الجمهورية اليمنية
جامعة العلوم والتكنولوجيا
كلية الحاسبات وتقنية المعلومات
قسم تقنية المعلومات

Republic of Yemen
University of Science and Technology
College of Computing
Department of Information Technology

مساعد ذكي مؤسسي معتمد على الذكاء الاصطناعي وتحليل البيانات

Smart Institutional Assistant Based on Artificial Intelligence and Data Analytics

إعداد الطلاب:

وليد محسن يحيى العنسي
محمد رشيد أحمد الشريف
أحمد علي محمد المتوكل
عبدالكريم قاسم عبده العبدلي
محمد نجيب العواضي

تحت إشراف: د. نبيل المخلافي

قُدِّم هذا المشروع استكمالًا لمتطلبات نيل درجة البكالوريوس في تقنية المعلومات، بكلية الحاسبات وتقنية المعلومات، للعام الجامعي 1448هـ الموافق 2026--2027م.

الملخص

سيتم كتابة الملخص عند الانتهاء من جميع فصول الأطروحة.

Abstract

The abstract will be written upon completion of all thesis chapters.

Authorization

Faculty Technology, and Science of University authorize hereby students, undersigned the We, report, project graduation our of copies supply to Technology, Information and Computing of titled

``Smart Institutional Assistant Based on Artificial Intelligence and Data Analytics''

requirements the of fuflillment in request, upon individuals interested or organizations, libraries, to Technology. Information in degree Bachelor's the for

Signature	No. Student	Name Student
.....	202310100941	Al-Ansi Yahya Mohsin Walid
.....	202310100869	Ahmed Rasheed Mohammed Al-Sharfi
.....	202310101668	Mohammed Ali Ahmed Al-Mutawakel
.....	202310101670	Al-Abdali Abdo Qassem AbdulKarim
.....	202310101207	Al-Awadi Najeeb Mohammed

2027 / 2026 / ..... / ..... **Date:**

الإهداء

الحمد لله الذي أنشأ وبرأ، وخلق الماء والثرى، وأبدع كل شيء وذرأ، الرحمن على العرش استوى، وله الحمد في الأولى والآخرة، وهو على كل شيء قدير. والصلاة والسلام على سيدنا محمد، المبعوث رحمةً للعالمين، وعلى آله الطيبين الطاهرين، وصحبه أجمعين.

أما بعد؛ فلم نطلب العلم ونتسلح بمعارف عصرنا إلا ابتغاء وجه الله تعالى، ورغبةً في أن نجعل ما تعلمناه نافعًا لديننا، وخادمًا لوطننا، ومُسهمًا في نهضته وتقدمه، إيمانًا منا بأن الأمم لا تُبنى إلا بالعلم، ولا تُصان أوطانها إلا بسواعد أبنائها وعقولهم.

قال تعالى: ﴿يَرْفَعِ اللَّهُ الَّذِينَ آمَنُوا مِنْكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ﴾، ونسأل الله سبحانه أن يجعل سعينا خالصًا لوجهه الكريم، وأن يبارك في علمنا وعملنا، وأن ينفع بنا، وأن يرزقنا الإخلاص والإتقان، وأن يكون هذا العمل سببًا في إدخال السرور على والدينا وأساتذتنا، وجزاءً يسيرًا لما بذلوه في تعليمنا وتوجيهنا.

اللهم علِّمنا ما ينفعنا، وانفعنا بما علَّمتنا، وزدنا علمًا، واجعل علمنا حجةً لنا لا علينا، ووفِّقنا لما تحب وترضى.

كلمة شكر وتقدير

الحمد لله الذي بنعمته تتم الصالحات، وله الحمد أولًا وآخرًا، وظاهرًا وباطنًا، أن وفقنا وأعاننا على إتمام هذا العمل، فما كان فيه من صواب فمن فضله وتوفيقه، وما كان فيه من نقص فنسأله سبحانه العفو والتوفيق.

وانطلاقًا من قول النبي ﷺ: «لا يشكر الله من لا يشكر الناس»، فإننا نتقدم، نحن أعضاء فريق المشروع، بخالص الشكر وعظيم الامتنان إلى كل من أسهم في إنجاز هذا العمل، وقدم لنا علمًا أو توجيهًا أو دعمًا كان له الأثر في مسيرتنا العلمية.

ونخص بجزيل الشكر والتقدير مشرف مشروعنا **الدكتور نبيل المخلافي**، على ما أولانا من اهتمام، وما بذله من وقتٍ وجهد، وما قدمه من توجيهاتٍ علميةٍ قيّمة وملاحظاتٍ بنّاءة، كان لها بالغ الأثر في تطوير هذا المشروع وإخراجه بالصورة التي نرجو أن تليق.

كما نتقدم بخالص الشكر والتقدير إلى عميد كلية الحاسبات وتقنية المعلومات **الدكتور بلال الفهيدي**، لما يوليه من اهتمام بدعم البيئة الأكاديمية وتشجيع البحث العلمي، وإلى رئيس قسم تقنية المعلومات **الدكتور وليد شاهر**، على متابعته وحرصه الدائم على تهيئة البيئة العلمية المناسبة للطلاب، فلهما منا جزيل الشكر والتقدير.

ونتقدم كذلك بالشكر والعرفان إلى جميع أعضاء هيئة التدريس بالكلية، الذين غرسوا فينا حب العلم، وأرشدونا إلى طريق المعرفة، وكان لهم بعد الله عظيم الأثر في بناء شخصياتنا العلمية والعملية.

ولا يسعنا إلا أن نعبر عن بالغ امتناننا لوالدينا وأسرنا الكريمة، الذين كانوا خير سندٍ وعون، بدعائهم، وصبرهم، وتشجيعهم المستمر، فنسأل الله أن يجزيهم عنا خير الجزاء، وأن يبارك في أعمارهم وأعمالهم.

كما نشكر كل من قدم لنا يد العون أو المشورة أو الدعم، وأسهم في تذليل الصعوبات التي واجهتنا خلال مراحل تنفيذ هذا المشروع، سائلين الله أن يجعل ذلك في موازين حسناتهم.

ونسأل الله تعالى أن يجعل هذا العمل خالصًا لوجهه الكريم، وأن ينفع به، وأن يجعله لبنةً نافعة في خدمة العلم والمجتمع، وأن يوفقنا لمواصلة مسيرة العطاء والبحث والابتكار، إنه ولي ذلك والقادر عليه.

تصريح الطلاب

نُقِرّ نحن الطلاب الموقعين أدناه بأن هذا المشروع هو نتاج جهدنا الشخصي، وقد تم بإشراف وتوجيه من مشرفنا الأكاديمي. يعكس هذا العمل ما توصلنا إليه من خلال دراستنا وبحثنا. جميع النصوص والنتائج المأخوذة من مصادر أخرى قد تم توثيقها والإشارة إليها بشكل كامل.

اسم الطالب	التوقيع
وليد محسن يحيى العنسي	....
محمد رشيد أحمد الشريف	....
أحمد علي محمد حسن المتوكل	....
عبد الكريم قاسم عبده مسعود العبدلي	....
محمد نجيب العواضي	....

المشرف: د. نبيل المخلافي

التاريخ: ..... / ..... / 2026 / 2027

إقرار المشرف

نشهد بأن إعداد هذا المشروع بعنوان:

«مساعد ذكي مؤسسي معتمد على الذكاء الاصطناعي وتحليل البيانات»

من إعداد الطلاب المذكورة أسماؤهم أعلاه، قد تم تحت إشرافي، استكمالاً لمتطلبات الحصول على درجة البكالوريوس في تقنية المعلومات بقسم تقنية المعلومات.

اسم المشرف: د. نبيل المخلافي

التوقيع: .....

التاريخ: ..... / ..... / 2026 / 2027

تأكيد

عنوان المشروع: مساعد ذكي مؤسسي معتمد على الذكاء الاصطناعي وتحليل البيانات

أعد وقدّم هذا المشروع كل من:

الرقم الجامعي	اسم الطالب
202310100941	وليد محسن يحيى العنسي
202310100869	محمد رشيد أحمد الشريف
202310101668	أحمد علي محمد حسن المتوكل
202310101670	عبد الكريم قاسم عبده مسعود العبدلي
202310101207	محمد نجيب عبدالله العواضي

تم اعتماد هذا المشروع من حيث النطاق والجودة والعرض التقديمي كمشروع تخرج.

قدم هذا التقرير استكمالاً لمتطلبات الحصول على درجة البكالوريوس في تقنية المعلومات.

كلية الحاسبات وتقنية المعلومات
جامعة العلوم والتكنولوجيا، اليمن

المشرف: د. نبيل المخلافي

التوقيع: .....

التاريخ: ..... / ..... / 2026 / 2027

لجنة الكشف والمناقشة

عنوان المشروع: مساعد ذكي مؤسسي معتمد على الذكاء الاصطناعي وتحليل البيانات

المشرف:

م	الاسم	الصفة	التوقيع
1	د. نبيل المخلافي	مشرف المشروع	.....

لجنة المناقشة:

م	الاسم	الصفة	التوقيع
1	.....	عضو داخلي	.....
2	.....	عضو داخلي	.....
3	.....	عضو خارجي	.....

ملاحظات اللجنة:

رئيس القسم: .....

التاريخ: ..... / ..... / 2026 / 2027

المحتويات

الملخص i

Abstract ii

Authorization iii

الإهداء iv

كلمة شكر وتقدير v

تصريح الطلاب vi

إقرار المشرف vii

تأكيد viii

لجنة الكشف والمناقشة ix

قائمة الاختصارات xviii

1 المقدمة 1

1.1 الخلفية (Background) 2

2.1 أهداف المشروع (Objectives) 3

3.1 أهمية المشروع (Significance) 3

4.1 مساهمات المشروع (Project Contributions) 4

5.1 مشكلة المشروع (Problem Statement) 4

6.1 تعريف المشروع (System Definition) 5

7.1 نطاق المشروع (Project Scope) 6
1.7.1 النطاق الجغرافي (Geographical Scope) 6
2.7.1 النطاق الوظيفي (Functional Scope) 6
3.7.1 أصحاب المصلحة (Stakeholders) 6
8.1 قيود المشروع (Limitations) 7
9.1 المنهجية المستخدمة في مشروع بيان (منهجية أجايل) 7
1.9.1 مرحلة التخطيط (Planning) 8
2.9.1 مرحلة التصميم (Design) 8
3.9.1 مرحلة التطوير (Development) 8
4.9.1 مرحلة الاختبار (Testing) 8
5.9.1 مرحلة الإطلاق (Deployment) 8
6.9.1 مرحلة المراجعة (Review) 8
10.1 الجدول الزمني للمشروع (Project Schedule) 9
11.1 متطلبات الأدوات (Required Tools) 10
1.11.1 الأدوات المادية (Hardware Tools) 10
2.11.1 الأدوات البرمجية (Software Tools) 12
12.1 هيكل المشروع (Project Structure) 13

2 الخلفية النظرية ومراجعة الأدبيات 14
1.2 مقدمة الفصل 15
2.2 الخلفية النظرية 15
1.2.2 النماذج اللغوية الكبيرة (Large Language Models - LLMs) 15
2.2.2 تقنية تحويل اللغة الطبيعية إلى SQL (NL2SQL) 16
3.2.2 تقنية الاسترجاع المعزز بالتوليد (RAG) 18
4.2.2 الأنظمة متعددة الوكلاء (Multi-Agent Systems) 19
5.2.2 تحديات معالجة اللغة العربية (Arabic NLP) 20
3.2 الأنظمة والدراسات السابقة 21
1.3.2 الأنظمة في مجال تحويل اللغة الطبيعية إلى SQL 21
2.3.2 الأنظمة في مجال الاسترجاع المعزز بالتوليد (RAG) 21
3.3.2 الأنظمة في مجال الأنظمة متعددة الوكلاء 22

4.3.2 المساعدات الذكية في التعليم العالي 22
5.3.2 الدراسات العربية 22
4.2 مقارنة الأنظمة السابقة 23
5.2 الفجوة البحثية 23
6.2 الخلاصة 26
1.6.2 معايير تقييم النظام (Evaluation Metrics) 26

3 المنهجية وتحليل النظام 28
1.3 منهجية التطوير 29
1.1.3 دورة التطوير الرشيقة 29
2.1.3 الدورات التنفيذية للمشروع 29
3.1.3 ملخص الدورات ومخرجاتها 32
2.3 دراسة الجدوى 33
3.3 تحليل المتطلبات 33
1.3.3 منهجية جمع المتطلبات 33
2.3.3 أصحاب المصلحة والمستخدمون المستهدفون 34
3.3.3 وصف النظام المقترح 34
4.3.3 المتطلبات الوظيفية 35
5.3.3 المتطلبات غير الوظيفية 37
6.3.3 متطلبات بيئة التشغيل التقنية 39
7.3.3 قيود ومحددات النظام 41
8.3.3 افتراضات النظام 41
9.3.3 خلاصة تحليل المتطلبات 42
4.3 حالات الاستخدام 42
5.3 تحليل المخاطر 42
6.3 ملخص الفصل 42

4 تصميم النظام 43
1.4 المعمارية العامة للنظام 44
2.4 تصميم قواعد البيانات 44

3.4 تصميم واجهات المستخدم 44
4.4 مخططات النظام 44
1.4.4 مخططات التدفق 44
2.4.4 مخططات التتابع 44
3.4.4 مخططات ER 44
5.4 تصميم واجهات برمجة التطبيقات (APIs) 44
6.4 ملخص الفصل 44

