوات تُسمى "Eloquent"، وهي تجعل التعامل مع قواعد البيانات أسرع وأسهل، حيث يمكن للمطور أن يكتب أوامر بسيطة بدلًا من استخدام الاستعلامات الطويلة والمعقدة.

كذلك يحتوي Laravel على أداة تُدعى Artisan، وهي تشبه المساعد الذي يمكن للمطور أن يُعطيه أوامر بسيطة لينفذ مهام متكررة مثل إنشاء الملفات أو ترتيب الجداول في قاعدة البيانات. هذا يقلل من الوقت والجهد المطلوبين أثناء تطوير الموقع.

الشكل 11-2: شعار إطار العمل Laravel.

يُعد Laravel خيارًا ممتازًا لأنه لا يحتاج إلى الكثير من الإعدادات المعقدة، ويُتيح بناء مواقع مرنة وقابلة للتطوير مستقبلاً، ما يجعله مناسبًا لمشاريع الشركات الصغيرة والكبيرة على حد سواء [11].

2-3-2-1 مفهوم MVC في إطار العمل Laravel



الشكل 12-2: بنية MVC.

1. Model: إدارة البيانات والمنطق الخاص بالأعمال

يمثل Model في Laravel البنية التي تدير البيانات والمنطق الخاص بالتطبيق. يتولى هذا المكون مسؤولية التعامل مع قاعدة البيانات، حيث يعمل كجسر بين النظام وقاعدة البيانات من خلال توفير واجهة برمجية مبنية على البرمجة الكائنية لتيسير العمليات المتعلقة بالبيانات [12].

يُسهل Eloquent ORM المدمج في Laravel عمليات التفاعل مع قواعد البيانات داخل Model، مما يمكّن من تنفيذ إجراءات مثل الإضافة، الاستعلام، التعديل، والحذف على السجلات بطريقة سلسة ومنظمة. كما يسمح Eloquent بتحديد العلاقات بين الكيانات المختلفة، مما يعزز من قدرة النظام على إدارة البيانات المرتبطة والمعقدة بكفاءة.

2. View: عرض واجهات المستخدم

تمثّل الـ View واجهة المستخدم (UI) في التطبيق، حيث تتولى مهمة عرض البيانات للمستخدم واستقبال مدخلاته. في Laravel، يتم كتابة Views عادةً باستخدام محرك القوالب Blade، الذي يوفر لغة بسيطة وواضحة لإنشاء مكونات واجهة مستخدم ديناميكية وقابلة لإعادة الاستخدام.

تُخزّن Views في ملفات منفصلة تحتوي على علامات HTML مع دمج أكواد PHP. تستقبل البيانات من الـ Controller وتعرضها باستخدام قواعد لغة Blade. كما تدعم Views التحكم الشرطي، الحلقات، والأجزاء الجزئية (partials)، مما يسهل بناء واجهات مستخدم منظمة وتفاعلية [12].

3. Controller: معالجة طلبات المستخدم

يقوم الـ Controller بدور الوسيط بين الـ Model والـ View. حيث يستقبل طلبات المستخدم ويعالجها، ويتفاعل مع الـ Model لاسترجاع البيانات أو تعديلها، ثم ينقل البيانات المعالجة إلى الـ View المناسب لعرضها. في Laravel، يكون الـ Controllers مسؤولين عن تعريف الدوال أو الإجراءات التي تتوافق مع مختلف طلبات المستخدم. هذه الإجراءات تحتوي على المنطق اللازم للتعامل مع الطلبات المحددة وإدارة تدفق البيانات. كما يُسهّل الـ Controllers عملية التحقق من صحة بيانات المستخدم ويجري الفحوصات اللازمة للتفويض قبل التفاعل مع الـ Model [12].

سير عمل MVC في Laravel

- **طلب المستخدم:** عندما يستخدم الشخص التطبيق، يرسل جهازه طلبًا إلى الخادم (السيرفر).

- **التوجيه:** يقوم نظام Laravel بتوجيه هذا الطلب إلى المكان الصحيح داخل التطبيق، حسب العنوان الذي تم طلبه.
- **الـ Controller:** هذا الجزء يستقبل الطلب ويبدأ في التعامل معه، مثل التحقق من المعلومات المطلوبة أو تعديلها، ويتواصل مع قاعدة البيانات للحصول على البيانات أو تحديثها.
- **الـ Model:** هو المسؤول عن التعامل مع قاعدة البيانات، سواء لجلب المعلومات أو حفظ التغييرات بناءً على ما طلبه الـ Controller.
- **تحضير البيانات:** يقوم الـ Controller بتحضير البيانات التي يحتاجها المستخدم ويرسلها إلى جزء العرض.
- **الـ View:** هو الذي يعرض البيانات بشكل منظم وجميل على شكل صفحات ويب، باستخدام أدوات خاصة تجعل العرض مناسبًا وسهل القراءة.
- **استجابة المستخدم:** الصفحة النهائية تظهر على جهاز المستخدم مع جميع العناصر اللازمة، مثل النصوص والصور، لتنتهي بذلك عملية الطلب والاستجابة.

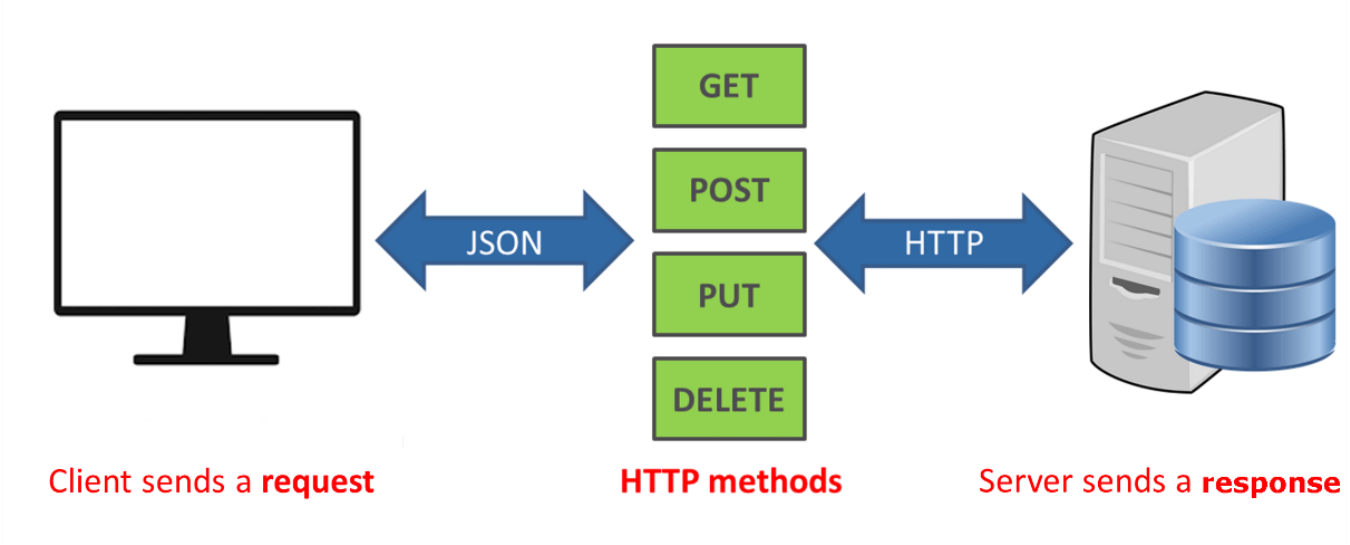
فوائد استخدام MVC في Laravel

- **فصل المهام:** يعزز نمط MVC تقسيم التطبيق إلى أجزاء واضحة: منطق التطبيق، واجهة العرض، وإدارة البيانات. هذا الفصل يسهل تنظيم الكود، صيانته، وإعادة استخدامه.

- **إعادة استخدام الكود:** يمكن إعادة استخدام أجزاء من الواجهات (Views) والمتحكمات (Controllers) في أماكن مختلفة من التطبيق، مما يقلل من تكرار الكود ويزيد من كفاءة التطوير.
- **التعاون وقابلية التوسع:** يسمح الفصل الواضح للمهام في MVC لعدة مطورين بالعمل على مكونات مختلفة في نفس الوقت، مما يسهل توسيع التطبيق وصيانته بشكل أفضل.

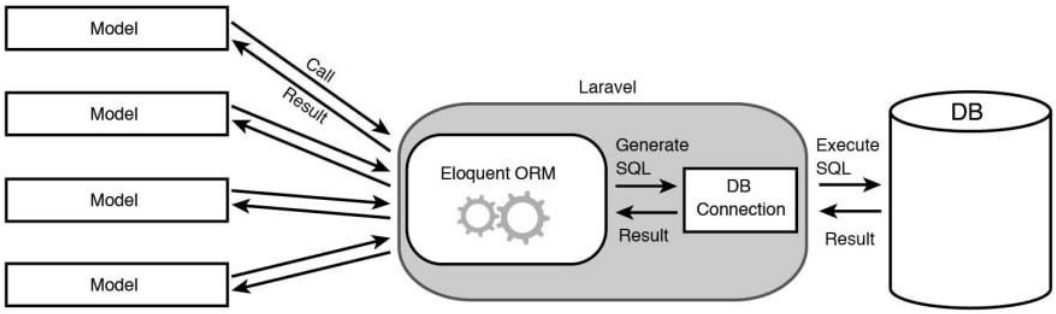
2-2-3-2 مميزات Laravel في بناء API والإدارة

يتميز Laravel كإطار عمل حديث لتطوير تطبيقات الويب بقدرات قوية في بناء واجهات برمجة التطبيقات (APIs) وأنظمة الإدارة، مما يجعله خيارًا مفضلاً بين المطورين. من أهم مميزاته دعمه الكامل لتصميم RESTful API، حيث يوفر أدوات وطرقًا مبسطة لبناء خدمات ويب تتبع معايير REST، مما يسهل عملية التواصل بين الخادم والعميل بشكل فعال ومنظم [11].



الشكل 2-13: RESTful API.

بالإضافة إلى ذلك، يسهل Laravel إدارة قواعد البيانات باستخدام نظام Eloquent ORM الذي يتيح للمطورين التعامل مع البيانات بأسلوب كائني ومنظم دون الحاجة إلى كتابة استعلامات SQL معقدة، حيث يُسهل هذا النظام عمليات الإدخال والتحديث والحذف والاستعلام عبر نماذج مبسطة تربط بين الجداول والبيانات بطريقة مباشرة وفعالة.



الشكل 14-2: Eloquent ORM.

من ناحية الأمان، يوفر Laravel دعمًا متقدمًا للتحكم في الصلاحيات وإدارة الوصول، من خلال نظام مصادقة متكامل (Authentication) وتفويض دقيق (Authorization) يسمح بتحديد مستويات مختلفة من الوصول للمستخدمين حسب أدوارهم، مما يعزز من حماية التطبيق ويمنع الوصول غير المصرح به إلى الموارد الحساسة أو وظائف النظام. هذه الميزات مجتمعة تجعل Laravel بيئة متكاملة لبناء أنظمة إدارة متطورة وآمنة وقابلة للتوسع بسهولة [11].

2-3-2-3 دور PHP في تطوير الخادم (Backend)

تُعدّ لغة PHP من أبرز لغات البرمجة المستخدمة في تطوير جانب الخادم (Backend) في تطبيقات الويب الحديثة، حيث تلعب دورًا محوريًا في تنظيم العمليات الخلفية وتسهيل التواصل بين الواجهة

الأمامية (Front-End) وقواعد البيانات. ومن أهم ما يميز PHP في هذا السياق أنها توفّر بيئة تشغيل مستقرة وموثوقة تُمكّن المطوّرين من إنشاء تطبيقات ذات أداء فعّال واستجابة عالية، حتى في ظل الضغط أو التوسّع في عدد المستخدمين [11].

الشكل 15-2: شعار لغة البرمجة PHP.

تعتمد PHP على بنية مرنة تجعلها قابلة للتكامل بسهولة مع العديد من الخوادم مثل Apache وNginx، مما يسمح بتشغيل التطبيقات على مختلف البيئات دون الحاجة إلى إعدادات معقّدة. كما أن توافقها مع عدد كبير من أنظمة التشغيل، بما في ذلك Linux وWindows، يوفّر للمبرمجين حرية اختيار بيئة العمل المناسبة لهم، وهو ما ينعكس إيجابًا على استقرار وأداء النظام.

الشكل 16-2: شعار Apache HTTP Server .

ومن جهة أخرى، تتميز PHP بسهولة التكامل مع قواعد البيانات، لا سيما قواعد البيانات العلائقية مثل MySQL وPostgreSQL، وذلك عبر واجهات برمجية موحدة مثل PDO (PHP Data Objects)، التي تتيح تنفيذ الاستعلامات ومعالجة البيانات بطريقة آمنة وفعّالة. هذا التكامل السلس يُعد من العوامل الأساسية في بناء أنظمة إدارة المحتوى، وتطبيقات التجارة الإلكترونية، وأنظمة التسجيل والمعالجة التي تعتمد بشكل كبير على إدارة البيانات.