5 تنفيذ واختبار النظام 45

1.5 بيئة التطوير والأدوات 46
2.5 تنفيذ النظام 46
3.5 النشر والتشغيل 46
4.5 اختبار النظام 46
1.4.5 اختبارات الوحدة (Unit Tests) 46
2.4.5 اختبارات التكامل (Integration Tests) 46
3.4.5 اختبارات الأداء 46
4.4.5 اختبار قبول المستخدم (UAT) 46
5.5 الخطة الزمنية للتنفيذ 46
6.5 ملخص الفصل 47

6 النتائج والمناقشة 48

1.6 النتائج 49
1.1.6 النتائج الكمية 49
2.1.6 النتائج النوعية 49
2.6 واجهات النظام النهائية 49
3.6 مناقشة النتائج 49
4.6 التحديات والحلول 49
5.6 الدروس المستفادة 49
6.6 ملخص الفصل 49

7 الخاتمة والأعمال المستقبلية 50

1.7 الخاتمة 51
2.7 ملخص تحقيق الأهداف 51
3.7 مساهمات المشروع 51
4.7 قيود النظام 51
5.7 التوصيات 51
6.7 الأعمال المستقبلية 51
7.7 ملخص الفصل 51

ملحق أ: قاموس المصطلحات 52

ملحق ب: قائمة المتطلبات الكاملة 53

ملحق ج: كود مصدري مختار 54

ملحق د: لقطات شاشة إضافية 55

ملحق هـ: استبيان تقييم المستخدمين 56

ملحق و: توزيع أدوار الفريق 57

المراجع 58

قائمة الأشكال

1-1 نموذج أجايل (Agile Model) المتبع في مشروع بيان. 7
2-1 مخطط جانت لمراحل المشروع — الجزء الأول. 9
3-1 مخطط جانت لمراحل المشروع — الجزء الثاني. 9
4-1 مخطط جانت لمراحل المشروع — الجزء الثالث. 10
5-1 مخطط جانت لمراحل المشروع — الجزء الرابع. 10
1-2 مخطط تدفق تقنية RAG في نظام بيان. 18
2-2 معمارية النظام متعدد الوكلاء القائم على LangGraph. 20
1-3 خارطة الدورات التطويرية الأربع المعتمدة في مشروع «بيان». 31
2-3 توزيع المتطلبات الوظيفية لنظام «بيان» حسب الأولوية. 35
3-3 توزيع المتطلبات غير الوظيفية حسب خصائص الجودة في معيار ISO/IEC 25010. 37
4-3 معمارية نشر نظام «بيان» عبر Docker Compose (إطار مؤقت). 40

قائمة الجداول

1-1 الأدوات المادية المطلوبة للمشروع. 11

2-1 الأدوات البرمجية المعتمدة في تطوير مشروع بيان. 13

1-2 مقارنة النماذج اللغوية المرشّحة لمشروع بيان. 16

2-2 مقارنة شاملة بين الأنظمة السابقة ومشروع بيان. 24

3-2 ملخص الفجوة البحثية ومساهمة مشروع بيان في سدها. 25

1-3 مراحل دورة التطوير الرشيقة المطبقة في مشروع «بيان». 30

2-3 الملخص التنفيذي للدورات الأربع في منهجية تطوير نظام «بيان». 33

3-3 أصحاب المصلحة في نظام «بيان». 34

4-3 المتطلبات الوظيفية لنظام «بيان». 35

5-3 المتطلبات غير الوظيفية وطرق اختبارها. 38

6-3 متطلبات الخادم المادية. 39

7-3 متطلبات بيئة التشغيل البرمجية. 40

8-3 قيود ومحددات نظام «بيان». 41

9-3 افتراضات نظام «بيان». 42

قائمة الاختصارات

الاختصار	التعريف
API	واجهة برمجة التطبيقات التي تتيح التواصل بين الأنظمة المختلفة.
BIRD	معيار قياسي لتقييم أنظمة NL2SQL على قواعد بيانات واقعية.
CSV	صيغة ملف نصي لتخزين البيانات الجدولية مفصولة بفواصل.
ERD	مخطط العلاقات بين الكيانات في قاعدة البيانات (Entity-Relationship Diagram).
GPU	معالج الرسوميات المستخدم لتسريع معالجة النماذج.
LangGraph	مكتبة لبناء وتنسيق وكلاء الذكاء الاصطناعي المتعددين.
LLM	النموذج اللغوي الكبير (Large Language Model).
NL2SQL	تحويل اللغة الطبيعية إلى استعلامات SQL (Natural Language to SQL).
Odoo	نظام ERP مفتوح المصدر، يُستخدم كبيئة محاكاة جامعية.
Onboarding	عملية الإعداد والتهيئة الأولية للنظام.
pgvector	امتداد PostgreSQL لدعم تخزين والبحث في المتجهات.
Qdrant	قاعدة بيانات متجهة مفتوحة المصدر.
RAG	تقنية الاسترجاع المعزز بالتوليد (البحث في المستندات ثم توليد إجابة اعتمادًا على نتائج البحث).
RTL	الكتابة من اليمين إلى اليسار — دعم اللغة العربية.
SaaS	البرمجية كخدمة تُقدَّم عبر الإنترنت دون الحاجة لتثبيتها محليًا.
Schema Linking	تقنية تصفية الجداول والأعمدة ذات الصلة قبل توليد SQL.
Spider	معيار قياسي لتقييم أنظمة NL2SQL.

الاختصار	التعريف
SQL	لغة الاستعلام البنيوية المستخدمة للتعامل مع قواعد البيانات.
UML	لغة النمذجة الموحدة (Unified Modeling Language).
Views	طبقة وسيطة مبسطة فوق قاعدة البيانات لرفع دقة NL2SQL.

الفصل

الأوّل

المقدمة

1.1 الخلفية (Background)

شهد العالم خلال السنوات الأخيرة تحولًا كبيرًا في طريقة إدارة المؤسسات، حيث أصبح الاعتماد على الأنظمة الرقمية والذكاء الاصطناعي عاملًا رئيسيًا في تحسين جودة الخدمات المقدمة، وتعزيز كفاءة التشغيل، وتقليل الجهد اليدوي. وفقاً لتقرير McKinsey Global Institute لعام 2024، من المتوقع أن يُسهم الذكاء الاصطناعي بخلق قيمة اقتصادية تتراوح بين 17 و25 تريليون دولار سنوياً بحلول عام 2030، يُخصص منها نحو 40% لتطبيقات المؤسسات والإدارة. ومع النمو المتسارع في حجم البيانات والمستندات المؤسسية — الذي يُقدر بـ 175 زيتابايت عالمياً بحلول 2025 وفق توقعات IDC — وازدياد عدد المستخدمين والخدمات المرتبطة بها، أصبحت إدارة المعلومات أكثر تعقيدًا من أي وقت مضى، ولم تعد الأساليب التقليدية قادرة على مواكبة حجم العمل ومتطلبات المستخدمين.

في المؤسسات الحديثة، تتنوع المهام اليومية التي تتولى الإدارة الإشراف عليها، بدءًا من استعلامات البيانات وتحليلها، مرورًا بالبحث في المستندات واللوائح، ووصولاً إلى إعداد التقارير واتخاذ القرارات الاستراتيجية. هذه المهام تحتاج إلى نظام مركزي موثوق يضمن تنظيمها وتوثيقها، ويمنع ضياع المعلومات أو تداخلها، خاصة مع اعتماد المستخدمين المتزايد على الخدمات السريعة والموثوقة.

ورغم هذا التطور العالمي، ما زالت الكثير من المؤسسات في اليمن وفي العديد من الدول العربية تُدار بطرق تقليدية تُعرقل سير العمل وتؤثر على كفاءة اتخاذ القرار. ما يزال الاعتماد على الاستعلامات المباشرة لقواعد البيانات من قبل متخصصين شائعًا، وكذلك البحث اليدوي في المستندات غير المفهرسة، الأمر الذي يفتقر إلى السرعة والدقة ويؤدي إلى ضياع الوقت أو تكرار الأخطاء. كما أن إعداد التقارير والتحليلات تقتصر غالبًا على الجهد اليدوي دون وجود أي أدوات تحليلية تمكّن المستخدم من فهم اتجاهات البيانات أو اتخاذ قرارات مبنية على المعلومات.

ومع غياب نظام موحّد يجمع كل هذه الوظائف، تصبح عملية إدارة المعلومات مرهقة وغير فعّالة، ويعاني المستخدم من نقص الشفافية وضعف التواصل مع إدارة المؤسسة. كما تواجه الإدارة تحديات مستمرة في متابعة استعلامات المستخدمين، والتحقق من دقة المعلومات، وتحديد احتياجات التدريب، وإصدار تقارير تساعد في تحسين مستوى الخدمة.

من جهة أخرى، أدى التقدم في مجال الذكاء الاصطناعي وتعلم الآلة إلى فتح الباب أمام حلول رقمية جديدة في مجال إدارة المؤسسات، سواء عبر المساعدات الذكية أو أنظمة المحادثة المتقدمة، مما جعل عملية التحول الرقمي أكثر قابلية للتطبيق حتى في البيئات محدودة الموارد. هذه الحلول أصبحت اليوم ضرورة وليس خيارًا، خاصة في ظل احتياج المستخدمين إلى خدمات أسرع وأكثر دقة وأعلى جودة.

وبناءً على ذلك، نشأت الحاجة إلى تطوير نظام شامل لإدارة المؤسسات يربط بين المستخدمين والإدارة في إطار واحد، ويتيح تبادل المعلومات بشكل فوري، ويعزز الشفافية، ويوفّر سجلًا موحدًا لكل العمليات اليومية. ومن هنا جاء مشروع «المساعد الذكي المؤسسي (بيان)» ليقدّم نظامًا ذكيًا مصممًا خصيصًا لمعالجة التحديات التي تواجه المؤسسات في الواقع المحلي، مع مراعاة سهولة الاستخدام، وقابلية التوسع، والقدرة على تقديم تجربة استخدام احترافية تجمع بين البساطة والفعالية.

يقدم نظام «بيان» خطوة مهمة نحو التحول الرقمي في إدارة المؤسسات، حيث يجمع بين أحدث تقنيات الذكاء الاصطناعي وتطوير التطبيقات وبين فهم عميق لاحتياجات المستخدمين والإدارة، مما يجعله حلًا مبتكرًا وعمليًا يمكن تطبيقه مباشرة في المؤسسات، ويسهم في رفع مستوى الخدمات وتحسين كفاءة العمل واتخاذ القرارات.

2.1 أهداف المشروع (Objectives)

يهدف مشروع «بيان» إلى تطوير نظام مؤسسي متكامل مدعوم بتقنيات الذكاء الاصطناعي وتحليل البيانات، لتمكين المؤسسات من الوصول الذكي إلى بياناتها ومستنداتها باللغة العربية الطبيعية، وتحويل الأسئلة والمستندات إلى إجابات وتقارير دقيقة وفورية. ويتفرع من هذا الهدف الرئيسي مجموعة من الأهداف التفصيلية الآتية:

- **بناء مساعد محادثة ذكي باللغة العربية:** التفاعل مع المستخدمين باللغة العربية الطبيعية (الفصحى والعامية).
- **تمكين الاستعلام الذكي عن البيانات** (NL2SQL): تحويل الاستعلامات الصادرة باللغة الطبيعية إلى استعلامات قواعد بيانات (SQL) دون حاجة المستخدم لمعرفة تقنية مسبقة.
- **تطوير نظام استرجاع المستندات** (RAG): البحث في المستندات واللوائح المؤسسية وتوليد إجابات دقيقة وموثقة بمصادرها.
- **توفير أدوات التحليل الذكي:** تحليل البيانات واستخراج الاتجاهات والقيم الشاذة وتقديم رؤى إحصائية تدعم اتخاذ القرار.
- **توليد التقارير المنسقة:** إتاحة إمكانية إنتاج تقارير متعددة الأشكال (نصية، جدولية، ورسوم بيانية) قابلة للتصدير بصيغ مختلفة.
- **ضمان الشفافية والرقابة:** توفير سجل موحد (Audit Log) لتتبع الاستعلامات والإجابات وسجلات الوصول تعزيزًا للمساءلة.
- **تحسين التواصل المؤسسي:** تعزيز التفاعل بين المستخدمين وإدارة المؤسسة عبر نظام إشعارات وتنبيهات داخلية موحدة.

3.1 أهمية المشروع (Significance)

تنبع أهمية مشروع «بيان» من كونه يستهدف سد فجوة حقيقية في البيئة المؤسسية العربية عامةً، واليمنية خاصةً، تتمثل في غياب الأنظمة الذكية الموحّدة التي تجمع بين الاستعلام عن البيانات والبحث في المستندات وتحليلها في إطار واحد. فالأنظمة التجارية المتاحة تُعرَض بأسعار مرتفعة وتتطلب بنية تحتية متقدمة، كما أنها لا تدعم اللغة العربية بالشكل المطلوب، مما يُقصي غالبية المؤسسات المحلية عن الاستفادة من هذه التقنيات.

على الصعيد التشغيلي، يُتوقع أن يسهم النظام في تسريع عملية اتخاذ القرار من خلال تقصير زمن الاستعلام عن المعلومة من دقائق أو ساعات (في الوضع اليدوي التقليدي) إلى ثوانٍ معدودة. كما يُتوقع أن يُقلّل من الأخطاء البشرية الناتجة عن التعامل اليدوي مع البيانات، ويرفع من مستوى الشفافية عبر سجل التدقيق الشامل (Audit Log) الذي يُوثّق كل استعلام وإجابة.

على الصعيد الأكاديمي، يُمثّل المشروع إضافة معرفية في مجال دمج التقنيات الأربع (NL2SQL، RAG، Multi-Agent، Arabic NLP) في إطار موحد، وهو حقل بحثي لم يُطرق بكثرة في الأدبيات العربية. كما يفتح الباب أمام دراسات لاحقة لتطوير النظام ليشمل اللهجات المحلية، أو توسيع نطاقه ليشمل مؤسسات حكومية أخرى.

على الصعيد المجتمعي، يُسهم المشروع في دعم التحول الرقمي الذي تتبناه الدولة في إطار رؤية التحول الرقمي اليمني، ويعزز من قدرة المؤسسات على تقديم خدمات أفضل للمواطنين عبر توفير معلومات دقيقة وسريعة، مما ينعكس إيجاباً على رضا المستفيدين وثقتهم في الخدمات الحكومية.

4.1 مساهمات المشروع (Project Contributions)

يُقدّم مشروع «بيان» مساهمات بحثية وتطبيقية في أربعة محاور رئيسية، يمكن البناء عليها في دراسات وأعمال مستقبلية:

1. **المساهمة التقنية:** دمجُ أربع تقنيات متقدمة (NL2SQL، RAG، Multi-Agent، Arabic NLP) في إطار معماري موحد عبر LangGraph، وهو تكامل غير مسبوق في الأدبيات العربية وغير معتاد حتى في الأدبيات الإنجليزية، حيث تُقدَّم هذه التقنيات عادةً بشكل منفصل.
2. **المساهمة الابتكارية:** ابتكار طبقة Smart Views الذكية كآلية لتبسيط مخطط قاعدة البيانات قبل توليد SQL، مما يرفع دقة NL2SQL بشكل ملحوظ ويُمثّل مساهمة بحثية أصلية قابلة للتعميم على أي نظام إدارة قواعد بيانات مؤسسية.
3. **المساهمة اللغوية:** تقديم أول نظام RAG+NL2SQL عربي مفتوح المصدر يدعم الفصحى والعامية في واجهة محادثة RTL، مما يسدّ فجوةً واضحةً في دعم اللغة العربية في أنظمة الذكاء الاصطناعي المؤسسية.
4. **المساهمة التطبيقية:** تقديم تطبيق ميداني متكامل للبنية متعددة الوكلاء في بيئة مؤسسية تعليمية، باستخدام نظام Odoo Education كبيئة تطبيق واقعية، مع توثيق كامل للمنهجية وتقييم الأداء، مما يوفر مرجعاً تطبيقياً للمؤسسات العربية الراغبة في تبني تقنيات مشابهة.

5.1 مشكلة المشروع (Problem Statement)

تكمن المشكلة الرئيسية في تعثر الوصول الذكي والسريع إلى البيانات والمستندات المؤسسية نتيجة الاعتماد على الوسائل التقليدية واليدوية، مما يترتب عليه بطء في اتخاذ القرار، وهدر الموارد التشغيلية، وانخفاض مستوى جودة الخدمات المقدمة.

مظاهر المشكلة

1. **صعوبة الوصول المباشر للبيانات:** اقتصار استعلام قواعد البيانات على المتخصصين التقنيين (SQL)، مما يحرم الإدارة والموظفين غير المختصين من الاستعلام المباشر ويؤخر الحصول على المعلومة.
2. **شتات المستندات المؤسسية:** غياب آلية موحدة للبحث في المستندات والسياسات واللوائح غير المفهرسة، مما يجعل الوصول إلى نصوص القرارات واللوائح أمراً مستغرقاً للوقت.
3. **بطء وتكلفة إعداد التقارير:** الاعتماد على تجميع البيانات وإعداد التقارير والتحليلات يدوياً، مما يزيد من احتمالية الأخطاء التشغيلية ويؤخر صُنّاع القرار.

4. **غياب الشفافية وسجلات التتبع:** تفتقر العديد من العمليات التشغيلية إلى سجل تتبع موحد للاستعلامات والإجابات، مما يعوق تقييم الأداء والرقابة.
5. **ضعف التواصل المؤسسي:** غياب القنوات الذكية المباشرة للتفاعل بين المستفيدين والإدارة، مما ينعكس سلباً على الشفافية ورضا المستخدمين.

6.1 تعريف المشروع (System Definition)

يُعرَّف نظام «المساعد الذكي المؤسسي (بيان)» بأنه منصة محادثة ذكية متقدمة قائمة على معمارية الأنظمة متعددة الوكلاء (Multi-Agent System)، تمكّن المستخدمين من التفاعل التكاملي مع قواعد البيانات والمستندات المؤسسية باللغة العربية الطبيعية. وتتكون البنية الأساسية للنظام من المكونات التالية:

- **واجهة محادثة عربية** (RTL): واجهة تفاعلية تدعم اتجاه الكتابة من اليمين إلى اليسار لطرح الأسئلة واستقبال الردود الفورية.
- **لوحة تحكم ويب** (Web Admin Dashboard): منصة إدارية لمراقبة أداء النظام، وتتبع الوكلاء، وإدارة الصلاحيات والأمان.
- **خمسة وكلاء ذكاء اصطناعي متخصصين** يعملون في إطار منسق عبر تقنية LangGraph، وهم:
  - الوكيل المشرف (Supervisor Agent): لتوجيه الاستعلامات وإدارة تدفق العمل بين الوكلاء.
  - وكيل SQL (SQL Agent): لتحويل اللغة الطبيعية إلى استعلامات قواعد بيانات.
  - وكيل RAG (RAG Agent): للاسترجاع المعزز بالتوليد والبحث في المستندات.
  - وكيل التحليل (Analyst Agent): لاستخراج الرؤى والاتجاهات الإحصائية من البيانات.
  - وكيل التقارير (Reporter Agent): لتنسيق المخرجات وتوليد التقارير بصيغ متعددة.
- **قاعدة بيانات موحدة وآمنة:** لحفظ سجلات المستخدمين، والأنشطة، والمحادثات المخزنة.
- **طبقة العرض الذكية** (Smart Views Layer): طبقة وسيطة مخصصة تبسط بنية قاعدة البيانات لرفع دقة تحويل النص إلى استعلامات (NL2SQL).
- **بيئة محاكاة Odoo ERP:** نظام ERP مفتوح المصدر يُستخدم كبيئة تطبيق واقعية، يُحقن ببيانات تعادل 15 عاماً من نشاط مؤسسي افتراضي لاختبار النظام في ظروف قريبة من الواقع.

ويعمل النظام على رقمنة عمليات الوصول للبيانات وأتمتتها من خلال الآليات التالية:

1. **المحادثة التفاعلية الذكية:** معالجة اللغة الطبيعية العربية وفهم سياق الأسئلة.
2. **تحويل النص إلى استعلام** (NL2SQL): ترجمة أسئلة المستخدمين مباشرة إلى استعلامات قواعد بيانات تنفيذية.
3. **الاسترجاع المعزز بالتوليد** (RAG): الفهرسة والبحث النصي الذكي في المستندات الرسمية وإسناد الإجابات للمصادر.
4. **التحليل الإحصائي الآلي:** استخراج المؤشرات، والاتجاهات العامة، والانحرافات الإحصائية من البيانات.

5. **توليد التقارير:** صياغة وتصدير تقارير الجداول والرسوم البيانية بصيغ متعددة.
6. **التهيئة الذكية** (Smart Onboarding): عملية مؤتمتة تربط النظام ببيانات المؤسسة وفهرسة وثائقها خلال زمن قدره 8 ساعات.
7. **الأمان المتدرج:** تصنيف أمن الجداول إلى أربعة مستويات أمنية (Public، Masked، Restricted، Blacklist).
8. **تتبع مسار التنفيذ** (Agent Trace): الشفافية الكاملة في إظهار الخطوات التحليلية التي اتبعها الوكلاء للوصول إلى النتيجة.

7.1 نطاق المشروع (Project Scope)

يحدد هذا القسم النطاق الجغرافي والوظيفي للمشروع، بالإضافة إلى أصحاب المصلحة الرئيسيين المستفيدين من النظام.

1.7.1 النطاق الجغرافي (Geographical Scope)

يستهدف المشروع المؤسسات والجهات الأكاديمية بشكل عام. وسيُطبق النظام في مرحلته الأولى على نظام Odoo Education المخصص لإدارة البيانات الجامعية، كبيئة تطبيق واقعية، مع إمكانية تعميم النظام لاحقاً على أي مؤسسة عربية تطلب بيئة محادثة ذكية باللغة العربية للوصول إلى بياناتها ومستنداتها.

2.7.1 النطاق الوظيفي (Functional Scope)

يشمل النطاق الوظيفي تطوير نظام محادثة ذكي يدمج بين أربع تقنيات أساسية: تحويل اللغة الطبيعية إلى SQL، والاسترجاع المعزز بالتوليد، والأنظمة متعددة الوكلاء، ومعالجة اللغة العربية. ويقدم النظام للأنظمة الجامعية حلاً شاملاً للاستعلام عن البيانات والبحث في المستندات وتوليد التقارير، مع ضمان الأمان والشفافية.

3.7.1 أصحاب المصلحة (Stakeholders)

تشمل قائمة أصحاب المصلحة الرئيسيين:

- **المديرون التنفيذيون:** يحتاجون رؤى استراتيجية سريعة وموثقة.
- **المحللون:** يجرون استعلامات معقدة على البيانات.
- **الموظفون:** يبحثون عن معلومات محددة بسرعة.
- **مسؤولو النظام:** يشرفون على التشغيل والأمان.
- **مشغلو التهيئة:** مسؤولون عن الإعداد الأولي للمؤسسة.
- **الإدارة الأكاديمية:** تتبني النظام كأداة تحول رقمي.

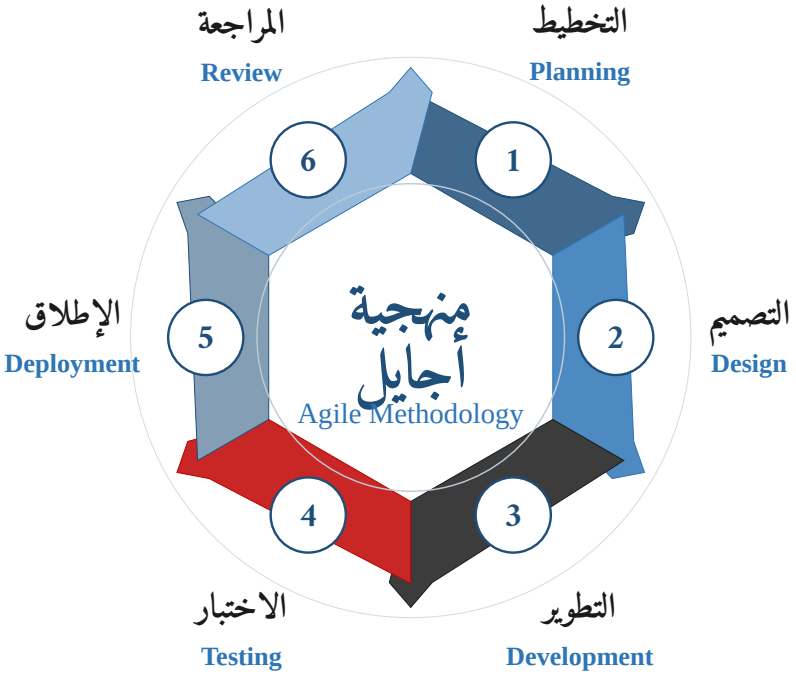
8.1 قيود المشروع (Limitations)

يواجه المشروع مجموعة من القيود التي يجب أخذها بعين الاعتبار:

- **قيود البنية التحتية:** محدودية الموارد الحاسوبية محلياً، مما يستوجب الاعتماد على نماذج لغوية متوسطة الحجم.
- **قيود اللغة:** ضعف دعم اللغة العربية في النماذج اللغوية التجارية، مما يستدعي حلولاً تكميلية.
- **قيود البيانات:** ندرة قواعد البيانات العربية الكاملة للاختبار.
- **قيود الوقت:** كون المشروع ضمن متطلبات التخرج يفرض جدولاً زمنياً محدوداً.
- **قيود التمويل:** عدم توفر ميزانية للاعتماد على نماذج سحابية مدفوعة، مما يُلزم بالاعتماد على حلول مفتوحة المصدر.
- **قيود الكوادر:** محدودية الفريق المعني بالتطوير والصيانة.

9.1 المنهجية المستخدمة في مشروع بيان (منهجية أجايل)

تم اعتماد منهجية أجايل (Agile) في تطوير مشروع بيان، نظراً لمرونتها وقابليتها للتكيّف مع المتطلبات المتغيرة. وتعتمد هذه المنهجية على تقسيم العمل إلى دورات تطوير قصيرة (Iterations)، مع التركيز على التسليم التدريجي والتغذية الراجعة المستمرة.