كما تدعم PHP التكامل مع خدمات الويب (Web Services) عبر بروتوكولات متعددة، مثل REST وSOAP، مما يسمح بإنشاء واجهات برمجية (APIs) تُسهّل تبادل البيانات مع أنظمة أو تطبيقات أخرى. ويمكّن هذا الدعم مطوري الأنظمة من توسيع وظائف التطبيقات أو ربطها بمنصات خارجية بسهولة، وهو أمر بالغ الأهمية في تطوير تطبيقات حديثة مرنة قابلة للتوسّع.

3-3-2 إطار العمل Bootstrap

يُعتبر Bootstrap واحداً من الأدوات الرائدة في مجال تطوير واجهات الويب، حيث يقدم حلولاً شاملة تسهم في تسريع عملية بناء التصميمات الرقمية بطريقة احترافية. تم تطوير هذا الإطار في البداية كمشروع داخلي داخل شركة Twitter، ثم تحول إلى أداة مفتوحة المصدر تدعم المطورين حول العالم في إنشاء مواقع وتطبيقات ذات تصميم جذاب ومتناسق. يعتمد Bootstrap على مجموعة من المبادئ التي تجعله أداة مرنة وقابلة للتكيف مع مختلف الاحتياجات، مما يجعله خياراً مثالياً للمشاريع التي تتطلب سرعة في التنفيذ وجودة في النتائج [13].

الشكل 17-2: شعار إطار العمل Bootstrap.

من بين الجوانب البارزة في Bootstrap هو تركيزه على تقديم حلول قابلة للتخصيص بشكل كبير. يتم ذلك من خلال توفير أدوات تتيح للمطورين تعديل الأنماط والألوان والخطوط بما يتناسب مع طبيعة المشروع الفردي. هذه المرونة تجعل من السهل تحقيق هوية بصرية فريدة لكل موقع أو تطبيق، حتى مع استخدام نفس الإطار الأساسي. بالإضافة إلى ذلك، فإن النظام يدعم تقنيات متقدمة مثل Sass، مما يمنح المطورين حرية أكبر في التحكم بالتفاصيل الدقيقة للتصميم دون الحاجة إلى كتابة كود CSS يدوياً.

على الرغم من أن Bootstrap يوفر العديد من المزايا، إلا أنه يواجه بعض التحديات التي يجب أخذها في الاعتبار. من أبرز هذه التحديات هو حجم الملفات المرتبطة بالإطار، حيث يمكن أن يؤدي استخدام جميع مكوناته إلى زيادة زمن تحميل الصفحات إذا لم يتم تحسينها بشكل صحيح. هذا الأمر يتطلب من المطورين اتخاذ خطوات إضافية لاختيار العناصر التي يحتاجونها فقط، بدلاً من

تضمين جميع المكونات بشكل افتراضي. كما أن الاعتماد الكبير على الإطار قد يؤدي إلى تقليل قدرة المطورين على التعامل مع التقنيات الأساسية مثل CSS وJavaScript بشكل مستقل.

في سياق آخر، يلعب Bootstrap دوراً مهماً في تعزيز التجربة العامة للمستخدمين النهائيين. من خلال توفير أدوات تسهم في تحسين التنقل والتفاعل داخل الموقع، يتمكن المطورون من تقديم واجهات مستخدم أكثر سلاسة وجاذبية. هذا الجانب يساهم بشكل كبير في رفع مستوى رضا المستخدمين عن الموقع أو التطبيق، مما يعزز من فرص نجاح المشروع على المدى الطويل. كما أن توافق Bootstrap مع مختلف المتصفحات الحديثة يجعله خياراً موثوقاً لضمان ظهور التصميمات بشكل متسق عبر جميع الأجهزة والأنظمة.

علاوة على ذلك، فإن المجتمع النشط الذي يحيط بـ Bootstrap يمثل عاملاً مهماً في تعزيز قيمته كأداة تطوير. يوفر هذا المجتمع دعماً مستمراً من خلال تقديم الحلول للمشكلات الشائعة، بالإضافة إلى توفير موارد تعليمية وإضافات جديدة تساعد المطورين على استكشاف الإمكانيات الكاملة للإطار. هذا الدعم المجتمعي يجعل من السهل على المطورين الجدد الاندماج في عالم Bootstrap والاستفادة من الخبرات السابقة لتحسين مشاريعهم الخاصة.

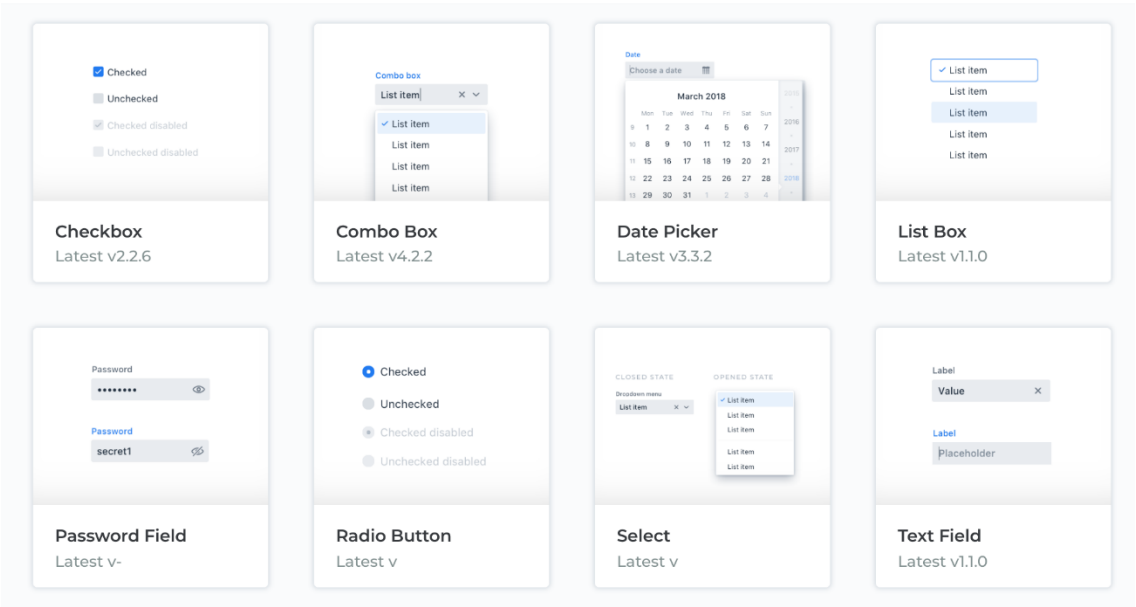
2-3-3-1 المكونات الأساسية لـ Bootstrap

يقدم Bootstrap مجموعة شاملة من الأدوات التي تجعل عملية بناء واجهات الويب أكثر كفاءة واحترافية (الشكل 2-18)، حيث يعتمد على مزيج من الأنماط الجاهزة والتقنيات الحديثة لتحقيق التوافق مع مختلف الاحتياجات. من بين هذه الأدوات، يمكن للمطورين الاستفادة من مجموعة

واسعة من الأنماط المدمجة التي تشمل عناصر مثل الأزرار، الجداول، النماذج، والقوائم المنبثقة. هذه العناصر تم تصميمها بعناية لتوفير مظهر احترافي دون الحاجة إلى كتابة كود CSS يدوي، مما يقلل من الوقت اللازم لتنفيذ العمليات الأساسية. بالإضافة إلى ذلك، فإن هذه الأنماط قابلة للتخصيص بشكل كبير، مما يتيح إنشاء تصاميم فريدة تعكس هوية المشروع الفردي [13].

في سياق آخر، يعتمد Bootstrap على نظام شبكة مرن (Grid System) يتيح إنشاء تصاميم تتكيف مع أحجام الشاشات المختلفة، بدءاً من الهواتف الذكية وصولاً إلى شاشات سطح المكتب الكبيرة. هذا النظام يستخدم تقنية القوالب المرنة التي تعتمد على أعمدة وصفوف، مما يجعل من السهل تنظيم المحتوى بطريقة مرنة ومنطقية. على سبيل المثال، يمكن تقسيم الصفحة إلى عدة أعمدة ذات عرض متغير حسب حجم الشاشة، مما يضمن ظهور المحتوى بشكل جيد ومريح للعين في أي جهاز.

علاوة على ذلك، يوفر Bootstrap مجموعة من المكونات التفاعلية التي تعتمد على JavaScript، مثل النوافذ المنبثقة (Modals)، شرائح العرض (Carousels)، وأشرطة التنقل (Navbars). هذه المكونات تسهم في إضافة وظائف ديناميكية إلى الموقع دون الحاجة إلى كتابة كود JavaScript من الصفر، مما يعزز من كفاءة التطوير ويقلل من التعقيدات التقنية. كما أن هذه المكونات مصممة لتكون متوافقة مع جميع المتصفحات الحديثة مثل Chrome، Firefox، Safari، وEdge، مما يضمن ظهور التصميمات بشكل متسق عبر مختلف المنصات.



الشكل 18-2: بعض المكونات التي يوفرها إطار العمل Bootstrap.

بالإضافة إلى ذلك، يتيح Bootstrap إمكانية تخصيص الأنماط والألوان والخطوط باستخدام ملفات Sass، وهي تقنية متقدمة تمنح المطورين حرية أكبر في التحكم بالتفاصيل الدقيقة للتصميم. من خلال هذه الملفات، يمكن تعديل المتغيرات الأساسية مثل الألوان الرئيسية، الخطوط، وحجم العناصر، مما يجعل من السهل تحقيق هوية بصرية فريدة لكل مشروع. هذا الجانب من المرونة يجعل Bootstrap أداة مثالية لمشاريع تتطلب تصميماً مخصصاً دون فقدان الفوائد التي يقدمها الإطار الأساسي [13].

المكونات الأساسية لـ Bootstrap تعمل معاً بشكل متناغم لتقديم حلول شاملة تلبي احتياجات المطورين والمصممين على حد سواء. من خلال الجمع بين الأنماط الجاهزة، نظام الشبكة المرن، المكونات التفاعلية، والتخصيص باستخدام Sass، يصبح من الممكن بناء واجهات مستخدم

احترافية وديناميكية بسرعة وكفاءة. هذه الميزات تجعل Bootstrap خياراً استراتيجياً لتطوير المشاريع الرقمية الحديثة، مع ضمان توافق التصميمات مع مختلف الأجهزة والمتصفحات.

2-3-3-2 أهمية Bootstrap

تُعتبر أهمية Bootstrap في عالم تطوير الويب أمراً لا يمكن إغفاله، حيث يمثل هذا الإطار أداة أساسية تسهم في تسريع عملية بناء واجهات المستخدم بطريقة احترافية ومتجاوبة. من خلال توفير مجموعة شاملة من الأدوات والمكونات الجاهزة، يتيح Bootstrap للمطورين التركيز على الجوانب الإبداعية والوظيفية للمشاريع بدلاً من الانشغال بتفاصيل التصميم الأساسية. هذه الميزة تجعله خياراً مثالياً للمشاريع التي تتطلب سرعة في التنفيذ دون التضحية بالجودة أو التناسق البصري. بالإضافة إلى ذلك، فإن النظام يدعم المرونة اللازمة لتلبية احتياجات المشاريع المختلفة، مما يجعله قابلاً للتكيف مع أي نوع من التطبيقات، سواء كانت مواقع شركات صغيرة أو تطبيقات ويب معقدة [13].

من الجوانب البارزة التي تعزز من قيمة Bootstrap هو دوره في تحسين تجربة المستخدم النهائي. من خلال تقديم تصاميم متجاوبة تتكيف مع أحجام الشاشات المختلفة، يتمكن المطورون من ضمان ظهور المحتوى بشكل جيد ومريح للعين على أي جهاز. هذا الجانب يلعب دوراً كبيراً في رفع مستوى رضا المستخدمين عن الموقع أو التطبيق، مما يعزز من فرص نجاح المشروع على المدى الطويل. كما أن توافق Bootstrap مع مختلف المتصفحات الحديثة يجعله أداة موثوقة

لضمان ظهور التصميمات بشكل متسق عبر جميع الأجهزة والأنظمة، وهو ما يقلل من الحاجة إلى إجراء تعديلات إضافية لكل بيئة تشغيل.