شكل 1-1: نموذج أجايل (Agile Model) المتبع في مشروع بيان.

1.9.1 مرحلة التخطيط (Planning)

في هذه المرحلة يتم تحديد متطلبات المشروع بشكل عام، وجمع احتياجات أصحاب المصلحة، وتحديد أولويات الميزات. يتم إعداد قائمة الميزات المطلوبة وتقدير الجهد الزمني لكل منها، مع تحديد أهداف الدورة القادمة بشكل واضح وقابل للقياس.

2.9.1 مرحلة التصميم (Design)

يتم في هذه المرحلة تصميم بنية النظام والمكونات الرئيسية، بما في ذلك تصميم قاعدة البيانات، وواجهات المستخدم، ووكلاء الذكاء الاصطناعي. يعتمد التصميم على مخططات UML ومخططات العلاقات (ERD) لضمان وضوح البنية وسهولة التطوير.

3.9.1 مرحلة التطوير (Development)

يتم فيها كتابة الشفرة البرمجية بناءً على التصميم المعتمد. تشمل هذه المرحلة تطوير الواجهة الخلفية (FastAPI)، والواجهة الأمامية (Next.js)، وتدريب وتهيئة وكلاء الذكاء الاصطناعي عبر LangGraph.

4.9.1 مرحلة الاختبار (Testing)

يتم فيها اختبار كل مكون من مكونات النظام بشكل منفصل (اختبار الوحدة)، ثم اختبار النظام ككل (اختبار التكامل). تشمل الاختبارات قياس دقة NL2SQL، ودقة الاسترجاع في RAG، وأداء وكلاء الذكاء الاصطناعي.

5.9.1 مرحلة الإطلاق (Deployment)

بعد التأكد من جاهزية النظام واكتمال الاختبارات، يتم إطلاق النسخة الأولية لاستخدام من قبل المؤسسة التجريبية. تشمل هذه المرحلة تشغيل النظام في البيئة الحقيقية، ومتابعة أدائه، وتقييم مدى تحقيقه لأهداف المشروع. يتم النشر باستخدام Docker Compose لتسهيل تشغيل جميع الخدمات (الواجهة، الخلفية، قواعد البيانات، نموذج LLM) في بيئة موحدة. كما يتم جمع الملاحظات من المستخدمين التجريبيين لتحسين النسخ القادمة.

6.9.1 مرحلة المراجعة (Review)

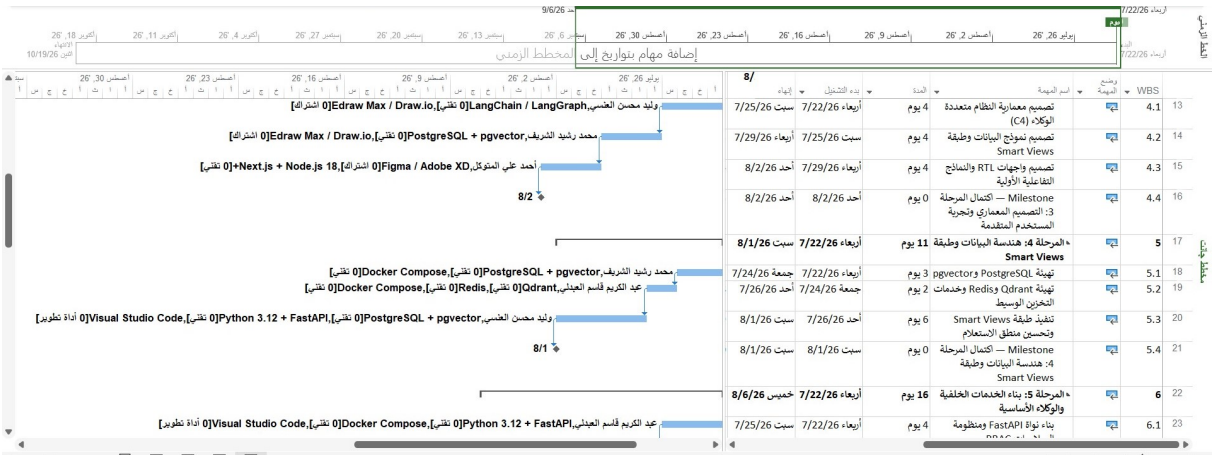
في هذه المرحلة تتم مراجعة العمل المنجز خلال دورة التطوير، وتقييم ما تم تحقيقه، وتحليل المشكلات التي ظهرت أثناء العمل. تهدف هذه المرحلة إلى تحسين العملية في الدورات التالية وضمان تطوير النظام بشكل أفضل. تُعد المراجعة عنصرًا أساسيًا في منهجية أجايل لأنها تساعد على التعلم المستمر وتحسين الأداء، خاصة في مشروع يعتمد على تقنيات الذكاء الاصطناعي التي قد تتطلب تعديلات مستمرة بناءً على نتائج الاختبارات وتغذية المستخدمين الراجعة.

10.1 الجدول الزمني للمشروع (Project Schedule)

تم تصميم جدول المشروع ليوضح تسلسل الأنشطة المشاركة في تطوير هذا المشروع. وقد قُسّم المشروع إلى مراحل رئيسية مع توزيع المهام على أعضاء الفريق والمدد الزمنية المتوقعة لإنجاز كل مرحلة. ونظراً لطول الفترة الزمنية للمشروع وتعدد مهام الفريق، قُسّمنا مخطط جانت إلى أربعة أجزاء تتابعية يعرض كلٌّ منها مرحلة زمنية محددة:

شكل 1-2: مخطط جانت لمراحل المشروع — الجزء الأول.

شرح الجزء الأول: يعرض هذا الجزء المراحل الأولى من المشروع، والتي تشمل جمع المتطلبات، تحليل النظام، ومرحلة التصميم المعماري الأولي للوكلاء وقاعدة البيانات.



شكل 1-3: مخطط جانت لمراحل المشروع — الجزء الثاني.

شرح الجزء الثاني: يركز هذا الجزء على مرحلة التنفيذ الأساسية للواجهة الخلفية (Backend) وتطوير وكلاء الذكاء الاصطناعي (LangGraph) وربطهم بقاعدة البيانات.

شكل 1-4: مخطط جانت لمراحل المشروع — الجزء الثالث.

شرح الجزء الثالث: يغطي هذا الجزء تطوير الواجهة الأمامية ،(Frontend) ودمجها مع الواجهة الخلفية، بالإضافة إلى مرحلة الاختبار الأولي للنظام وتصحيح الأخطاء.

شكل 1-5: مخطط جانت لمراحل المشروع — الجزء الرابع.

شرح الجزء الرابع: يختتم هذا الجزء مراحل المشروع باختبارات الأداء وتقييم النتائج، يليه توثيق الأطروحة وإعداد العرض التقديمي للمناقشة.

11.1 متطلبات الأدوات (Required Tools)

يستند تطوير مشروع بيان إلى مجموعة من الأدوات المادية والبرمجية التي تضمن جودة المنتج النهائي وكفاءة عملية التطوير. وفيما يلي عرض تفصيلي لهذه الأدوات.

1.11.1 الأدوات المادية (Hardware Tools)

يوضح الجدول 1-1 المواصفات المادية المطلوبة لكل عضو في فريق التطوير، إلى جانب الأجهزة اللازمة لتشغيل بيئة الخوادم.

جدول 1-1: الأدوات المادية المطلوبة للمشروع.

الأداة	الكمية	الاستخدام	المواصفات
اتصال بالإنترنت	---	للبحث، وتصحيح الأخطاء، وتحميل النماذج والأدوات، والتواصل مع الفريق	سرعة مناسبة للبحث والعمل الجماعي وتحميل ملفات بحجم عدة جيجابايتات (مثل نماذج Ollama).
لابتوب	5	لكل عضو جهاز للعمل على تطوير المشروع	**المعالج:** Core i7 أو ما يعادله (لتشغيل بيئة التطوير). **الرام** (RAM): 16 GB كحد أدنى، ويفضل 32 GB لتشغيل النماذج المحلية. **مساحة التخزين:** 512 GB SSD أو أكثر (لأنظمة التشغيل، قواعد البيانات، ونماذج الذكاء الاصطناعي).
خادم محلي أو حاسوب إضافي	1	تشغيل خدمات الخلفية الأساسية مثل قواعد البيانات (PostgreSQL، Qdrant، Redis) ونماذج Ollama محلياً	**المعالج:** Core i7 أو ما يعادله مع دعم لتسريع CUDA (بطاقة رسوميات NVIDIA). **الرام** (RAM): 32 GB أو أكثر. **مساحة التخزين:** 1 TB SSD أو أكثر. **نظام التشغيل:** Linux (Ubuntu) أو Windows مع WSL2.
ذاكرة فلاش (Flash memory)	2	لعمل نسخ احتياطية للمشروع، وملفات التطوير، والبيانات الأولية	سعة 64 GB أو أكثر، ويفضل أن تدعم USB 3.0 للسرعة.

2.11.1 الأدوات البرمجية (Software Tools)

يستعرض الجدول 1-2 الأدوات البرمجية الرئيسية المعتمدة في تطوير المشروع، مع ذكر الاستخدام المخصص لكل أداة.

البرنامج (Software)	الاستخدام
Windows 11	نظام تشغيل لتشغيل النظام على حواسيب التطوير، مع دعم WSL2 لتشغيل بيئات Linux على Windows.
LaTeX (latexmk + TeX Live)	لكتابة وتوثيق تقرير المشروع الأكاديمي.
Microsoft Project 2021	لإعداد الجدول الزمني للمشروع ومخططات جانت.
Microsoft PowerPoint 2021	لإعداد العرض التقديمي للمشروع.
Draw.io	لإنشاء المخططات والنماذج الخاصة بالمشروع.
Figma	لتصميم واجهات النظام (UI/UX).
Visual Studio Code	بيئة التطوير المتكاملة لكتابة كود المشروع (Python، JavaScript، TypeScript).
Python 3.12+	لغة البرمجة الأساسية لتطوير الواجهة الخلفية (FastAPI) ووكلاء LangGraph.
Node.js 18+	لغة البرمجة لتطوير الواجهة الأمامية (Next.js) وإدارة الحزم عبر npm.
FastAPI	إطار عمل لبناء الواجهة الخلفية (Backend API) وخدمات الوكلاء.
Next.js	إطار عمل لبناء الواجهة الأمامية (Frontend) مع دعم RTL للغة العربية.
LangGraph / LangChain	مكتبات لبناء وتنسيق الوكلاء المتعددين (Multi-Agent System).
PostgreSQL مع pgvector	قاعدة بيانات علائقية لتخزين البيانات المؤسسية والمتجهات (لـ RAG).
Qdrant	قاعدة بيانات متجهة (Vector Database) لتخزين واسترجاع تضمينات المستندات.
Redis	تخزين مؤقت (Cache) وطابور رسائل لتنسيق الوكلاء وإدارة الجلسات.

البرنامج (Software)	الاستخدام
Ollama	أداة لتشغيل النماذج اللغوية محلياً (مثل Qwen، Gemma، Llama).
Docker Desktop	منصة للحاويات لتشغيل جميع الخدمات (PostgreSQL، Qdrant، Redis، Ollama) بشكل موحد.
GitHub / Git	نظام التحكم في الإصدارات وإدارة المستودع البرمجي للمشروع.
Postman / Insomnia	اختبار واجهات برمجة التطبيقات (APIs) للواجهة الخلفية.

LT

جدول 1-2: الأدوات البرمجية المعتمدة في تطوير مشروع بيان.

12.1 هيكل المشروع (Project Structure)

يتكون هذا المشروع من سبعة (7) فصول رئيسية، يُكمل كلٌّ منها ما سبقه:

- **الفصل الأول — المقدمة:** الخلفية، الأهداف، الأهمية، المساهمات، المشكلة، النطاق، المنهجية، الجدول الزمني، والأدوات.
- **الفصل الثاني — الخلفية النظرية ومراجعة الأدبيات:** الأسس النظرية للتقنيات الأربع، الأنظمة السابقة، المقارنة، والفجوة البحثية.
- **الفصل الثالث — المنهجية وتحليل النظام:** منهجية التطوير، دراسة الجدوى، تحليل المتطلبات، حالات الاستخدام، وتحليل المخاطر.
- **الفصل الرابع — تصميم النظام:** المعمارية العامة، تصميم قواعد البيانات، تصميم واجهات المستخدم، المخططات، وتصميم APIs.
- **الفصل الخامس — تنفيذ واختبار النظام:** بيئة التطوير، التنفيذ، النشر، واختبارات (الوحدة، التكامل، الأداء، وقبول المستخدم).
- **الفصل السادس — النتائج والمناقشة:** النتائج الكمية والنوعية، واجهات النظام النهائية، المناقشة، التحديات، والدروس المستفادة.
- **الفصل السابع — الخاتمة والأعمال المستقبلية:** الخاتمة، تحقيق الأهداف، المساهمات، القيود، التوصيات، والأعمال المستقبلية.

الفصل

الثاني

الأدبيات ومراجعة النظرية الخلفية

1.2 مقدمة الفصل

يستعرض هذا الفصل الأسس النظرية والمراجعة النقدية للبيئة البحثية والتقنية التي يستند إليها مشروع «المساعد الذكي المؤسسي (بيان)». تتوزع مراجعة الأدبيات في هذا الفصل على خمسة محاور رئيسية تشمل: النماذج اللغوية الكبيرة (LLMs)، وتقنيات تحويل اللغة الطبيعية إلى استعلامات قواعد البيانات (NL2SQL)، ونظم الاسترجاع المعزز بالتوليد (RAG)، والمعماريات القائمة على الأنظمة متعددة الوكلاء (Multi-Agent Systems)، بالإضافة إلى تحديات معالجة اللغة العربية (Arabic NLP). ويُختتم الفصل بتحليل مقارن يبرز الفجوة البحثية والتطبيقية التي يستهدف المشروع معالجتها.

تكتسب هذه المراجعة أهمية مضاعفة في مشروع بيان، نظراً لأنه يدمج أربع تقنيات متقدمة في إطار موحد. لذا، يتطلب المشروع فحصاً دقيقاً لأسس كل تقنية على حدة، ثم تحليل كيفية تكاملها في الأدبيات السابقة. كما يسهم هذا الفصل في تحديد الفجوة البحثية بدقة، وتبرير القرارات التصميمية المُتخذة في الفصل الأول، خاصة اختيار LangGraph لإطار التنسيق بين الوكلاء، واعتماد طبقة Views الذكية لتحسين دقة NL2SQL، واستخدام Odoo ERP كبيئة محاكاة واقعية.

يتضمن الفصل ستة أقسام رئيسية: يبدأ بالخلفية النظرية للتقنيات الأساسية، ثم يستعرض الأنظمة والدراسات السابقة، يلي ذلك قسم مقارن يقدم جدولاً تحليلياً، ثم تحليل الفجوة البحثية، وأخيراً الخلاصة. ويعتمد الفصل على نظام استشهاد IEEE مع استمرارية الترقيم من الفصل الأول.

2.2 الخلفية النظرية

تستند الخلفية النظرية لمشروع بيان إلى خمس ركائز علمية مترابطة، تشكل مجتمعةً الأساس النظري للنظام. وفيما يلي عرض تفصيلي لكل ركيزة مع التركيز على التطورات الحديثة والأدوات التجارية المتاحة.

1.2.2 النماذج اللغوية الكبيرة (Large Language Models - LLMs)

تُعرَّف النماذج اللغوية الكبيرة بأنها نماذج ذكاء اصطناعي ضخمة تُدرَّب على كميات هائلة من النصوص بهدف فهم اللغة البشرية وتوليدها. وقد شهدت هذه النماذج تطوراً مذهلاً منذ عام 2020، حيث انتقلت من نماذج متخصصة في مهمة واحدة إلى نماذج عامة قادرة على أداء آلاف المهام بدقة عالية.

تعدّ عائلة GPT من OpenAI الأبرز في هذا المجال، حيث أطلقت الشركة نموذج GPT-3.5 أواخر 2022 ثم GPT-4 في مارس 2023 [16]. يُقدَّر أن GPT-4 يضم نحو 7.1 تريليون معامل، ويتميز بقدرات استدلالية متقدمة وفهم سياقي عميق. وفي المنطقة العربية، أطلقت شركة Inception الإماراتية نموذج «Jais» عام 2023 وهو أول نموذج لغوي عربي كبير، كما أطلقت Alibaba نموذج «Qwen» الذي يدعم العربية بشكل جيد [6].

تعتمد هذه النماذج على معمارية المحول (Transformer) التي قدمتها ورقة Vaswani وآخرين عام 2017 [17]، والتي أحدثت ثورة من خلال آلية الانتباه الذاتي (Self-Attention). هذه الآلية تتيح للنموذج فهم العلاقات بين الكلمات بغض النظر عن مواقعها في الجملة، مما يحسّن الفهم السياقي بشكل كبير.

رغم قوتها الواضحة، تعاني النماذج اللغوية الكبيرة من قيود جوهرية: الهلوسة (Hallucination) أي توليد معلومات خاطئة بثقة عالية، وعدم القدرة على الوصول للمعرفة الخاصة بالمجال، وقيود حجم السياق. هذه القيود هي ما دفع الباحثين لتطوير تقنيات RAG التي تُعالج لاحقاً في هذا الفصل.

النموذج اللغوي المعتمد في مشروع بيان

بناءً على مراجعة النماذج المتاحة وموازنتها مع قيود المشروع (البنية التحتية المحدودة، الحاجة لدعم العربية، الترخيص المفتوح)، يعتمد مشروع بيان نموذج Qwen 2.5 (14B) من Alibaba كنموذج رئيسي يُشغَّل محلياً عبر Ollama. وقع الاختيار على هذا النموذج للأسباب الآتية: (1) دعمه الجيد للغة العربية مقارنةً بنماذج مماثلة الحجم، (2) كفاءته العالية في المهام البرمجية وتوليد SQL، (3) حجمه المتوسط (14 مليار معامل) الذي يمكن تشغيله على خادم بـ 32 GB RAM دون الحاجة لبطاقة رسوميات باهظة، (4) ترخيصه المفتوح (Apache 2.0).

كما يُتيح النظام إمكانية التبديل إلى نموذج Jais (13B) الإماراتي المتخصص في العربية عند الحاجة لمعالجة نصوص عربية معقدة، وإلى نماذج سحابية (مثل GPT-4o) عند توفّر اتصال إنترنت مستقر وميزانية للتشغيل. جدول المقارنة التالي يُلخّص مواصفات النماذج المرشّحة:

جدول 2-1: مقارنة النماذج اللغوية المرشّحة لمشروع بيان.

النموذج	الحجم	دعم العربية	الترخيص	الاستخدام المقترح
Qwen 2.5 (14B)	14B	جيد	Apache 2.0	**النموذج الرئيسي** — تشغيل محلي افتراضي.
Jais (13B)	13B	ممتاز	مفتوح	بديل للنصوص العربية المعقدة.
Llama 3.1 (8B)	8B	متوسط	Llama 3 License	للأجهزة محدودة الموارد.
GPT-4o	غير معلن	جيد	تجاري	خيار سحابي عند توفّر الميزانية.

2.2.2 تقنية تحويل اللغة الطبيعية إلى SQL (NL2SQL)

تُعدّ تقنية NL2SQL إحدى أهم تطبيقات الذكاء الاصطناعي في مجال قواعد البيانات، حيث تُمكّن المستخدمين من الاستعلام عن البيانات بلغتهم الطبيعية دون الحاجة لمعرفة SQL [4]. وقد شهدت هذه التقنية قفزة نوعية منذ ظهور النماذج اللغوية الكبيرة، حيث ارتفعت الدقة على معيار Spider القياسي من نحو 35% عام 2018 إلى أكثر من 85% عام 2024، في حين لا تزال الدقة على معيار BIRD الأكثر واقعيةً دون 72%، مما يُبرز فجوة الأداء بين البيئات المعملية والواقعية.

من أبرز الأنظمة الحالية في هذا المجال Vanna.ai، وهو إطار عمل مفتوح المصدر يدمج بين RAG و NL2SQL، ويُحقق دقة تتجاوز 80% عند توفير السياق المناسب [2]. يعتمد Vanna على تدريب نموذج خاص بكل قاعدة بيانات، ويُستخدم على نطاق واسع في التطبيقات التجارية. كما أطلقت Snowflake أوائل 2025 نموذج Arctic-Text2SQL-R1 الذي حقق 83.71% دقة على معيار BIRD الأكثر تحدياً، باستخدام تقنية التعلم المعزز في التدريب [3].

من أهم التحديات في NL2SQL ربط المخطط (Schema Linking)، حيث يجب على النظام تحديد الجداول والأعمدة ذات الصلة بالسؤال من بين مئات العناصر في قاعدة البيانات. ولهذا السبب، يلجأ كثير من المطورين إلى تبسيط المخطط عبر طبقات وسيطة، وهو ما يبرر ابتكار طبقة Views الذكية في مشروع بيان. وتُقدَّر الدقة الأساسية (Baseline) لأي نظام NL2SQL يعتمد على نموذج لغوي عام (دون طبقة Views) بنحو 70% على قواعد البيانات المعقدة، وهو ما يهدف مشروع بيان لرفعه إلى 90%+ عبر طبقة Views الذكية وتقنيات Prompt Engineering المتقدمة.

تقنيات *Prompt Engineering* في *NL2SQL*

تعتمد جودة مخرجات NL2SQL بشكل جوهري على كيفية صياغة المُحفّز (Prompt) المُمرَّر للنموذج اللغوي. وقد تطورت تقنيات هندسة المُحفّزات في هذا المجال بشكل ملحوظ منذ عام 2023، وأبرزها:

- **التعلم القليلي (Few-Shot Learning):** إدراج أمثلة سؤال-جواب سابقة في المُحفّز لتوجيه النموذج نحو نمط الاستعلامات المطلوب، مما يرفع الدقة بنحو 15-20% مقارنةً بالتوليد الصفري (Zero-Shot).
- **سلسلة التفكير (Chain-of-Thought - CoT):** مطالبة النموذج بشرح خطواته التحليلية قبل توليد SQL، مما يحسّن الدقة في الاستعلامات متعددة الخطوات والوصلات (JOINs) المعقدة.
- **ربط المخطط الديناميكي (Dynamic Schema Linking):** تصفية الجداول والأعمدة ذات الصلة بالسؤال قبل تمريرها للنموذج، بدلاً من تمرير مخطط قاعدة البيانات كاملاً، مما يقلّل من تشتت النموذج ويحسّن الدقة.
- **التحقق الذاتي (Self-Verification):** مطالبة النموذج بإعادة فحص SQL المُولَّد وتصحيحه قبل التنفيذ، مما يقلّل من الأخطاء النحوية.

يعتمد مشروع بيان هذه التقنيات مجتمعةً ضمن وكلاء LangGraph، مع ابتكار طبقة Views الذكية كطبقة تجريدية إضافية تُبسّط المخطط قبل تطبيق هذه التقنيات، وهو ما يُتوقع أن يُحقّق تحسناً إضافياً في الدقة يتجاوز 20 نقطة مئوية فوق الـ Baseline.

ظاهرة الهلوسة في *NL2SQL*

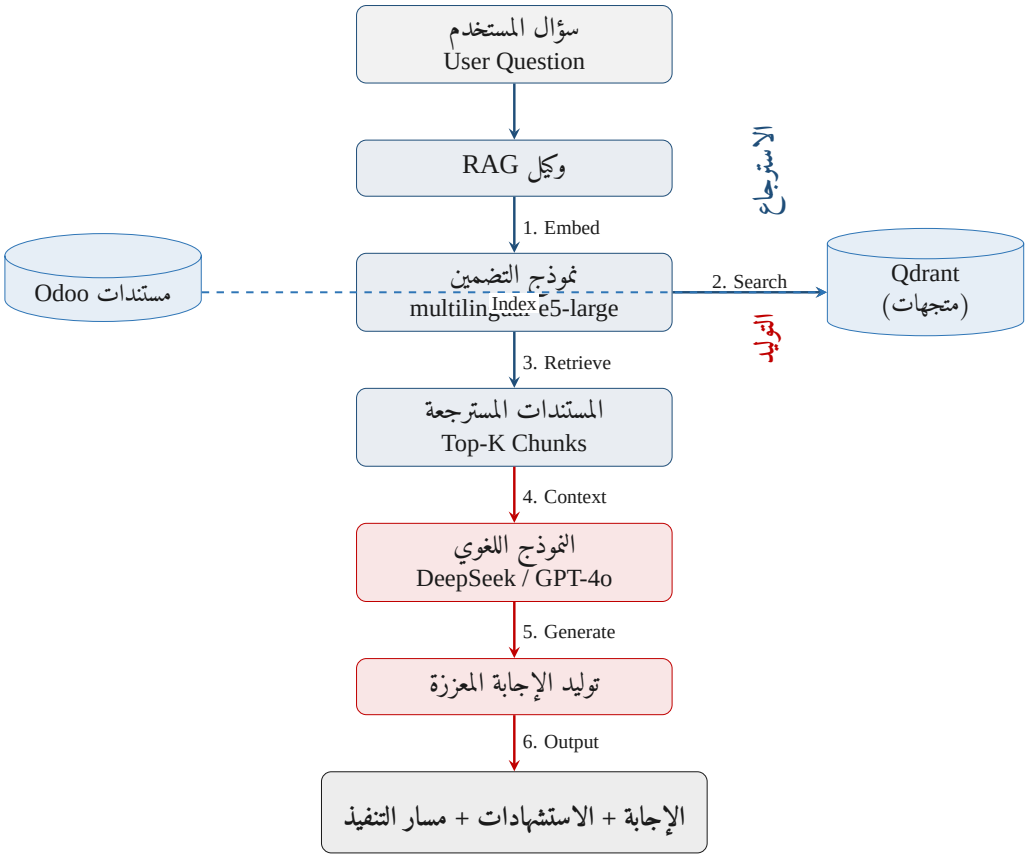
رغم التقدم الكبير في دقة NL2SQL، تظل ظاهرة الهلوسة (Hallucination) تحدياً جوهرياً يتمثل في قيام النموذج بتوليد استعلام SQL يبدو نحوياً صحيحاً لكنه دلالياً خاطئ — أي يُرجع بيانات غير مطلوبة، أو يتعرّض لجداول/أعمدة غير موجودة، أو يحسب دالة تجميعية بشكل خاطئ [18]. هذا النوع من الأخطاء أخطر من الأخطاء النحوية لأنه يمر دون أن يُكتشف في كثير من الأحيان، وقد يؤدي إلى اتخاذ قرارات مؤسسية بناءً على بيانات مضلّلة.