علاوة على ذلك، يساهم Bootstrap في تقليل التعقيدات التقنية المرتبطة بتطوير واجهات الويب. من خلال توفير مكونات تفاعلية تعتمد على JavaScript، مثل النوافذ المنبثقة وأشرطة التنقل، يصبح من السهل إضافة وظائف ديناميكية إلى الموقع دون الحاجة إلى كتابة كود معقد من الصفر. هذه المكونات ليست فقط سهلة الاستخدام، بل إنها مصممة أيضاً لتكون متوافقة مع مختلف البيئات التشغيلية، مما يضمن تحقيق الأداء الأمثل دون الحاجة إلى القلق بشأن مشكلات التوافق. كما أن النظام يقدم أدوات للتخصيص باستخدام تقنيات متقدمة مثل Sass، مما يمنح المطورين حرية أكبر في التحكم بالتفاصيل الدقيقة للتصميم.

في سياق آخر، يلعب المجتمع النشط الذي يحيط بـ Bootstrap دوراً مهماً في تعزيز قيمته كأداة تطوير. يوفر هذا المجتمع دعماً مستمراً من خلال تقديم الحلول للمشكلات الشائعة، بالإضافة إلى توفير موارد تعليمية وإضافات جديدة تساعد المطورين على استكشاف الإمكانيات الكاملة للإطار. هذا الدعم المجتمعي يجعل من السهل على المطورين الجدد الاندماج في عالم Bootstrap والاستفادة من الخبرات السابقة لتحسين مشاريعهم الخاصة. كما أن وجود وثائق رسمية مفصلة ومصادر تعليمية غنية يساهم في تسهيل عملية التعلم والاستخدام الفعال لهذا الإطار [13].

2-3-3-3 عيوب إطار العمل Bootstrap

- قد تعاني المنصة من قيود في المرونة عند التعامل مع مشاريع تتطلب تصميمات غير تقليدية أو معقدة للغاية، حيث إن الأنماط والمكونات الجاهزة التي توفرها قد لا تكون كافية لتلبية احتياجات التصميم الفريدة، مما يجبر المطورين على إجراء تعديلات مكثفة.
- تواجه المنصة تحدياً في إدارة حجم الملفات الناتجة عن استخدام جميع المكونات بشكل افتراضي، حيث يمكن أن يؤدي ذلك إلى زيادة زمن تحميل الصفحات، خاصة إذا لم يتم تحسين الموارد أو اختيار العناصر الضرورية فقط.
- قد تكون بعض المكونات التفاعلية المدمجة في المنصة غير مرنة بما يكفي لتلبية متطلبات المشاريع الكبيرة أو المعقدة، مما يتطلب من المطورين كتابة كود إضافي أو حتى استبدال هذه المكونات بحلول خارجية أكثر تخصيصاً.
- رغم أن المنصة تدعم التخصيص باستخدام أدوات مثل Sass، إلا أن عملية التخصيص قد تكون معقدة نسبياً للمستخدمين الذين ليس لديهم خبرة كافية بهذه التقنيات، مما يحد من قدرتهم على تحقيق التصميم المطلوب دون مساعدة خارجية.
- تعتمد المنصة بشكل كبير على JavaScript لتحقيق الوظائف التفاعلية، وهو ما قد يؤدي إلى مشكلات في الأداء إذا كانت البيئة المستخدمة لتشغيل التطبيق لا تدعم JavaScript بشكل كامل أو إذا كان هناك تعارض بين الإصدارات المختلفة.

- قد تكون هناك صعوبة في تحقيق التوافق الكامل مع بعض المتصفحات القديمة أو غير الشائعة، حيث إن المنصة تركز بشكل أساسي على دعم المتصفحات الحديثة، مما قد يتطلب جهداً إضافياً لحل مشكلات العرض أو الأداء في البيئات غير المدعومة.
- تواجه المنصة تحدياً في تقديم حلول شاملة لمشاريع تتطلب أداءً عالياً مع موارد محدودة، حيث إن بعض المكونات قد تكون ثقيلة نسبياً وتستهلك موارد النظام بشكل أكبر مما هو مقبول في مشاريع صغيرة أو بسيطة.
- قد يكون من الصعب تحقيق التوازن بين سهولة الاستخدام والمرونة اللازمة لمشاريع كبيرة ومعقدة، حيث إن الاعتماد الكامل على المكونات الجاهزة قد يؤدي إلى فقدان السيطرة على بعض الجوانب التقنية الدقيقة التي تتطلب حلولاً مخصصة.
- تواجه المنصة تحدياً في تقديم تحديثات مستمرة دون التأثير على التوافق مع الإصدارات السابقة، حيث إن التغييرات التي يتم إدخالها في الإصدارات الجديدة قد تؤدي إلى مشكلات في المشاريع القديمة التي تعتمد على إصدارات سابقة.
- قد تكون الوثائق الرسمية للمنصة غير كافية لتغطية جميع السيناريوهات المتقدمة، مما يجبر المطورين على البحث عن حلول خارجية أو الاعتماد على التجربة والخطأ لاكتشاف الحلول المناسبة لمشكلاتهم الفنية.

3

الفصل الثالث: التحليل وتصميم النظام

3-1 متطلبات النظام

3-1-1 المتطلبات الوظيفية

1. **نظام تسجيل الدخول والمصادقة:** يتيح النظام للمستخدمين تسجيل الدخول بناءً على أدوارهم (ولي أمر، إدارة)، مع التحقق من الهوية وتفعيل صلاحيات الوصول المناسبة.
2. **إدارة بيانات الأطفال:** تُمكّن لوحة التحكم من إضافة، تعديل، أو حذف بيانات الأطفال مع ربط كل طفل بولي أمر وفصل دراسي محدد.
3. **عرض تقارير التقدم الأكاديمي والصحي:** يُتيح النظام لأولياء الأمور استعراض تقارير الأداء الأكاديمي والصحي لأطفالهم بشكل دوري ومنظم.
4. **سجلات الحضور والانصراف:** يتم تسجيل حضور الأطفال تلقائيًا يوميًا، مع إمكانية توليد تقارير إحصائية للإدارة وأولياء الأمور.
5. **نظام إرسال الإعلانات والإشعارات:** يمكن للإدارة إرسال تنبيهات أو إعلانات عامة أو موجهة لأولياء الأمور بشأن الأنشطة أو الملاحظات المهمة.
6. **إدارة الجداول اليومية:** توفر لوحة التحكم أدوات لإنشاء جداول زمنية يومية للفصول الدراسية والأنشطة وعرضها على المستخدمين.
7. **تحميل واستعراض الوسائط المتعددة:** يُمكن إضافة صور أو مقاطع فيديو للأنشطة التعليمية والاحتفاظ بها ضمن ملفات الأطفال للاستعراض لاحقًا.

8. **التواصل المباشر:** يُتيح النظام تبادل الرسائل بين أولياء الأمور والإدارة أو المعلمين داخل بيئة آمنة وخاصة.

9. **عرض قائمة الوجبات اليومية:** يُعرض الجدول الغذائي الخاص بالأطفال بطريقة مرئية تفاعلية وفقًا للأيام والمناسبات.

10. **دعم الأنشطة التعليمية التفاعلية:** يقدم النظام محتوى تعليميًا تفاعليًا مناسبًا لمرحلة الطفولة، يمكن الوصول إليه من خلال التطبيق.

3-1-2 المتطلبات غير الوظيفية

1. **قابلية الاستخدام (Usability):** يُصمم النظام بواجهة استخدام بسيطة وسلسة تدعم اللغة العربية وتناسب جميع فئات المستخدمين.

2. **الأداء (Performance):** يجب أن تكون سرعة استجابة النظام عند تنفيذ العمليات الأساسية أقل من ثانيتين لضمان تجربة استخدام فعالة.

3. **الاعتمادية (Reliability):** يعمل النظام بثبات دون أعطال متكررة، ويعالج الأخطاء بطريقة منظمة لضمان استمرارية العمل.

4. **قابلية التوسع (Scalability):** يُمكن تطوير النظام مستقبلًا ليدعم عددًا أكبر من المستخدمين أو الوظائف الإضافية دون الحاجة إلى إعادة بناءه بالكامل.

5. **الأمان (Security):** يتم استخدام التشفير وإدارة الصلاحيات لحماية بيانات المستخدمين، خاصة ما يتعلق بالأطفال وأولياء الأمور.

6. **قابلية النقل (Portability):** يعمل التطبيق على أجهزة متعددة بما في ذلك هواتف Android وiOS دون الحاجة إلى تغييرات كبيرة.
7. **الصيانة (Maintainability):** يُكتب الكود بطريقة منظمة، مع وجود توثيق داخلي يسهل عملية التعديل أو الإضافة مستقبلًا.
8. **التكامل (Integration):** يتيح النظام التكامل مع أنظمة أو خدمات خارجية مستقبلًا من خلال واجهة API موثقة وفق معايير OpenAPI.
9. **التوفر (Availability):** يجب أن يكون النظام متاحًا للمستخدمين على مدار الساعة مع أوقات توقف محدودة للصيانة فقط.
10. **الامتثال للمعايير (Standards Compliance):** يلتزم النظام بأفضل ممارسات تطوير البرمجيات، بما في ذلك RESTful API ومعايير أمان وتوثيق حديثة.

2-3 تحليل المستخدمين والأدوار

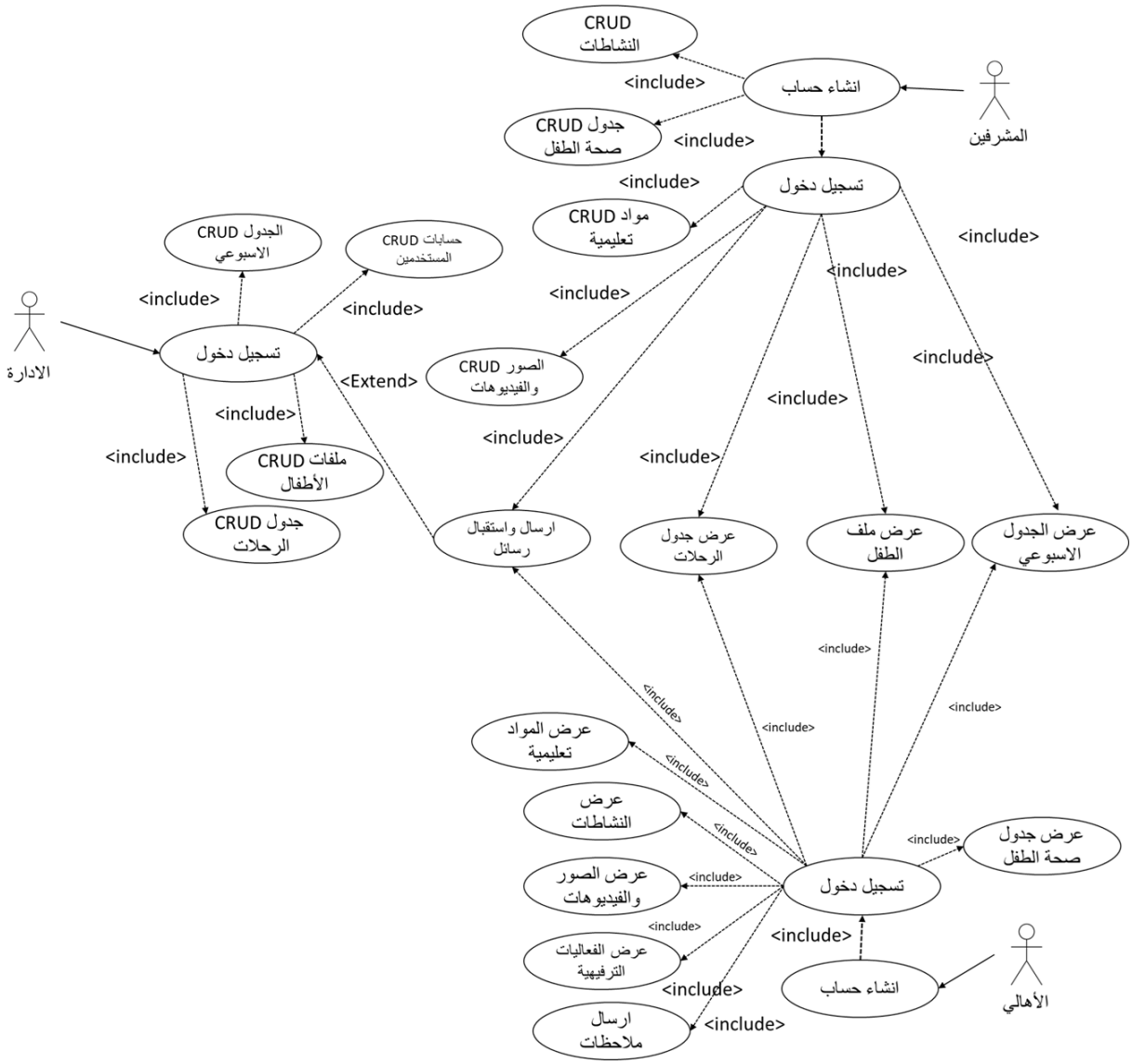
في نظام إدارة رياض الأطفال المقترح، تم تحديد ثلاث فئات رئيسية من المستخدمين، وهم أولياء الأمور، إدارة الروضة، والأطفال. يمثل أولياء الأمور فئة محورية في هذا النظام، حيث تم تصميم واجهة استخدام خاصة بهم تتيح لهم الوصول إلى معلومات شاملة عن أطفالهم، بما في ذلك متابعة التقدم الأكاديمي والسجلات الصحية، والاطلاع على الأنشطة اليومية والفعاليات المخطط لها داخل الروضة. كما يتيح النظام لهم التواصل المباشر مع الإدارة لتقديم الملاحظات أو الاستفسارات، إضافة إلى الاطلاع على الجداول الزمنية وقوائم الوجبات والمحتوى التعليمي التفاعلي المتاح.

أما إدارة الروضة فهي الجهة المسؤولة عن تشغيل النظام والإشراف الكامل على مكوناته ووظائفه. وتمتلك هذه الفئة صلاحيات موسعة تشمل إدارة بيانات الأطفال والفصول الدراسية، وتنظيم الجداول الزمنية والأنشطة، ونشر الإعلانات والوسائط التعليمية، بالإضافة إلى إدارة حسابات المستخدمين وتحديد صلاحياتهم بناءً على الأدوار المخصصة. وتعتبر هذه الصلاحيات ضرورية لضمان سلاسة العمل الداخلي وتوفير تجربة رقمية منظمة تلبي متطلبات الروضة التربوية والإدارية.

فيما يتعلق بفئة الأطفال، فإن استخدامهم للنظام محدود بطبيعته، إذ يُعدّ الطفل محور البيانات وليس مستخدمًا مباشرًا بالمعنى التقليدي. ومع ذلك، يمكن أن يشمل النظام مستقبلاً خصائص تفاعلية مبسطة موجهة للأطفال، مثل الوصول إلى أنشطة تعليمية تفاعلية تتناسب مع أعمارهم، أو عرض الوسائط الخاصة بهم كصور الأنشطة التي شاركوا فيها. ويتم تصميم هذا الاستخدام المحدود بعناية لضمان الجوانب التربوية والحفاظ على خصوصية الأطفال وأمانهم داخل بيئة رقمية آمنة.

3-3 تحليل الحالات والاستخدامات (Use Case Analysis)

يعرض مخطط حالات الاستخدام الموضح في الشكل (3-1) التفاعلات الأساسية التي تتم بين أنواع المستخدمين الثلاثة في النظام—الإدارة، والأهالي، والمشرفين—مع الوظائف المختلفة التي يوفرها النظام. يُظهر المخطط أن جميع المستخدمين يبدؤون تفاعلهم من خلال حالة الاستخدام الأساسية "تسجيل الدخول"، والتي تُعد شرطًا مبدئيًا للوصول إلى باقي وظائف النظام.



الشكل 1-3: Use Case Analysis Diagram.

أما المشرفون، فيُمنحون إمكانية إنشاء حساب جديد، وهي حالة استخدام مرتبطة أيضًا بالأهالي، مما يشير إلى أن المشرف مسؤول عن تفعيل حسابات أولياء الأمور.

بالنسبة للإدارة، يتضح أن لديها صلاحيات واسعة تشمل عمليات CRUD (إنشاء، قراءة، تحديث، حذف) على عدد من الوحدات، مثل: ملفات الأطفال، وجدول الرحلات، والجدول الأسبوعي، وجدول صحة الطفل، والأنشطة، والمواد التعليمية، والمستخدمين. تُشير خطوط <include> إلى أن هذه الوظائف تُنفّذ بشكل دائم عند استدعاء حالات استخدام مرتبطة بها. على سبيل المثال، حالة استخدام "ملفات الأطفال" تتضمن بشكل تلقائي "جداول الرحلات" و"صحة الطفل" وغيرهما.

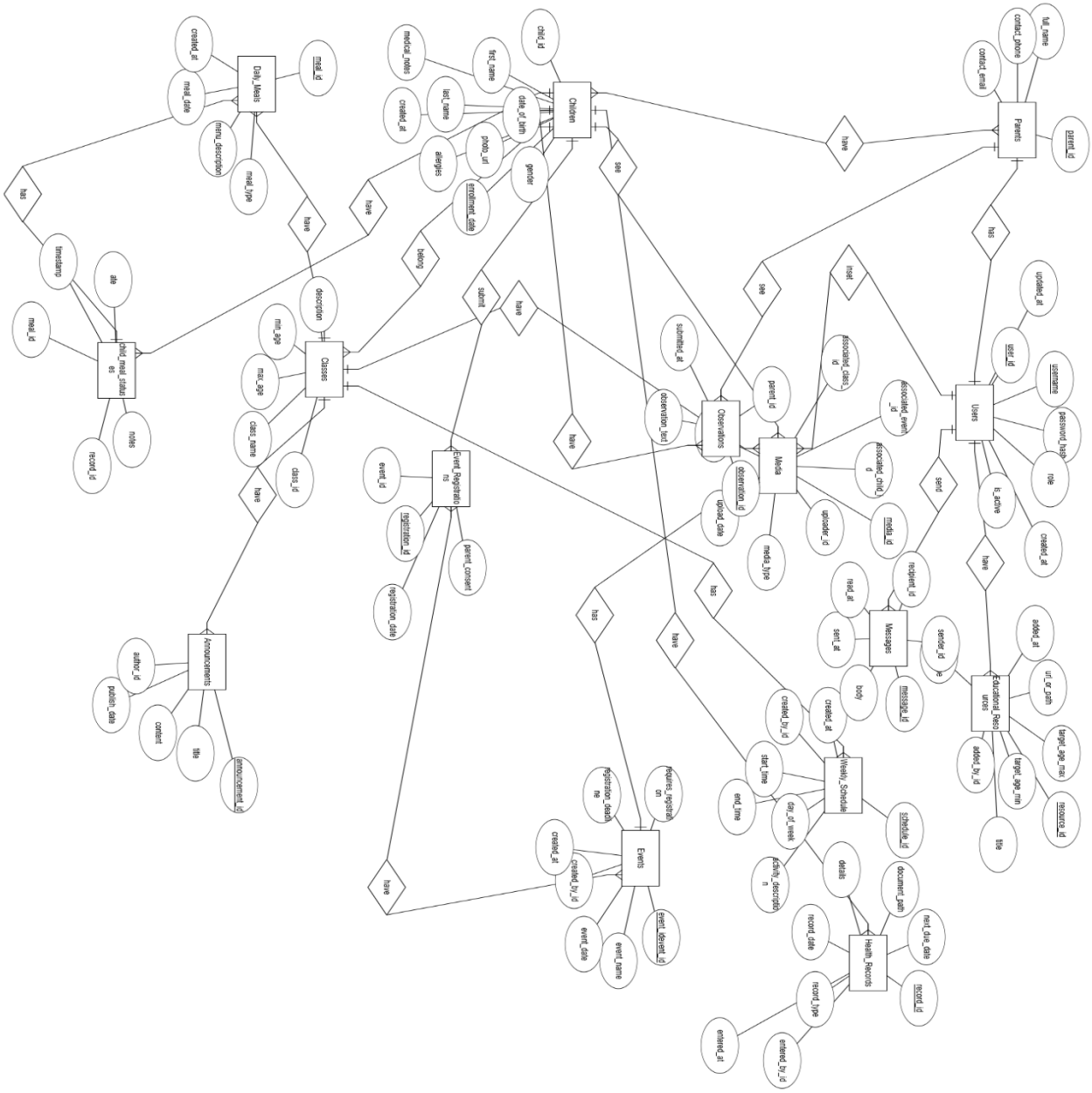
تتمتع الإدارة أيضًا بقدرة على إرسال واستقبال الرسائل، ما يعزز التواصل داخل النظام. كما أن لها إمكانية عرض الصور والفيديوهات، مما يشير إلى إمكانية إدارة الوسائط المتعددة المرتبطة بأنشطة الأطفال أو الفعاليات اليومية.

أما الأهالي، فلديهم مجموعة من الوظائف المرتبطة مباشرة بتتبع ومتابعة طفلهم. يمكنهم عرض ملف الطفل، وجدوله الأسبوعي، وجدول صحته، بالإضافة إلى عرض المواد التعليمية، والأنشطة، والصور، والفيديوهات، وهو ما يعزز دورهم في متابعة الأداء اليومي والأكاديمي. كما يمكنهم عرض التغذيات الراجعة التربوية وإرسال ملاحظات، مما يشير إلى وجود قناة تفاعلية بين ولي الأمر والإدارة أو المشرف.

يجسّد هذا المخطط العلاقة التفاعلية بين المكونات البشرية للنظام ووظائفه البرمجية، ويعكس تبنّي مبدأ الفصل بين الأدوار (Role-Based Access) من خلال تخصيص صلاحيات ووظائف مختلفة لكل فئة من المستخدمين. كما أن وجود عمليات CRUD للإدارة يدل على أن النظام

مصمم ليكون مرنًا من حيث التحكم بالبيانات، بينما تم تقييد وظائف الأهالي لتكون قائمة على العرض والتفاعل، بما يناسب طبيعة استخدامهم للنظام.

4-3 مخطط ERD



الشكل 2-3: مخطط ERD.

5-3 تصميم قواعد البيانات

الشكل 3-3: مخطط قاعدة البيانات.

يُعد مخطط الكيانات والعلاقات أحد الأدوات الأساسية في تصميم قواعد البيانات العلائقية، حيث يساعد على توضيح الهيكل المنطقي للبيانات والعلاقات بين الجداول المختلفة في نظام إدارة رياض الأطفال. ويعكس هذا المخطط مستوى عاليًا من التنظيم والارتباط بين الكيانات الأساسية التي تخدم أدوار المستخدمين واحتياجات النظام.

يضم المخطط عدة كيانات رئيسية أبرزها: users, children, health_records, weekly_schedules, activities, educational_materials, meals, messages, feedbacks, وmedia_files. يُلاحظ أن الكيان المركزي في المخطط هو users، والذي يُمثل جميع أنواع المستخدمين في النظام مثل الإدارة، الأهالي، والمشرفين. يرتبط كيان users بعلاقات متعددة مع كيانات أخرى، ما يعكس تعدد الأدوار والوظائف المرتبطة بكل مستخدم.

كيان children يُمثل الأطفال المسجلين في الروضة، ويرتبط مباشرة بكيان users من خلال علاقة (many-to-one)، حيث يمكن لكل مستخدم من نوع "ولي أمر" أن يكون مسؤولاً عن أكثر من طفل، في حين أن كل طفل مرتبط بمستخدم واحد فقط كولي أمر. كما أن كيان children يرتبط بعلاقات مباشرة مع عدد من الكيانات الأخرى مثل health_records، weekly_schedules، feedbacks، meals، وmedia_files، مما يشير إلى أن معظم بيانات النظام تتمحور حول الطفل.

الكيانات المرتبطة بالصحة مثل health_records تمثل سجلات طبية مخصصة لكل طفل، وتتضمن معلومات متغيرة زمنياً، مما يجعل العلاقة (one-to-many) منطقية. بالمثل، يعبر كيان

weekly_schedules عن جداول تنظيمية أسبوعية تشمل الأنشطة والوجبات، ويرتبط أيضًا بالطفل بعلاقة مشابهة، حيث يمكن أن يكون لكل طفل أكثر من جدول خلال فترة زمنية مختلفة.

كيان activities وeducational_materials يُظهران المحتوى الذي توفره الإدارة للأنشطة التعليمية والترفيهية، ويرتبطان غالبًا بالإدارة أو المشرفين كمصدر إداري للمحتوى، في حين تكون هذه الكيانات قابلة للعرض من قبل أولياء الأمور في النظام، دون أن تكون العلاقة مباشرة في هذا المخطط.

أما كيان messages فيعبر عن وسيلة التواصل بين الأطراف المختلفة في النظام، وخاصة بين أولياء الأمور والإدارة، وهو ما يعكس التفاعل داخل المنصة. أما feedbacks فهي ملاحظات يتم تسجيلها من قبل الإدارة أو المشرفين وتتعلق بسلوك أو أداء الطفل، وهي مرتبطة بالطفل نفسه ومرتبطة بمستخدم قام بإنشائها. كيان media_files يُستخدم لتخزين الصور والفيديوهات المرتبطة بالأنشطة، مما يدعم الجوانب التوثيقية والتفاعلية للنظام.

يتميز هذا المخطط بتكامل العلاقات بين الكيانات المختلفة، مما يسهل تنفيذ العمليات الأساسية مثل الإنشاء والتعديل والحذف (CRUD) دون فقدان الترابط أو الاتساق بين الجداول. كما أن هذا التصميم يدعم التوسع المستقبلي في وظائف النظام، مثل إضافة تقارير أو ربط واجهات جديدة، مما يجعله قاعدة متينة لبناء نظام إدارة متكامل وفعال لرياض الأطفال.