ولمعالجة هذه الظاهرة، يعتمد مشروع بيان ثلاث آليات دفاعية: (1) التحقق الذاتي من قبل النموذج قبل الإخراج، (2) تنفيذ الاستعلام في بيئة اختبار معزولة والتحقق من النتائج، (3) إسناد كل إجابة لمصدرها من خلال تتبع مسار التنفيذ (Agent Trace) الذي يُمكّن المستخدم من مراجعة الخطوات التي اتخذها الوكيل للوصول للنتيجة.

3.2.2 تقنية الاسترجاع المعزز بالتوليد (RAG)

تُمثل تقنية RAG إطاراً هجيناً يجمع بين قدرات النماذج اللغوية على التوليد، وقواعد المعرفة الخارجية على توفير المعلومات الدقيقة [5]. وقد أصبحت هذه التقنية المعيار الذهبي لتطبيقات LLM في البيئات المؤسسية منذ عام 2023، حيث تُعالج مشكلة الهلوسة وتُمكّن النموذج من الوصول للمعرفة الخاصة بالمجال.

يعمل نظام RAG وفق سير عمل من ثلاث مراحل: الاسترجاع (Retrieval) حيث يتم تحويل سؤال المستخدم إلى متجه وبحث عن المستندات الأكثر صلة، ثم التوليد المعزز (Augmented Generation) حيث يُمرَّر المستندات المسترجعة مع السؤال إلى النموذج لتوليد إجابة دقيقة، وأخيراً الاستشهاد (Citation) حيث يُربط كل جزء من الإجابة بالمصدر المستخرج منه.



شكل 2-1: مخطط تدفق تقنية RAG في نظام بيان.

يُوضح الشكل 2-1 دورة معالجة الطلب الفعلية لتقنية RAG في نظام بيان. تبدأ الدورة باستقبال وكيل RAG لسؤال المستخدم بالعربية، ثم تحويله إلى متجه رقمي عبر نموذج التضمين multilingual-e5-large. يُستخدم هذا المتجه للبحث في قاعدة البيانات المتجهة Qdrant التي تحتوي على مستندات Odoo المُفهرسة مسبقاً. تُسترجع أكثر المقاطع صلة، وتُمرَّر مع السؤال الأصلي إلى النموذج اللغوي (DeepSeek أو GPT-4o) لتوليد إجابة دقيقة مدعومة بالاستشهادات. يُختتم التدفق بإرجاع الإجابة مع مسار التنفيذ (Agent Trace) لضمان الشفافية. ينقسم التدفق إلى مرحلتين رئيسيتين: الاسترجاع (Retrieval) والتوليد المعزز (Augmented Generation).

من أبرز الأطر الحالية لبناء أنظمة RAG، يأتي LangChain [9] في المقدمة كإطار العمل الأكثر انتشاراً، ويقدم مكونات

قابلة للتخصيص للبحث والتوليد. ويأتي LlamaIndex [10] كإطار متخصص في RAG، يقدم أدوات متقدمة لفهرسة المستندات وإدارة السياق. كما تقدم Microsoft خدمة Azure AI Search المتكاملة مع RAG، وتقدم Google خدمة Vertex AI Search.

يعتمد أداء RAG بشكل جوهري على جودة نظام الاسترجاع، والذي يتطلب قاعدة بيانات متجهة (Vector Database) متخصصة. وقد ظهرت عدة حلول مفتوحة المصدر وتجارية في السنوات الأخيرة، أبرزها Qdrant [8] الذي اعتمد عليه مشروع بيان، و Pinecone التجاري، و Milvus، إضافة إلى امتداد pgvector لـ PostgreSQL.

كما يعتمد الاسترجاع على نموذج التضمين (Embedding Model) المسؤول عن تحويل النصوص إلى متجهات رقمية قابلة للبحث الشعاعي. ومن أبرز النماذج المتاحة: text-embedding-3-small و text-embedding-3-large من OpenAI (تجارية)، و all-MiniLM-L6-v2 (مفتوح المصدر، خفيف)، و multilingual-e5-large (مفتوح المصدر، يدعم العربية بشكل جيد). يعتمد مشروع بيان نموذج multilingual-e5-large لكونه مفتوح المصدر ويدعم اللغة العربية بشكل فعّال، مع إمكانية التبديل إلى نموذج Jais Embedding الإماراتي المتخصص في العربية عند توفره.

4.2.2 الأنظمة متعددة الوكلاء (Multi-Agent Systems)

تُعرَّف الأنظمة متعددة الوكلاء في سياق الذكاء الاصطناعي الحديث بأنها أنظمة تُكلف فيها عدة وكلاء مستقلين بإجراء مهام متخصصة، مع وجود آلية تنسيق تضمن تكامل مخرجاتهم. وقد ظهر هذا المفهوم بقوة منذ عام 2023 مع تطور قدرات النماذج اللغوية، حيث أصبح كل وكيل عبارة عن LLM مُوجَّه بمهمة وشخصية محددة.

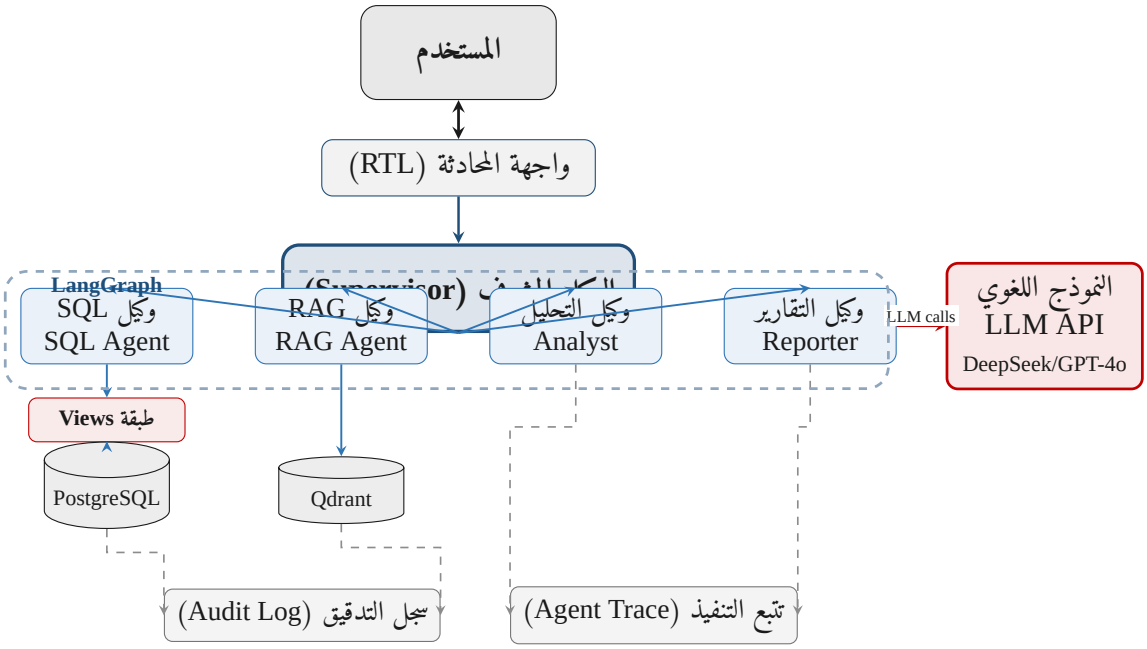
يُعدّ إطار LangGraph [7] من LangChain الأبرز في هذا المجال، ويستخدم نظرية المخططات لتمثيل تدفق العمل بين الوكلاء كعقد (Nodes) وحواف (Edges). يتميز بتحكم دقيق في التدفق ودعم الحلقات التكرارية والتفرعات المشروطة، وهو ما اعتمد عليه مشروع بيان. ويأتي إطار AutoGen [11] من Microsoft Research كخيار مرن قائم على المحادثات بين الوكلاء، لكنه يعاني من زيادة في استهلاك الـ Tokens.

إطار CrewAI [12] هو إطار حديث أطلق عام 2024، يركز على محاكاة فرق العمل البشرية بأدوار محددة، ويتميز بسهولة الاستخدام. أما إطار MetaGPT [13] فيقدم نهجاً مختلفاً، حيث يحاكي فريق تطوير برمجيات تقليدي (Product Manager, Architect, Engineer) لإنتاج شفرات برمجية متكاملة، وقد حقق تفوقاً ملحوظاً في إنتاج الكود الصحيح.

تُظهر هذه المراجعة أن LangGraph هو الإطار الأنسب لمشروع بيان، نظراً لكونه يدعم بناء أنظمة معقدة ذات وكلاء متخصصين، مع إمكانية تتبع مسار التنفيذ (Agent Trace) المطلوبة للشفافية في البيئة الجامعية.

يُوضح الشكل 2-2 المعمارية الكاملة لنظام بيان كنظام متعدد الوكلاء. يتفاعل المستخدم عبر واجهة محادثة عربية (RTL) مع الوكيل المشرف (Supervisor Agent) الذي يعمل كموجّه مركزي (Router)، فيُحلّل سؤال المستخدم ويُحدد الوكيل المتخصص الأنسب لمعالجته. يتوزع العمل على أربعة وكلاء متخصصين داخل إطار تنسيق LangGraph:

- **وكيل SQL:** يحوّل السؤال إلى استعلام قواعد بيانات عبر طبقة Views الذكية التي تُبسّط المخطط قبل التوليد، ثم ينفذه على PostgreSQL المتصل ببيانات Odoo Education.
- **وكيل RAG:** يبحث في المستندات المؤسسية المُفهرسة في Qdrant ويولّد إجابات مدعومة بالاستشهادات.
- **وكيل التحليل:** يستخرج الرؤى الإحصائية والاتجاهات من البيانات التي يُوفرها وكيل SQL.



شكل 2-2: معمارية النظام متعدد الوكلاء القائم على LangGraph.

- **وكيل التقارير:** ينسّق المخرجات من الوكلاء الآخرين ويُنتج تقارير منسقة بصيغ متعددة.

يُلاحظ أن جميع الوكلاء يتشاركون في الوصول إلى النموذج اللغوي الكبير (DeepSeek أو GPT-4o) عبر واجهة برمجية موحدة. كما تُسجَّل جميع العمليات في سجل التدقيق (Audit Log) لضمان المساءلة، ويُعرض مسار التنفيذ (Agent Trace) للمستخدم كحلقة تغذية راجعة تُعزز الشفافية والثقة في النظام.

5.2.2 تحديات معالجة اللغة العربية (Arabic NLP)

تواجه معالجة اللغة العربية طبيعياً تحديات جوهرية تختلف عن نظيراتها في اللغة الإنجليزية، وتعود هذه التحديات إلى عوامل لغوية وتقنية [6]. فمن الناحية اللغوية، تتميز العربية بالصرف الغني حيث يمكن اشتقاق عشرات الكلمات من جذر واحد ثلاثي، مما يزيد من حجم المفردات وصعوبة التطابق. كما تتميز بتركيب الجملة المرن الذي يصعّب التحليل النحوي الآلي، وظاهرة الإعراب التي تُغير الحركة النهائية للكلمات.

أما من الناحية التقنية، فتعاني العربية من فجوة الموارد الحادة؛ إذ تشير الإحصائيات إلى أن نسبة المحتوى العربي على الإنترنت لا تتجاوز 2.1% من الإجمالي، في حين أن المتحدثين بالعربية يشكلون نحو 5% من سكان العالم [6]. وتعتمد معظم التطبيقات التجارية على نماذج إنجليزية تُظهر أداءً ضعيفاً على العربية، وتشمل النماذج المدعومة للعربية: Jais النموذج الإماراتي المتخصص في العربية، و AraBERT النموذج المبني على BERT والمتخصص في النصوص العربية، و Qwen 2.5 من Alibaba الذي يدعم العربية جزئياً ويتميز بكفاءة عالية في المهام البرمجية.

تتفاقم هذه التحديات في السياق اليمني تحديداً، حيث تُستخدم لهجة محلية تختلف عن الفصحى والعاميات الخليجية والمصرية، وغير مدعومة بأي موارد برمجية معروفة. ولهذا السبب، يعتمد مشروع بيان استراتيجية لغوية مرنة تقبل الفصحى والعامية، مع الاستفادة من قدرة النماذج اللغوية الحديثة على التعميم بدلاً من تدريب نموذج متخصص.

3.2 الأنظمة والدراسات السابقة

تم تصنيف الأنظمة والدراسات السابقة المراجعة في خمس فئات رئيسية، وفقاً لمجال التخصص وارتباطها بمشروع بيان. وقد رُوعي في الاختيار التركيز على الأنظمة الحديثة (2021-2025) والتجارية المتاحة فعلياً.

1.3.2 الأنظمة في مجال تحويل اللغة الطبيعية إلى SQL

يُعدّ معيار Spider الذي أُطلق عام 2018 المرجع القياسي لتقييم أنظمة NL2SQL، حيث يضم 10,181 سؤالاً عبر 200 قاعدة بيانات. وفي عام 2023، أُطلق معيار BIRD الأكثر تحدياً لتركيزه على قواعد البيانات الواقعية مع قيم شاذة وأخطاء. حققت أفضل الأنظمة 40% فقط على BIRD مقابل 60% على Spider، مما يُبرز فجوة الأداء في البيئات الواقعية [4].