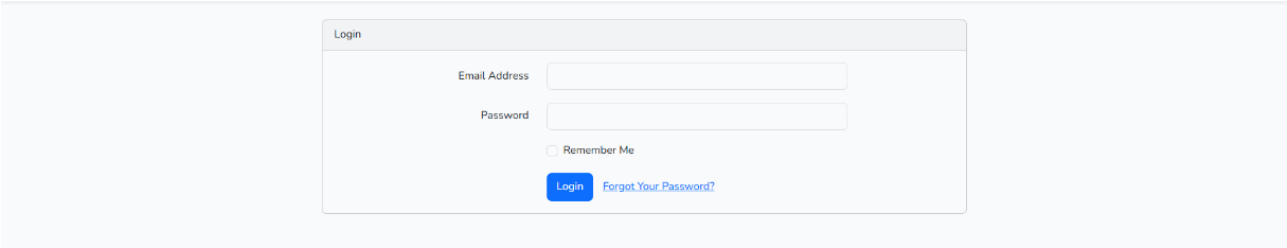
4

الفصل الرابع: التنفيذ

4-1 واجهة لوحة تحكم المدير (Web Dashboard Interface)

4-1-1 واجهة تسجيل الدخول

يظهر في الشكل (4-1) واجهة تسجيل الدخول الخاصة بلوحة تحكم المدير ضمن الواجهة الخلفية للنظام، والمبنية على إطار Laravel. تتضمن الحقول الأساسية مثل البريد الإلكتروني وكلمة المرور، إلى جانب خيار تذكّر البيانات وإمكانية استعادة كلمة المرور، مما يضمن سهولة الوصول وسلامة العملية الإدارية.

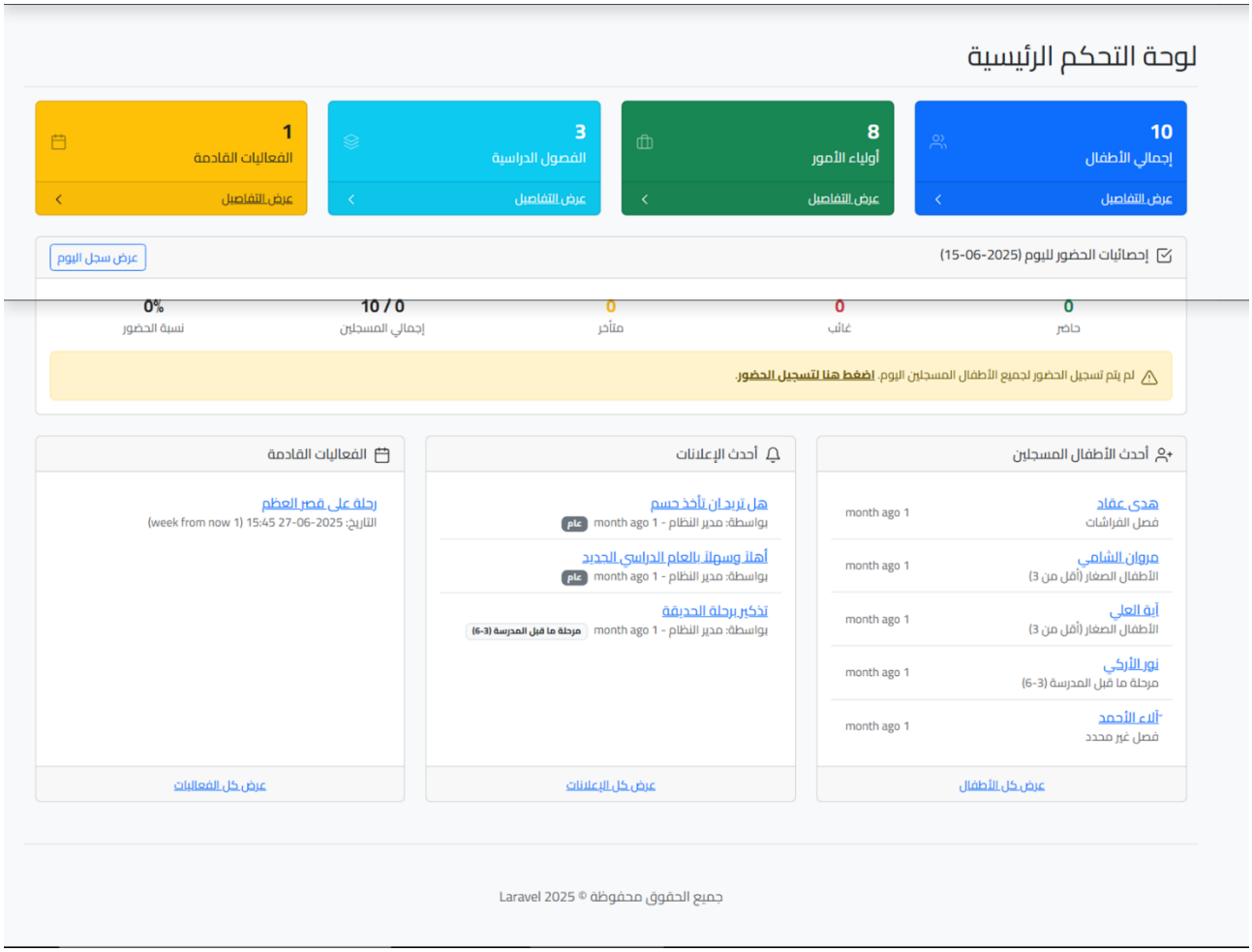


الشكل 4-1: واجهة تسجيل الدخول.

4-1-2 لوحة التحكم الرئيسية

يظهر الشكل (4-2) لوحة التحكم الرئيسية في الواجهة الخلفية للنظام، وهي تُعد النقطة الأساسية التي توفر لمديري النظام رؤية شاملة ومباشرة حول العمليات اليومية. تتضمن اللوحة عناصر رئيسية مثل نسبة الحضور والتفاصيل المتعلقة بالأطفال المسجلين (إجمالي الأعضاء، المتغيبين، الحاضرين)، بالإضافة إلى أحدث الفعاليات القادمة والإعلانات المنشورة. كما تعرض قائمة بأحدث الأطفال المسجلين مع تفاصيل عن فصولهم ومستوياتهم التعليمية. هذه الواجهة مصممة لتوفير

تجربة إدارية سلسة ومرتبطة بالبيانات الحيوية، مما يساعد المديرين على إدارة العمليات بفعالية وسرعة.

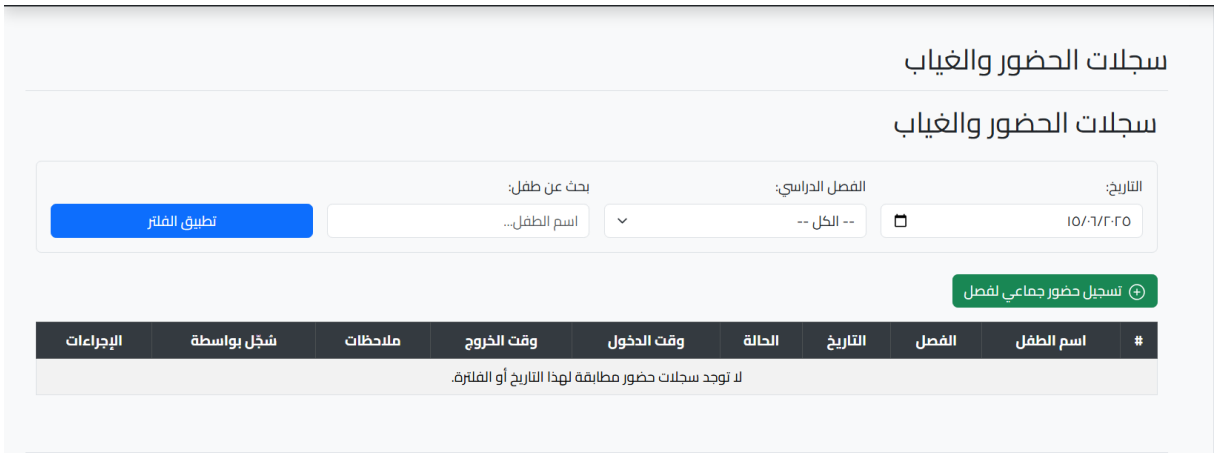


الشكل 2-4: لوحة التحكم الرئيسية.

4-1-3 الحضور والغياب

يظهر الشكل (4-3) واجهة سجلات الحضور والغياب في الواجهة الخلفية للنظام، وهي تُعد أداة أساسية لتسجيل ومتابعة حضور الأطفال المسجلين. تتضمن الواجهة عدة عناصر رئيسية مثل مربع بحث عن الطفل، واختيار الفصل الدراسي، وتاريخ التسجيل، بالإضافة إلى زر تسجيل الحضور

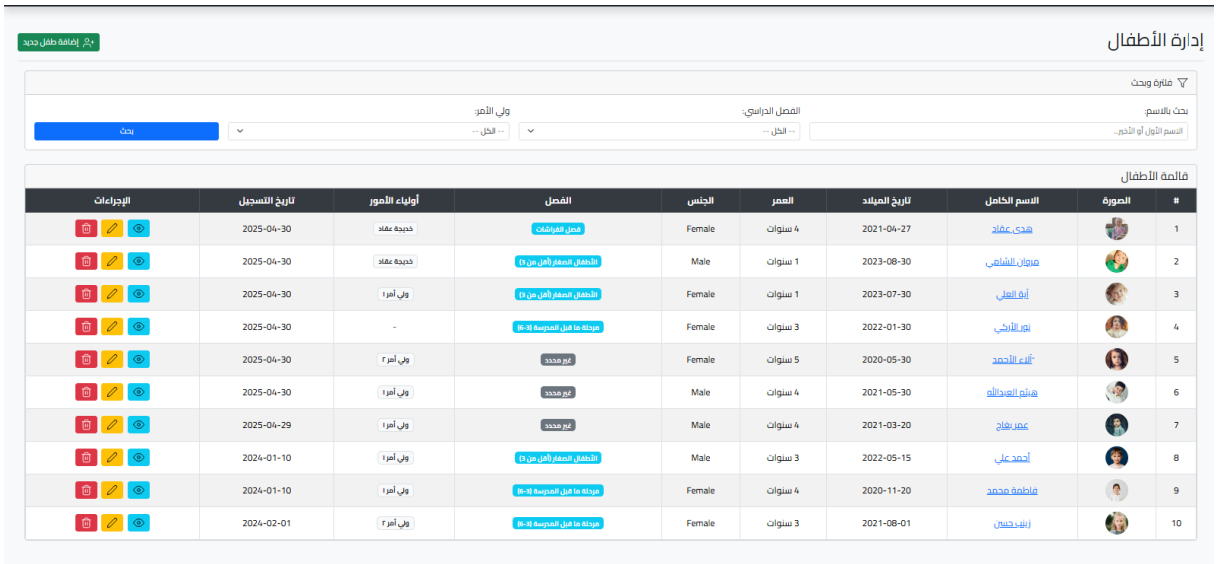
الجماعي للفصل. كما تحتوي على جدول يعرض بيانات الحضور والغياب لكل طفل، بما في ذلك اسم الطفل، الفصل الدراسي، التاريخ، وقت الدخول والخروج، حالة الحضور (حاضر أو غائب)، والملاحظات الإضافية.



الشكل 3-4: واجهة الحضور والغياب.

4-1-4 إدارة الأطفال

تُعرض في هذه الواجهة بيانات شاملة للأطفال المسجلين ضمن النظام، حيث تُظهر قائمة بأسمائهم مرفقة بصورهم ومعلومات تفصيلية مثل العمر، الجنس، تاريخ الميلاد، الفصل الدراسي، واسم ولي الأمر. تتيح الواجهة أدوات مباشرة لتعديل أو حذف السجلات من خلال أيقونات واضحة، مع وجود خيارات بحث متقدمة لتصفية البيانات بحسب الاسم أو الفصل أو تاريخ الميلاد. كما تحتوي على زر مخصص لإضافة طفل جديد، مما يجعلها أداة عملية تُمكن الإدارة من التعامل مع سجلات الأطفال بسهولة وكفاءة، وبشكل منظم يلبي احتياجات المتابعة اليومية.



الشكل 4-4: واجهة إدارة الأطفال.