من أبرز الأنظمة الحالية في هذا المجال، يُبرز نظام Vanna.ai [2] كأحد أكثر الأطر مفتوحة المصدر انتشاراً. يعتمد Vanna على دمج RAG مع NL2SQL، ويتدرب على قاعدة بيانات كل عميل لرفع الدقة. أما نظام Snowflake Arctic-Text2SQL-R1 [3] فهو نموذج متخصص أطلق أوائل 2025، يستخدم تقنية التعلم المعزز في التدريب ويحقق 83.71% على معيار BIRD.

تجدر الإشارة إلى أن جميع هذه الأنظمة مصممة للبيئة الإنجليزية، ولا تقدم دعماً كافياً للغة العربية. كما أنها تعمل كأنظمة أحادية (Single-Agent)، ولا تستفيد من البنية متعددة الوكلاء في المهام المعقدة.

2.3.2 الأنظمة في مجال الاسترجاع المعزز بالتوليد (RAG)

منذ تقديم مفهوم RAG بشكلها الحديث عام 2020 [5]، شهدت هذه التقنية تطوراً متسارعاً وأصبحت المعيار الذهبي لتطبيقات LLM المؤسسية. أظهرت الدراسات أن RAG يقلل الهلوسة بنسبة تصل إلى 50% مقارنة بالنماذج اللغوية التقليدية، مما جعله خياراً مفضلاً للتطبيقات التي تتطلب دقة عالية.

من أبرز الأطر الحالية لبناء أنظمة RAG، يأتي LangChain [9] في المقدمة كإطار العمل الأكثر انتشاراً، ويقدم مكونات قابلة للتخصيص للبحث والتوليد. ويأتي LlamaIndex [10] كإطار متخصص في RAG، يقدم أدوات متقدمة لفهرسة المستندات وإدارة السياق. كما تقدم Microsoft خدمة Azure AI Search المتكاملة مع RAG، وتقدم Google خدمة Vertex AI Search.

على صعيد قواعد البيانات المتجهة، يبرز Qdrant [8] كحل مفتوح المصدر يتميز بأداء عالٍ ودعم فلترة المتجهات بالبيانات الوصفية، وهو ما اعتمد عليه مشروع بيان. كما تقدم Pinecone حلاً تجارياً سحابياً، ويقدم Milvus حلاً مفتوحاً موزعاً. وفي سياق قواعد البيانات العلائقية، يقدم امتداد pgvector لـ PostgreSQL إمكانية البحث الشعاعي دون الحاجة لقاعدة بيانات منفصلة.

3.3.2 الأنظمة في مجال الأنظمة متعددة الوكلاء

رغم التطور المتسارع الذي شهدته الأنظمة متعددة الوكلاء منذ عام 2023 (انظر المراجعة التفصيلية في القسم 4.2.2)، فإن التطبيقات المؤسسية الجاهزة التي تعتمد هذه البنية لا تزال محدودة. تتركز الأطر المتاحة (LangGraph، AutoGen، CrewAI، MetaGPT) على المستوى التقني لبناء الوكلاء، دون تقديم حلول جاهزة للمؤسسات. هذا يعني أن أي مؤسسة ترغب في اعتماد البنية متعددة الوكلاء مطالبة ببناء نظامها من الصفر، وهو ما يفسر ندرة الأنظمة المؤسسية المبنية على هذه التقنية، ويبرر أهمية مشروع بيان الذي يقدم تطبيقاً ميدانياً متكاملاً للبنية متعددة الوكلاء في بيئة جامعية.

4.3.2 المساعدات الذكية في التعليم العالي

رغم الانتشار الواسع للمساعدات الذكية في مجال خدمة العملاء، تبقى التطبيقات المتخصصة في التعليم العالي محدودة. أطلقت OpenAI عام 2024 خدمة ChatGPT Edu المخصصة للجامعات، التي تتيح بناء مساعدات مخصصة لكل مؤسسة. ومع ذلك، تظل هذه الخدمة عامة ولا تدمج NL2SQL للاستعلام عن قواعد البيانات الجامعية.

تقدم Microsoft خدمة Copilot for Education مع تكاملات محدودة مع أنظمة إدارة التعلم (LMS) مثل Canvas و Blackboard، لكنها لا تدعم العربية بشكل كامل. كما تقدم Anthropic خدمة Claude for Education التي تركز على المساعدة الأكاديمية للطلاب.

على الصعيد العربي، أطلقت العديد من الجامعات مساعدات ذكية بسيطة للأسئلة الشائعة، لكنها تعتمد على قواعد معرفة محدودة ولا تستفيد من تقنيات RAG أو NL2SQL. وأكدت دراسة Al-Khalifa وآخرين [14] أن معظم الجامعات العربية تستخدم chatbots بسيطة دون الاستفادة من التقنيات المتقدمة.

تُبرز هذه المراجعة أن الميدان يفتقر إلى نظام متكامل يجمع بين NL2SQL و RAG والأنظمة متعددة الوكلاء في بيئة جامعية عربية، وهو ما يسعى مشروع بيان لسده.

5.3.2 الدراسات العربية

على الصعيد العربي، تظل الدراسات في مجال المساعدات الذكية المؤسسية محدودة وتتسم بالتشتت بين أربع اتجاهات رئيسية لا تلتقي عادةً في بحث واحد: معالجة اللغة العربية، استرجاع المعلومات العربية، NL2SQL للعربية، وتطبيقات الذكاء الاصطناعي في التعليم.

دراسة Magdy وآخرين [15] استعرضت واقع البحث في استرجاع المعلومات العربية، وأكدت أن الأنظمة التجارية تعاني من ضعف 30-40% في الدقة على العربية مقارنة بالإنجليزية. كما أكدت دراسة El-Haj [6] المرجعية في معالجة اللغة العربية طبيعياً، أن المشاريع العربية تواجه ثلاث تحديات رئيسية: ندرة corpora عربية عالية الجودة، عدم وجود معايير قياسية موحدة، وضعف التمويل البحثي.

أما في مجال NL2SQL العربية تحديداً، فالدراسات شبه منعدمة. أغلب الأبحاث العربية في قواعد البيانات تركز على واجهات التعبير النمطي أو الترجمة الآلية للاستعلامات، دون الاستفادة من النماذج اللغوية الكبيرة. وقد رصدت دراسة Al-Khalifa [14] غياب أي نظام NL2SQL عربي منشور حتى عام 2023، مما يجعل مشروع بيان من الرواد في هذا المجال.

في مجال المساعدات الذكية التعليمية العربية، أطلقت جامعة الملك سعود عام 2022 مساعداً ذكياً للأسئلة الأكاديمية اعتمد على قاعدة معرفة ثابتة دون RAG، كما طوّرت جامعة الملك عبدالعزيز نظاماً مشابهاً لإرشاد الطلاب. لكن هذه المحاولات تظل بسيطة ولا تدمج تقنيات الاسترجاع المعزز بالتوليد، ولا تدعم الاستعلام عن قواعد البيانات الجامعية.

في البيئة اليمنية تحديداً، تكاد تنعدم الدراسات المنشورة في مجال الذكاء الاصطناعي المؤسسي. وأكد تقرير الأمم المتحدة للتحول الرقمي [1] أن اليمن تحتل المرتبة 175 عالمياً في مؤشر تطور الحكومة الإلكترونية. ويُعزى ذلك إلى عوامل متعددة، تشمل: محدودية البنية التحتية الرقمية، وضعف الاستثمار في البحث والتطوير، وندرة الكوادر المتخصصة. هذا الواقع يُبرز أهمية مشروع بيان كمساهمة بحثية وتطبيقية في سد هذه الفجوة، حيث يقدم أول نظام RAG+NL2SQL عربي مفتوح المصدر مخصص للبيئة الجامعية اليمنية، مع توثيق كامل لمنهجيته وتقييم أدائه.

4.2 مقارنة الأنظمة السابقة

يقدم هذا القسم تحليلاً مقارناً للأنظمة والدراسات السابقة المراجعة، مع التركيز على ستة أبعاد رئيسية: التقنيات المستخدمة، اللغة المدعومة، البيئة التطبيقية، دقة NL2SQL، دقة RAG، والبنية المعمارية. ويُلخص الجدول 2-2 أبرز الأنظمة ومقارنتها بمشروع بيان.

يُقرأ الجدول من اليسار إلى اليمين، حيث يُمثل العمود الأول اسم النظام أو الإطار البحثي، يليه التقنيات الأساسية التي بُني عليها، ثم اللغة المدعومة والبيئة التطبيقية المستهدفة. أما العمودان الرابع والخامس فيعرضان دقة النظام في تحويل اللغة الطبيعية إلى SQL ودقة الاسترجاع في RAG على التوالي (بعض القيم تركت فارغة ``---" لعدم توفر بيانات قياسية في الأدبيات). يُختتم الجدول بالبنية المعمارية للنظام (أحادية أم متعددة الوكلاء). يهدف هذا الجدول إلى إبراز الفجوة الواضحة في الأنظمة الحالية مقارنة بمشروع بيان، خاصة في دعم اللغة العربية والدمج بين التقنيات الأربع.

تكشف المقارنة عن عدة ملاحظات جوهرية. أولاً، تركز معظم الأنظمة في NL2SQL على البيئة الإنجليزية، مع غياب شبه تام للأنظمة العربية المتخصصة. ثانياً، تعتمد الأنظمة المتقدمة مثل Vanna.ai على دمج NL2SQL و RAG، لكنها لا تستخدم البنية متعددة الوكلاء، مما يُحدها في المهام المعقدة. ثالثاً، تفتقر التطبيقات في التعليم العالي إلى التكامل بين الاستعلام عن البيانات الديناميكية والبحث في المستندات الثابتة.

كما يُلاحظ أن مشروع بيان هو الوحيد الذي يجمع بين التقنيات الأربع (NL2SQL، RAG، Multi-Agent، Arabic NLP) في إطار موحد، مع تخصيصه للبيئة الجامعية اليمنية. هذا التكامل يمثل نقطة قوة، لكنه في الوقت ذاته يفرض تحديات تصميمية معقدة، خاصة في تنسيق الوكلاء وضمان جودة الأداء عبر المكونات المختلفة.

5.2 الفجوة البحثية

يتميز مشروع «بيان» بكونه النظام الوحيد — في حدود ما اطلعت عليه الدراسة — الذي يجمع بين التقنيات الأربع الأساسية (NL2SQL، RAG، Multi-Agent Systems، Arabic NLP) في إطار موحد ومحلي الاستضافة، مع تخصيصه المباشر للبيئة المؤسسية والتعليمية. ورغم أن هذا التكامل يمثل نقطة قوة واضحة، إلا أنه يفرض في الوقت ذاته تحديات تصميمية وتنفيذية معقدة، خاصة في تنسيق عمل الوكلاء البرمجيين وضمان جودة الأداء والاعتمادية عبر المكونات المختلفة.

جدول 2-2: مقارنة شاملة بين الأنظمة السابقة ومشروع بيان.

النظام/الإطار	التقنيات المستخدمة	اللغة	البيئة التطبيقية	دقة NL2SQL	دقة RAG	البنية
[2] Vanna.ai (2024)	NL2SQL + RAG	إنجليزية	تجارية	80%	غير مذكورة	أحادية
Arctic-[3] Text2SQL (2025)	NL2SQL + RL	إنجليزية	تجارية	83%.71 (BIRD)	---	أحادية
[9] LangChain (2023)	RAG + LLM	متعددة	عامة	---	عالية	أحادية
LlamaIndex (2023) [10]	RAG + LLM	متعددة	عامة	---	عالية	أحادية
[11] AutoGen (2023)	Multi-Agent	إنجليزية	عامة	---	---	متعددة
[12] CrewAI (2024)	Multi-Agent	إنجليزية	عامة	---	---	متعددة
[13] MetaGPT (2024)	Multi-Agent	إنجليزية	تطوير برمجي	---	---	متعددة
ChatGPT Edu (2024)	LLM + RAG	متعددة	تعليم عالٍ	---	جيدة	أحادية
[8] Qdrant (2024)	Vector DB	متعددة	عامة	---	عالية	---
مشروع بيان (المقترح)	NL2SQL + RAG + Multi-Agent	عربية (فصحى وعامية)	جامعي (يمني)	85% ≥ (مستهدف)	80% ≥ (مستهدف)	متعددة (LangGraph)

بناءً على المراجعة الشاملة للأنظمة والدراسات السابقة والتحليل المقارن، يمكن تحديد الفجوة البحثية والتطبيقية التي يسعى مشروع «بيان» لسدها في **خمسة محاور رئيسية**:

المحور الأول — الفجوة التقنية: رغم وجود أنظمة NL2SQL متقدمة (Vanna.ai، Arctic-Text2SQL) وأنظمة RAG فعالة (LangChain، LlamaIndex)، إلا أنه لا يوجد نظام يجمع بين التقنيتين في إطار موحد مع بنية متعددة الوكلاء. تعمل هذه الأنظمة بشكل منفرد، مما يُفقدها التكامل المطلوب في البيئات المؤسسية المعقدة. مشروع بيان يسد هذه الفجوة عبر الدمج العضوي للتقنيات الأربع (NL2SQL، RAG، Multi-Agent، Vector DB).