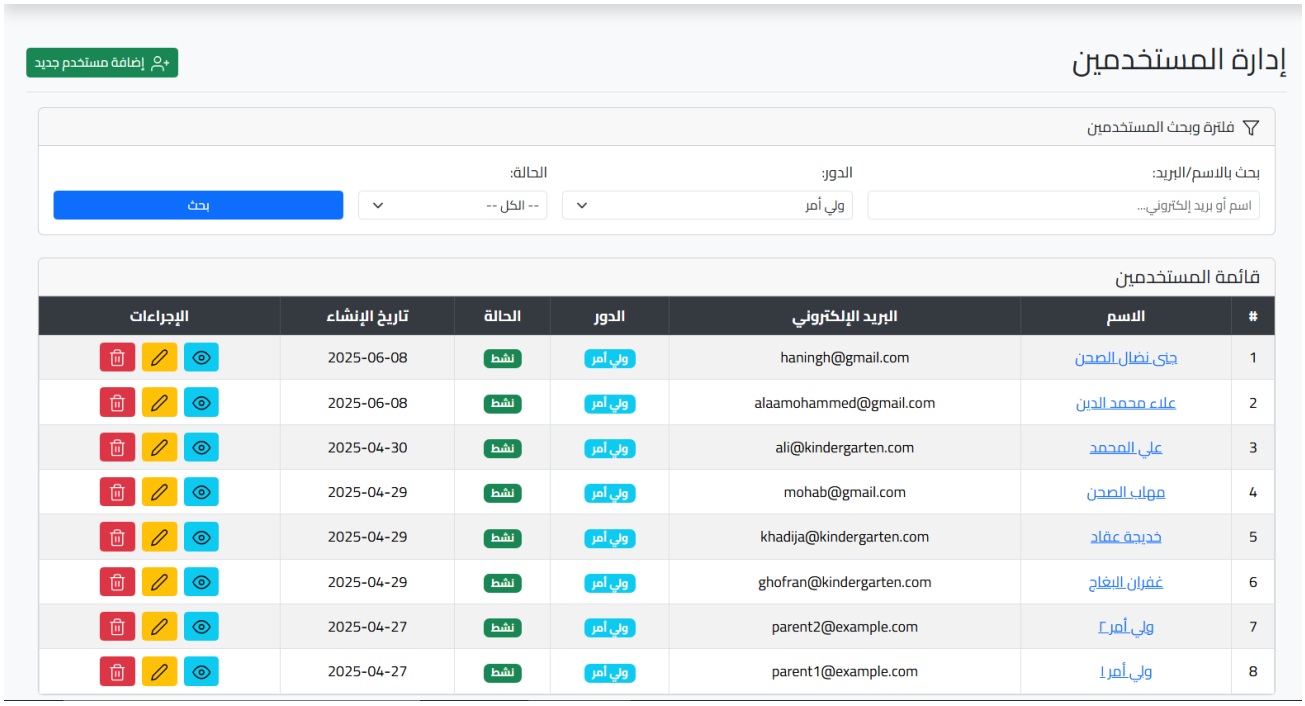
4-1-5 إدارة الفصول

تعرض هذه الواجهة قائمة منظمة للفصول الدراسية ضمن النظام، حيث تحتوي على جدول يُظهر معلومات تفصيلية لكل فصل، مثل الاسم، الوصف، الفئة العمرية المستهدفة، عدد الأطفال المسجلين، وتاريخ إنشاء الفصل. توفر الواجهة أدوات تحكم مباشرة لكل سجل، من خلال أيقونات تتيح التعديل، الحذف، أو عرض التفاصيل.

الشكل 5-4: واجهة إدارة الفصول الدراسية.

6-1-4 إدارة أولياء الأمور

تعرض هذه الواجهة سجلات المستخدمين المرتبطين بالنظام، مع تصنيفهم حسب الدور (مثل ولي أمر، معلمة، مدير...). يمكن من خلالها متابعة حالة الحسابات (نشطة أو غير نشطة) وتعديلها حسب الحاجة. تُسهل أدوات التصفية والبحث المتقدم الوصول للمستخدمين وفق الاسم أو البريد الإلكتروني، وتُستخدم الأيقونات المختصرة لتنفيذ العمليات الإدارية على الحسابات بشكل مباشر. تُعد هذه الصفحة محورًا لتنظيم وصول الأفراد للنظام وتحديد صلاحياتهم.

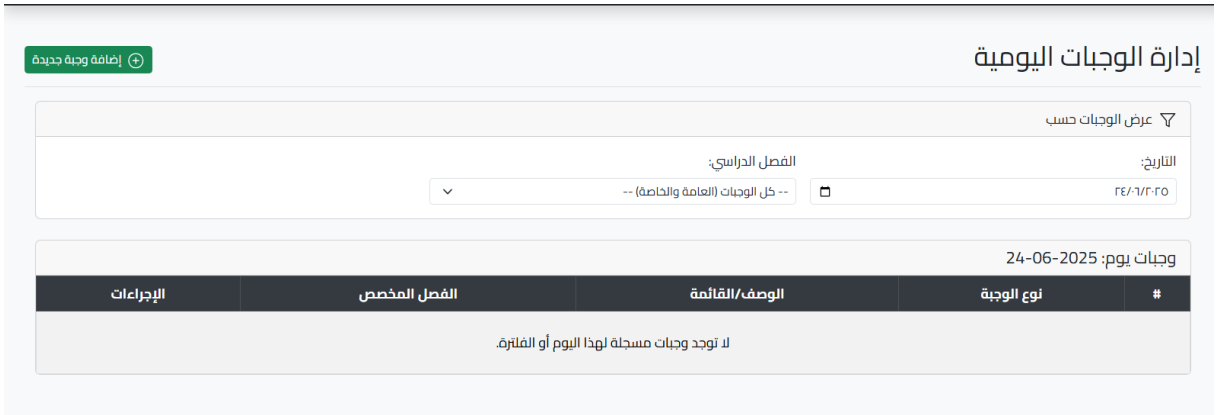


الشكل 6-4: واجهة إدارة أولياء الأمور.

7-1-4 إدارة الوجبات

تعنى هذه الواجهة بعرض وتحديث بيانات الوجبات المقدمة للأطفال، حيث تتيح تحديد تاريخ العرض واختيار الفصل الدراسي لفرز المحتوى المعروض. تتضمن أدوات تحكم في الأعلى تُمكن

من إضافة وجبة جديدة وتغيير السياق الزمني أو الأكاديمي للعرض. يظهر في الجدول المخصص بيانات تفصيلية للوجبات، مثل النوع والوصف وتاريخ الإضافة.



الشكل 7-4: واجهة إدارة الوجبات اليومية.

8-1-4 إدارة الإعلانات

تعرض هذه الواجهة سجلًا للإعلانات المنشورة داخل النظام، مع تفاصيل تتعلق بعنوان الإعلان، الفئة المستهدفة، وتاريخ النشر. تتيح أدوات الإدارة تنفيذ عمليات مثل التعديل أو الحذف بشكل مباشر، إلى جانب زر مخصص لإنشاء إعلان جديد. تُستخدم هذه الصفحة لتنظيم المحتوى الإعلاني وتوجيهه لفئات محددة ضمن الروضة بطريقة منظمة وسهلة التحكم.

الشكل 8-4: واجهة إدارة الإعلانات.

9-1-4 إدارة الفعاليات

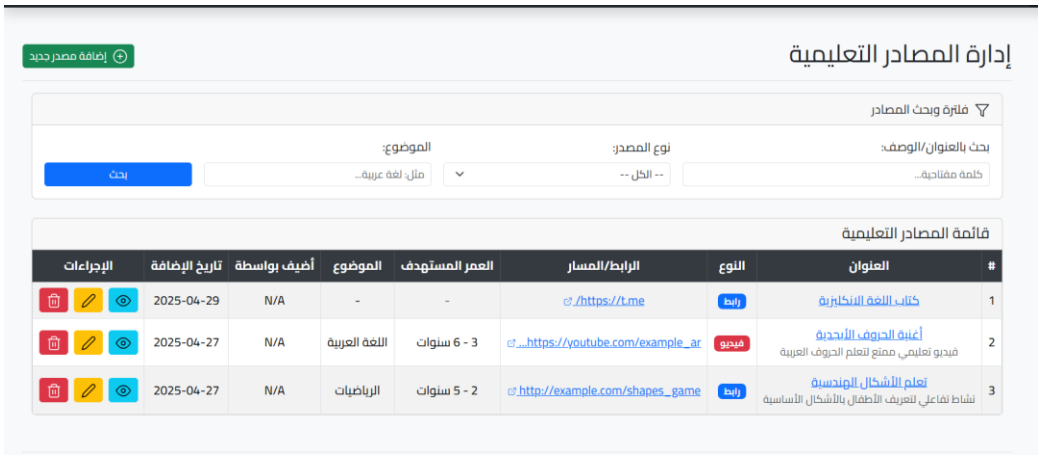
تُستخدم هذه الواجهة لمتابعة الفعاليات والرحلات المجدولة ضمن الروضة، حيث تُعرض قائمة تفصيلية تتضمن معلومات مثل الاسم، التوقيت، المكان، وعدد المشاركين. تتيح أدوات التحكم تنفيذ تعديلات سريعة أو حذف الفعالية، مع وجود زر مخصص لإضافة فعاليات جديدة. كما تشمل خيارات بحث لتسهيل الوصول إلى حدث معين حسب الاسم أو الموقع، مما يجعل تنظيم الأنشطة أمرًا مرنًا ودقيقًا ضمن النظام.

الشكل 9-4: إدارة الفعاليات والرحلات.

10-1-4 المصادر التعليمية

تعرض هذه الواجهة قائمة بالمصادر التعليمية المتوفرة داخل النظام، مع تصنيفات توضح نوع كل مصدر والموضوع والعمر المستهدف. تتيح للمستخدم إدارة المحتوى التعليمي من خلال أدوات سريعة للتحرير أو الحذف، بالإضافة إلى زر لإضافة مصادر جديدة. يدعم شريط البحث إمكانية

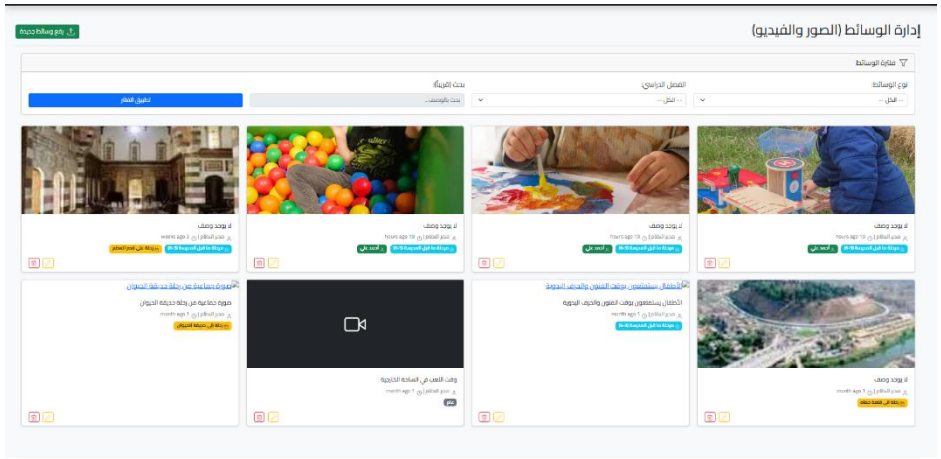
تصفية المحتوى وفق العنوان أو الكلمات المفتاحية، مما يسهم في تنظيم الموارد التعليمية وتسهيل الوصول إليها بحسب احتياجات الأطفال.



الشكل 10-4: واجهة إدارة المصادر التعليمية.

11-1-4 إدارة الوسائط

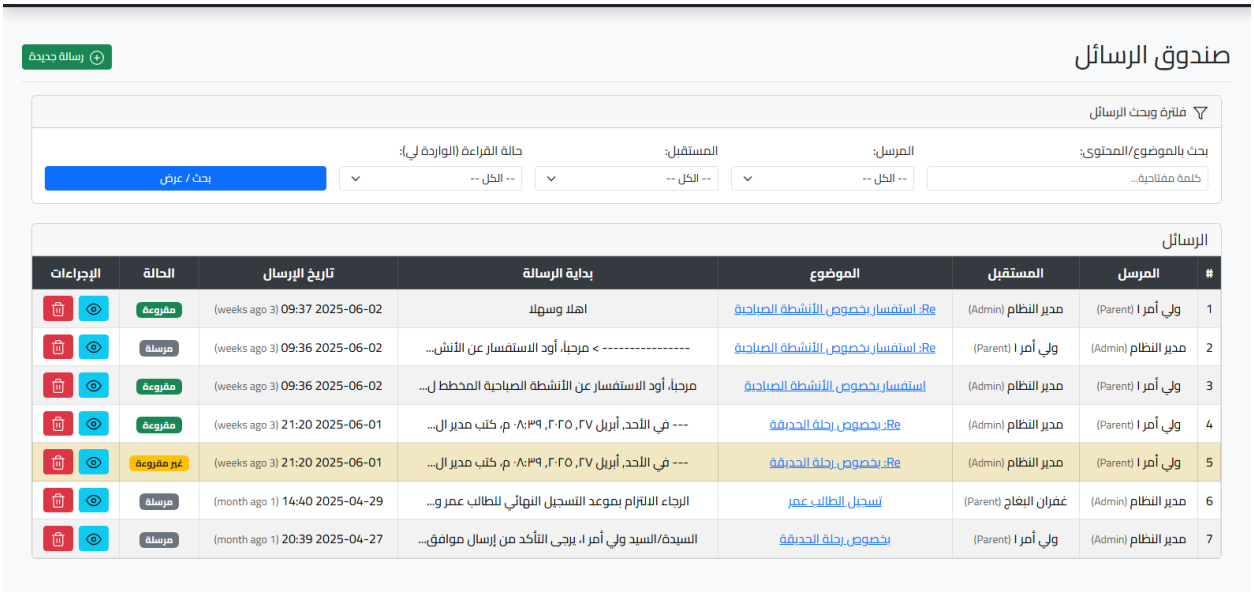
تعرض هذه الواجهة مجموعة الوسائط المرئية داخل النظام، مع تفاصيل مثل نوع الوسيلة، تاريخ الرفع، والفصل الدراسي المستهدف. تتيح إدارة الوسائط عبر أدوات سريعة للتعديل أو الحذف، وزر لإضافة وسائط جديدة، مع خاصية بحث متقدم لتسهيل التنظيم والوصول بحسب الفئة أو الوصف أو الفصل الدراسي.



الشكل 11-4: واجهة إدارة الوسائط.

4-1-12 الرسائل الواردة

تعرض هذه الواجهة صندوق الرسائل داخل النظام، مع قائمة بالرسائل الواردة والصادرة مع تفاصيل المرسل، المستلم، التاريخ، الموضوع، وحالة القراءة. تتيح إدارة الرسائل من خلال أدوات سريعة للحذف أو القراءة، وزر لإرسال رسالة جديدة، بالإضافة إلى بحث متقدم لتصفية الرسائل حسب الموضوع أو المرسل أو المحتوى.



الشكل 4-12: واجهة صندوق الرسائل.

4-1-13 ملاحظات أولياء الأمور

تعرض هذه الواجهة قائمة ملاحظات أولياء الأمور مع تفاصيل مقدم الملاحظة، الطفل المستهدف، التاريخ، ونص الملاحظة. تتيح إدارة الملاحظات عبر أدوات سريعة للحذف أو القراءة، بالإضافة إلى بحث متقدم لتصفية الملاحظات حسب النص أو ولي الأمر أو الطفل المستهدف.

الشكل 13-4: واجهة ملاحظات أولياء الامور.

2-4 واجهات التطبيق

يعرض هذا القسم مجموعة من الواجهات التفاعلية التي تم تطويرها ضمن التطبيق الإلكتروني المخصص لأولياء الأمور، بهدف تسهيل التواصل مع الروضة وتمكينهم من متابعة مختلف جوانب حياة أطفالهم التعليمية والصحية والترفيهية. الشكل (14-4) يُظهر القائمة الرئيسية للتطبيق، والتي تتضمن قائمة شاملة من الخيارات المهمة مثل: "الوجبات"، "الوسائط اليومية"، "الأنشطة اليومية"، "الحضور والغياب"، "قائمة الملاحظات"، "المؤشرات الصحية"، "السجل الصحي"، "المواد التعليمية"، "الفعاليات"، "الألعاب التعليمية"، و"تسجيل الخروج"، وهي مصممة لتقديم تجربة استخدام سهلة وشاملة لولي الأمر.

أما الشكل (15-4) فيوضح واجهة خاصة بعرض قائمة الفعاليات القادمة داخل الروضة، حيث تظهر معلومات الفعالية، توقيتها، والجهة المنظمة، مما يتيح متابعة دقيقة دون الحاجة لتسجيل الحضور. وفي السياق ذاته، يعرض الشكل (16-4) جدول الأنشطة الأسبوعية للأطفال، بحيث يتم

تنظيم الأنشطة لكل يوم من أيام الأسبوع، مع إظهار تفاصيل مثل الدروس الموسيقية وغيرها، مما يُساعد أولياء الأمور على تتبع الجدول الزمني للطفل.

أما الجوانب الصحية، فقد تم تمثيلها في الشكل (4-17) من خلال صفحة خاصة بمؤشرات الصحة، حيث تُعرض البيانات بشكل بصري عبر دوائر ملوّنة توضّح نسب الشرب، الأكل، الصحة النفسية، النشاط البدني، ومؤشرات الرؤية والسمع، لتقديم تقييم سريع لحالة الطفل. وفي الشكل (4-18)، تظهر صفحة الحساب الشخصي لأحد أولياء الأمور، والتي تتيح عرض المعلومات الشخصية مثل الاسم، البريد الإلكتروني، الهاتف، والعنوان، بالإضافة إلى خيارات إدارة الحساب مثل تعديل البيانات وتغيير كلمة السر.

أما التواصل بين الأطراف، فيُجسّده الشكل (4-19) من خلال صفحة "التواصل"، والتي تتضمن الرسائل المتبادلة بين أولياء الأمور ومديري النظام، مع إمكانية بدء مراسلة جديدة، مما يدعم التفاعل الفوري حول المواضيع المختلفة كالحضور والأنشطة. كما يستعرض الشكل (4-20) واجهة مخصصة للفصول الدراسية داخل الروضة، حيث تُعرض فئات مختلفة مثل "الفئة A" و"الفئة B"، مع إبراز نوعية الأنشطة المقدّمة عبر صور توضيحية تُمكّن ولي الأمر من التعرف على مضمون كل فئة.

أما الصفحة الرئيسية للتطبيق، فهي موضحة في الشكل (4-21)، حيث تتضمن تحية مخصصة لولي الأمر، إلى جانب إشعارات فورية حول الفعاليات، تحديثات الوجبات، والتنبيهات الطبية الدورية. بالنسبة للترفيه والتعلم التفاعلي، يستعرض الشكل (4-22) "معرض الألعاب للأطفال"، والذي

يضم ألعابًا تعليمية مثل "الذاكرة"، "التلوين"، "الأحرف"، و"الفقاعات"، لتعزيز مهارات الطفل بطريقة ممتعة.

الشكل 14-4: القائمة الرئيسية.

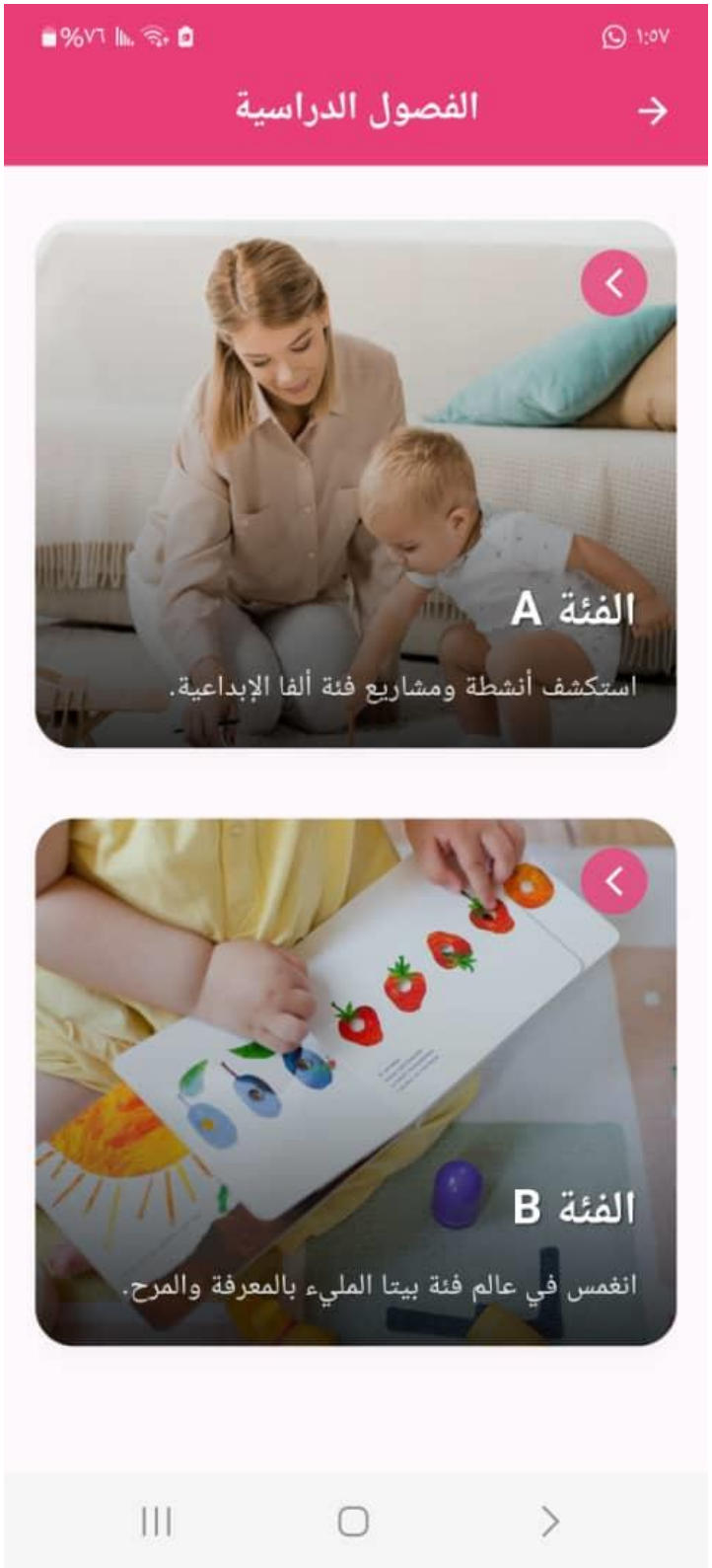
الشكل 15-4: الفعاليات القادمة.

الشكل 16-4: جدول الأنشطة الأسبوعي.

الشكل 17-4: مؤشرات الطالب الصحية.

الشكل 18-4: الحساب الشخصي لولي الأمر.

الشكل 19-4: صندوق الوارد.



الشكل 20-4: الفصول الدراسية.

الشكل 21-4: الصفحة الرئيسية.

الشكل 22-4: معرض الألعاب للأطفال.

3-4 النتائج

أسفرت نتائج مشروع "نظام إدارة رياض الأطفال" عن تطوير تطبيق رقمي متكامل يسهم في تحسين الكفاءة الإدارية والتربوية داخل مؤسسات رياض الأطفال، من خلال أتمتة العديد من المهام الروتينية التي كانت تُنجز يدويًا. وقد أتاح النظام للإدارة القدرة على تنظيم بيانات الأطفال ومتابعة الحضور، إلى جانب تتبع حالتهم الصحية والسلوكية، وتنظيم الفصول والوجبات والأنشطة. كما مكّن أولياء الأمور من متابعة تقدم أبنائهم لحظة بلحظة، والتفاعل مع إدارة الروضة بشكل مباشر من خلال إشعارات وتحديثات فورية. واعتمد المشروع على بنية تقنية حديثة باستخدام Flutter للواجهة الأمامية وLaravel للواجهة الخلفية، ما وفّر منصة موثوقة وسهلة الاستخدام. أظهر النظام استقرارًا جيدًا خلال الاختبارات، ونجح في معالجة البيانات بكفاءة مع توفير تجربة استخدام سلسة، مما يشير إلى إمكانية اعتماده فعليًا في بيئة رياض الأطفال. وتُعد هذه النتائج خطوة واعدة نحو رقمنة قطاع التعليم المبكر وتعزيز التواصل بين الأسرة والمؤسسة التعليمية.

5

الفصل الخامس: الخاتمة والتطويرات المستقبلية

لقد شكّل مشروع "نظام إدارة رياض الأطفال" تجربة تطبيقية رائدة في مجال تطوير الحلول البرمجية الموجهة لقطاع الطفولة المبكرة، من خلال الجمع بين التحديات الواقعية التي تواجه مؤسسات رياض الأطفال، والتقنيات الحديثة التي تتيح إمكانية تصميم أنظمة ذكية وفعّالة. إذ انطلق المشروع من فهم دقيق لاحتياجات البيئة التعليمية في هذه المؤسسات، خصوصًا من حيث الجوانب الإدارية والوظيفية التي تعتمد غالبًا على الأساليب التقليدية، مما ينتج عنه تأخير في معالجة البيانات، وصعوبة في إيصال المعلومات، وغياب قنوات التواصل الفورية بين الإدارة وأولياء الأمور، وهي تحديات أثبتت العديد من الدراسات التربوية أثرها السلبي على جودة التعليم والرعاية المقدّمة للأطفال في مرحلة ما قبل المدرسة.

في ضوء ذلك، قام هذا المشروع بوضع رؤية متكاملة لبناء نظام رقمي يُعنى بإدارة الجوانب التشغيلية والتربوية لرياض الأطفال، مستندًا إلى منهجية علمية شاملة تبدأ بتحليل الاحتياجات والمتطلبات، ثم تصميم قاعدة بيانات مركزية، وتطوير واجهات تفاعلية، وصولًا إلى مرحلة اختبار النظام وضمان توافقه مع مختلف المستخدمين. وقد تم اختيار تقنيات حديثة ومفتوحة المصدر لهذا الغرض، مثل إطار Flutter لبناء واجهات الهواتف المحمولة، وLaravel لتصميم البنية الخلفية للنظام، مدعومًا بـ MySQL لإدارة البيانات، ومفهوم RESTful API لضمان التكامل المرن مع أنظمة أو خدمات مستقبلية. وقد جاءت هذه الاختيارات التقنية بناءً على دراسة متعمقة لمزايا كل تقنية على مستوى الأداء، القابلية للتوسع، وسهولة التعلّم والصيانة.

من خلال تنفيذ هذا المشروع، تم تقديم حلول عملية تُعالج مختلف التحديات التي تواجه رياض الأطفال، بداية من تسجيل الحضور وتتبع الحالة الصحية للأطفال، وحتى تنظيم الجداول اليومية، وتوثيق الرحلات والفعاليات، وتوفير مكتبة تعليمية مصنفة، وكل ذلك ضمن منصة واحدة سهلة الاستخدام، آمنة، ومتجاوبة مع الهواتف المحمولة. وقد تم تزويد كل فئة من المستخدمين بواجهات مخصصة تلائم احتياجاتهم؛ حيث يستطيع أولياء الأمور من خلال حساباتهم مراقبة تقارير الأداء التربوي والصحي لأطفالهم، والاطلاع على التحديثات، والتفاعل مع إدارة الروضة؛ في حين تمكّن واجهة الإدارة من الإشراف على سير العمليات التربوية اليومية، وجدولة الفصول، وتسجيل ملاحظات وتحديثات مرتبطة بكل طفل.

وقد أظهرت نتائج التقييم الأولي للنظام فعالية عالية من حيث التفاعل، ومرونة في الاستخدام، وانخفاضًا ملحوظًا في الوقت اللازم لأداء المهام الإدارية اليومية، مقارنة بالطرق اليدوية التقليدية. كما لوحظ تحسّن في جودة التواصل بين الإدارة وأولياء الأمور، نتيجة تفعيل نظام الإشعارات والرسائل الفورية الذي يُبقي جميع الأطراف على اطلاع دائم بأحدث المستجدات. هذه المؤشرات تبرهن على أن دمج التكنولوجيا الذكية ضمن بيئة رياض الأطفال لا يسهم فقط في رفع كفاءة العمل، بل يعزز أيضًا من جودة الرعاية والتعليم في مرحلة الطفولة المبكرة، من خلال خلق بيئة أكثر تفاعلية وشفافية.

ومن ناحية أخرى، لا يخلو تنفيذ المشروع من مجموعة من التحديات، مثل الحاجة إلى ضمان أمان البيانات الخاصة بالأطفال، وتحقيق التوازن بين البساطة والوظائف المتقدمة، والتأكد من توافق النظام مع مختلف أنواع الأجهزة المحمولة، إضافة إلى صعوبات ربط الواجهة الأمامية بالخلفية. إلا أن هذه التحديات تم التعامل معها عبر خطوات مدروسة، تضمنت مراجعة بنية الكود، وتصحيح الأخطاء المنطقية أثناء الاختبار، وتحسين تجربة المستخدم عبر تصميم واجهات مرنة وبديهية. كما تم اعتماد أسلوب التوثيق باستخدام OpenAPI لتسهيل التوسعة المستقبلية للنظام، وربطه بخدمات تعليمية أو صحية أخرى.

ورغم أن النسخة الحالية من النظام تمثل منتجًا متكاملًا وقابلًا للاستخدام الفعلي ضمن بيئة محلية، إلا أن آفاق التطوير ما تزال واسعة، ويمكن استثمار الأساس القوي لهذا المشروع للارتقاء به إلى مستويات أعلى من التخصص والتكامل الذكي. ومن خلال المراجعة الفنية وتحليل ردود الفعل المحتملة للمستخدمين النهائيين، تم اقتراح مجموعة من التحسينات المستقبلية التي يمكن تنفيذها ضمن مراحل لاحقة من المشروع، أو اعتمادها كنواة لمشاريع تطويرية جديدة ذات صلة مباشرة.

وفيما يلي أبرز التطويرات المستقبلية المقترحة للنظام:

- دمج خاصية الدفع الإلكتروني لتمكين أولياء الأمور من تسديد الرسوم الدراسية والمصاريف الإضافية مباشرة عبر التطبيق بطريقة آمنة وسريعة.

- إضافة وحدة تحليل ذكية تعتمد على تقنيات الذكاء الاصطناعي لتفسير سلوك الأطفال ضمن الروضة، واقتراح تدخلات تعليمية أو صحية استباقية.
- تطوير تطبيق موازٍ مخصص لنظام iOS لضمان الوصول إلى شريحة أوسع من المستخدمين، مع الحفاظ على التوافق الوظيفي بين مختلف الأنظمة.
- إتاحة خاصية تصدير تقارير الأطفال بصيغ متعددة (PDF / Excel) بهدف استخدامها في التوثيق أو مشاركتها مع مختصين خارجيين عند الحاجة.
- توسيع مكتبة المحتوى التعليمي لتشمل ألعابًا تربوية تفاعلية تناسب مراحل عمرية مختلفة، مع إمكانية تقييم مدى استيعاب الطفل لها.
- تصميم نظام تنبيهات متقدم يعتمد على سلوك المستخدم وتفضيلاته، لتوفير إشعارات مخصصة حسب طبيعة كل ولي أمر أو إدارة.
- توفير دعم متعدد اللغات، خاصة اللغة الإنكليزية، لجعل النظام أكثر ملاءمة للاستخدام في مؤسسات دولية أو متعددة الجنسيات.
- إنشاء واجهات مخصصة للمعلمات تتيح لهن تسجيل ملاحظات يومية وتقييمات تعليمية بطريقة منظمة ومؤرشفة.
- ربط النظام بمنصات تقويم خارجية مثل Google Calendar لتوفير تنبيهات تلقائية حول الفعاليات والتطعيمات وجداول الوجبات.
- بناء وحدة ذكاء أعمال مصغّرة تقدم للإدارة تحليلات بيانية عن الحضور، النشاطات، والصحة العامة للأطفال لاتخاذ قرارات مستندة إلى البيانات.

وبناء على ما سبق، يُمكن القول إن المشروع لم يحقق فقط أهدافه الموضوعة بدقة، بل شكّل خطوة تأسيسية نحو مستقبل رقمنة مؤسسات التعليم المبكر في بيئتنا المحلية. كما يعكس مستوى النضج التقني والمعرفي الذي تم اكتسابه من قبل الفريق المنفذ، ويفتح المجال أمام مبادرات أوسع تتبنى الذكاء الاصطناعي، إنترنت الأشياء، وتحليلات البيانات الكبرى في خدمة الطفولة. ومن هنا فإن مشروع "نظام إدارة رياض الأطفال" لا يجب أن يُنظر إليه على أنه منتج

نهائي، بل كنواة ديناميكية قابلة للتوسع، ومثال واقعي على قدرة الحلول البرمجية المحلية على التنافس في تقديم خدمات تعليمية ذات جودة عالمية.

التعريف	المصطلح باللغة العربية	The term in English
نظام رقمي يهدف إلى تنظيم العمليات الإدارية والتربوية داخل رياض الأطفال.	نظام إدارة رياض الأطفال	Kindergarten Management System
بيئة تطوير توفر أدوات وقوالب لبناء التطبيقات بسرعة وكفاءة.	إطار العمل	Framework
إطار عمل مفتوح المصدر من Google لتطوير تطبيقات متعددة المنصات باستخدام لغة Dart.	فلاتر	Flutter
إطار عمل PHP لتطوير تطبيقات الويب باستخدام نمط MVC.	لارافيل	Laravel
نمط تصميم يقسم النظام إلى ثلاثة أجزاء: النموذج، العرض، المتحكم.	نمط MVC	Model – View – Controller
نظام لتخزين البيانات بشكل منظم لسهولة إدارتها واسترجاعها.	قاعدة البيانات	Database
واجهة برمجية تتيح التواصل بين الأنظمة باستخدام بروتوكولات HTTP.	واجهة برمجة التطبيقات وفق نمط REST	RESTful Application Programming Interface
مكتبة تابعة للارافيل للتعامل مع قواعد البيانات بطريقة كائنية.	Eloquent ORM	Eloquent ORM

Dart Language	لغة Dart	لغة برمجة مطورة من Google تُستخدم مع Flutter لتطوير التطبيقات.
User Interface (UI)	واجهة المستخدم	الجزء الذي يتفاعل معه المستخدم في التطبيق.
Usability	قابلية المستخدم	مدى سهولة وفعالية استخدام النظام من قبل المستخدمين.
Scalability	قابلية التوسع	قدرة النظام على التكيف مع زيادة الحمل أو عدد المستخدمين.
Authentication	المصادقة	التحقق من هوية المستخدم للسماح له بالدخول إلى النظام.
Authorization	التفويض	تحديد صلاحيات المستخدم داخل النظام.
Parental Communication	التفاعل مع أولياء الأمور	عملية التواصل الرقمي بين الروضة وأولياء الأمور.
Multimedia Content	محتوى متعدد الوسائط	مواد تعليمية تحتوي على صور، فيديو، وصوت.

Object-Oriented Programming	برمجة كائنية	نمط برمجي يعتمد على الكائنات والصفوف في تنظيم التعليمات البرمجية.
API Library	مكتبة واجهات برمجية	مجموعة من الوظائف التي تتيح للتطبيقات التفاعل مع خدمات أخرى.

المراجع

[1]: حامد، ن.م.، & عبده، ن.ع.أ. (2023). التحول الرقمي في مؤسسات رياض الأطفال بمصر على ضوء منهج 2.0: تصور مقترح للتفعيل. مجلة التعليم والعلوم التربوية، 14(1)، 1–70.
https://doi.org/10.21608/tessj.2023.306405

[2]: غندورة، ر.ع. (2021). درجة تضمين مناهج رياض الأطفال في السعودية لبعض مفاهيم المواطنة الرقمية. مجلة العلوم التربوية والنفسية، 5(39)، 23–48.
https://doi.org/10.26389/AJSRP.G280421

[3]: Erdreich, L. (2021). Managing parent capital: Parent-teacher digital communication among early childhood educators. Italian Journal of Sociology of Education, 13(1), 135–159. https://doi.org/10.14658/pupj-ijse-2021-1-6

[4]: Santos Urbina, S., Ferrer-Ribot, M., & Villatoro Moral, S. (2025). School-Family Communication in Early Childhood Education Through Digital Tools. International Journal of Early Childhood, (March), Article 000.
https://doi.org/10.1007/s13158-025-00419-3

[5]: Abdulkadir, M. (2019). Kindergarten management information system: A case of Kaafi Kindergarten in Borama [Master's thesis, University of Borama]. Academia.edu.

https://www.academia.edu/39755679/KINDERGARTEN_MANAGEMENT_INFORMATION_SYSTEM_A_Case_of_Kaafi_Kindergarten_in_Borama

[6]: Brightwheel. (n.d.). The leading platform for early education. Retrieved June 12, 2025, from https://mybrightwheel.com

[7]: Kinderpedia. (n.d.). Smart digital solution for schools and kindergartens. Retrieved June 12, 2025, from https://www.kinderpedia.co

[8]: Illumine. (n.d.). Preschool & childcare management software. Retrieved June 12, 2025, from https://www.illumine.app

[9]: Flutter. (n.d.). Architectural overview. Flutter Documentation. Retrieved June 12, 2025, from https://docs.flutter.dev/resources/architectural-overview#architectural-layers

[10]: Flutter. (n.d.). Showcase. Flutter Documentation. Retrieved June 12, 2025, from https://flutter.dev/showcase

[11]: Laravel. (n.d.). Laravel documentation (Version 11.x). Retrieved June 12, 2025, from https://laravel.com/docs/11.x/readme

[12]: Hanif, F. (2023, June 26). Understanding the MVC architecture in Laravel: A comprehensive guide. Medium. https://fkrihnif.medium.com/understanding-the-mvc-architecture-in-laravel-a-comprehensive-guide-8f620cc139b6:contentReference[oaicite:4]{index=4}

[13]: Bootstrap. (n.d.). Bootstrap: The most popular HTML, CSS, and JS library in the world. Retrieved from https://getbootstrap.com/

Syrian Arab Republic

Ministry of Higher Education and Scientific Research

Al-Wataniya Private University

Faculty of Engineering

Department of Computer Engineering

A Graduation Project Submitted for the Degree of Bachelor in Computer Engineering Entitled:

Kindergarten Management System

Prepared by:

Khadija Al-Akkad

Ghofran Al-Baghaj

Supervisor:

Dr. Eng. Hani Al-Khatib

2024 – 2025