المحور الثاني — الفجوة اللغوية: أكدت دراسات El-Haj [6] و Magdy [15] على الفجوة الواضحة في دعم اللغة العربية في أنظمة الذكاء الاصطناعي. تعاني الأنظمة التجارية من انخفاض الدقة 30-40% على العربية مقارنة بالإنجليزية، كما أنها غير مصممة لفهم اللهجات المحلية. مشروع بيان يعالج هذه الفجوة عبر دعم العربية الفصحى والعامية في واجهة محادثة RTL، مع الاستفادة من نماذج Qwen و Jais التي تدعم العربية.

المحور الثالث — الفجوة التطبيقية (Applied Gap): تفتقر الأنظمة التطبيقية في المؤسسات والقطاع الأكاديمي إلى التكامل بين الاستعلام عن البيانات الرقمية والمالية (NL2SQL) والبحث في اللوائح والسياسات والمستندات التنظيمية (RAG). تركز الأنظمة الحالية (مثل ChatGPT Enterprise أو ChatGPT Edu) على جانب واحد فقط، مما يحد من قدرتها على تلبية الاحتياجات التشغيلية الكاملة للمستفيدين. يقدم نظام «بيان» حلاً موحداً يدمج الجانبين عبر وكلاء متخصصين.

المحور الرابع — الفجوة السياقية: الأنظمة التجارية مثل Vanna.ai و ChatGPT Edu مصممة لبيئات تجارية غربية، ولا تأخذ في الاعتبار خصوصية البيئة الجامعية اليمنية (اللوائح المحلية، اللهجة، البنية التحتية المحدودة). مشروع بيان مصمم خصيصاً لهذه البيئة، مع استخدام Odoo ERP كبيئة محاكاة واقعية تُحقن ببيانات تعادل 15 عاماً.

المحور الخامس — فجوة الابتكار: لا توجد دراسة سابقة قدمت طبقة Views ذكية كآلية لتبسيط مخطط قاعدة البيانات وتحسين دقة NL2SQL. هذا الابتكار يمثل مساهمة بحثية أصلية يمكن البناء عليها في دراسات مستقبلية، وقد يفتح آفاقاً جديدة في مجال Database Abstraction Layer.

يُلخص الجدول 2-3 الفجوة البحثية ومساهمة مشروع بيان في سدها عبر المحاور الخمسة.

جدول 2-3: ملخص الفجوة البحثية ومساهمة مشروع بيان في سدها.

المحور	الفجوة المحددة	مساهمة مشروع بيان
تقنية	غياب الدمج بين NL2SQL و RAG و Multi-Agent	دمج التقنيات الأربع في إطار موحد
لغوية	ضعف دعم العربية واللهجات	دعم فصحى وعامية في واجهة RTL
تطبيقية	فصل الاستعلام عن البيانات عن البحث في المستندات	نظام موحد يجمع الجانبين
سياقية	غياب حلول مخصصة للبيئة المؤسسية والمحلية	تخصيص كامل للبيئة المؤسسية مع بيئة محاكاة لأكثر من نوع من قواعد البيانات
ابتكارية	غياب طبقة تجريدية مبسطة فوق قواعد البيانات	ابتكار طبقة Views ذكية

6.2 الخلاصة

استعرض هذا الفصل الإطار النظري الشامل الذي يستند إليه «**مشروع المساعد الذكي المؤسسي (بيان)**»، بدءاً من المفاهيم الأساسية في النماذج اللغوية الكبيرة (LLMs)، مروراً بالتقنيات المتخصصة في NL2SQL و RAG والأنظمة متعددة الوكلاء (Multi-Agent Systems)، وصولاً إلى تحديات معالجة اللغة العربية طبيعياً (Arabic NLP). وقد كشف الاستعراض النظري أن المشروع يعتمد على ركائز علمية متينة ومثبتة، مع الاستفادة من أحدث الأطر والأدوات المتاحة (2021-2025) في كل مجال.

كما قدم الفصل مراجعة شاملة للأنظمة والدراسات السابقة ومقارنة مرجعية بينها. وأظهرت المراجعة أن مجال NL2SQL شهد تقدماً كبيراً في السنوات الثلاث الأخيرة مع ظهور أطر مثل Vanna.ai و Arctic-Text2SQL. وفي مجال RAG، أثبتت الأطر المتقدمة مثل LangChain و LlamaIndex كفاءتها العالية. وفي مجال الأنظمة متعددة الوكلاء، أظهر إطار **LangGraph** تفوقاً ملحوظاً في إدارة الحالات (State Management) والتحكم في مسارات التنفيذ.

كشف التحليل المقارن أن مشروع «بيان» يمثل خطوة متقدمة في الدمج الشامل للتقنيات الأربع في بيئة عمل مؤسسية، حيث اتُخذت جامعة العلوم والتكنولوجيا كدراسة حالة وتطبيق تجريبي. وخلص الفصل إلى تحديد خمسة محاور رئيسية للفجوة البحثية (تقنية، لغوية، تطبيقية، سياقية، وابتكارية) يسعى المشروع لسدها، مما يُبرر القرارات التصميمية والهندسية المتخذة، ويهيئ القارئ للفصل الثالث الذي يقدم تحليلاً تفصيلياً للمتطلبات والمنهجية المتبعة في تطوير النظام.

1.6.2 معايير تقييم النظام (Evaluation Metrics)

لضمان تقييم منهجي وموضوعي لنتائج النظام، تم تحديد مجموعة من المقاييس المعيارية الموزعة على ثلاثة محاور:

- **مقاييس دقة NL2SQL:**
  - Execution Accuracy (EX): نسبة الاستعلامات التي تُرجع النتيجة الصحيحة عند التنفيذ.
  - Exact Match (EM): نسبة استعلامات SQL المُولَّدة المطابقة تماماً للحل المرجعي.
- **مقاييس جودة RAG (RAGAS Framework):**
  - Faithfulness: مدى التزام الإجابة بالمستندات المسترجعة دون هلوسة.
  - Answer Relevancy: مدى صلة الإجابة بسؤال المستخدم.
  - Context Precision/Recall: جودة المستندات المسترجعة.
- **مقاييس الأداء التقني وتجربة المستخدم:**
  - Latency: زمن الاستجابة (مستهدف $\leq$ 3 ثوانٍ).
  - Throughput: عدد الاستعلامات في الثانية.
  - System Usability Scale (SUS): مقياس قابلية الاستخدام (مستهدف $\geq$ 70).
  - Customer Satisfaction (CSAT): رضا المستخدمين (مستهدف $\geq$ 80%).

سيتم تطبيق هذه المقاييس تفصيلياً في الفصل السادس (النتائج والمناقشة) بعد إكمال تنفيذ النظام واختباره.

الفصل

الثالث

النظام وتحليل المنهجية

1.3 منهجية التطوير

اعتمد فريق العمل منهجية التطوير الرشيقة Agile في بناء نظام «بيان»، وذلك لمواءمتها مع طبيعة المشروع الذي يجمع بين مكونات برمجية وتقنيات ذكاء اصطناعي متقدمة، مثل توليد استعلامات قواعد البيانات من اللغة الطبيعية NL2SQL، والاسترجاع المعزز بالتوليد RAG، ومعالجة اللغة العربية، وإدارة الأنظمة متعددة الوكلاء المنسقة باستخدام إطار LangGraph.

وتتطلب هذه التقنيات تكراراً مستمراً في تحسين النماذج اللغوية الكبيرة LLMs، والمطالبات Prompts، وهيكلة المعرفة، فضلاً عن التحقق المتكرر من دقة النتائج وزمن الاستجابة. لذلك، تقسم المنهجية الرشيقة نطاق العمل إلى دورات تطوير قصيرة ومحددة زمنياً تسمى التكرارات أو الدورات Iterations / Sprints. وفي نهاية كل دورة، تُراجع المخرجات وتُقيّم قبل الانتقال إلى الدورة اللاحقة، مما يتيح التكيف السريع مع التغيرات في المتطلبات أو مع نتائج تقييم النماذج اللغوية دون التأثير الجوهري في الجدول الزمني العام للمشروع.

1.1.3 دورة التطوير الرشيقة

تتكون كل دورة تطوير في مشروع «بيان» من ست مراحل متتابعة، مع إمكانية تكرارها عند ظهور متطلبات جديدة أو الحاجة إلى تحسين أحد المكونات. وتضمن هذه المراحل الانتقال المنظم من تحديد نطاق العمل إلى التقييم والتحسين، كما هو موضح في الجدول 3-1.

2.1.3 الدورات التنفيذية للمشروع

لضمان التنفيذ المنهجي لمكونات نظام «بيان»، قُسّم نطاق المشروع إلى أربع دورات تطوير تكرارية متتالية، بحيث تركز كل دورة على مكون رئيسي أو مجموعة مترابطة من المكونات، وتنتهي بمخرج قابل للتقييم والدمج مع بقية النظام.

الدورة الأولى: بناء وكيل استعلام قواعد البيانات

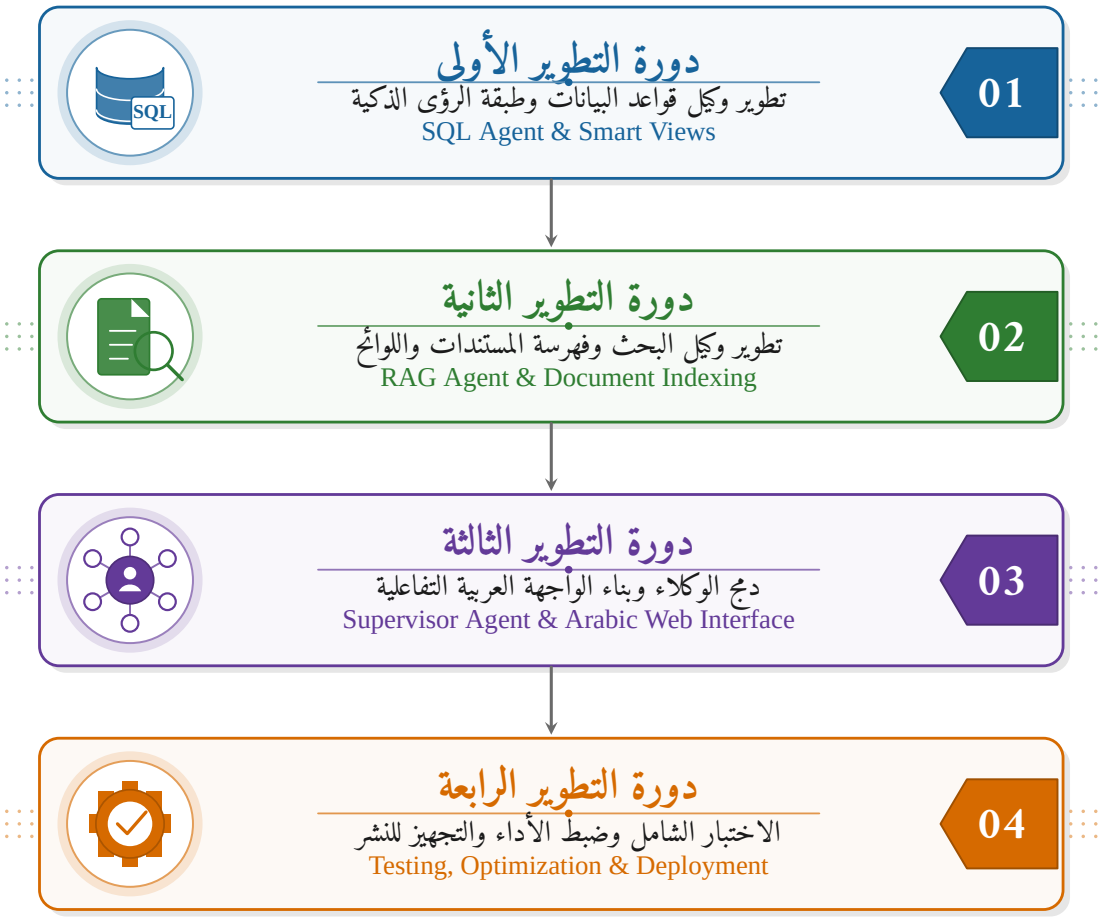
المستهدف: إعداد المعمارية الأساسية للتعامل مع البيانات المهيكلة في قاعدة بيانات PostgreSQL، مع ضمان أمن الاستعلامات.

الأنشطة: تطوير طبقة الرؤى الذكية Smart Views لعزل الجداول المعقدة، وبناء محرك تحويل النص إلى استعلام NL2SQL Engine، وإدراج آليات العزل للقراءة فقط Read-Only Sandboxing لمنع التعديل غير المصرح به، ثم اختبار دقة التوليد على استعلامات مالية وإدارية وأكاديمية.

المخرج: وكيل SQL مستقل يعمل عبر FastAPI ويسترجع البيانات بدقة من الجداول المحددة مع الالتزام بقيود الوصول الآمن.

جدول 3-1: مراحل دورة التطوير الرشيقة المطبقة في مشروع «بيان».

المرحلة	أبرز الأنشطة	المخرج الرئيسي
التخطيط Planning	تحديد أهداف الدورة الحالية، وإعداد قائمة مهام الدورة Sprint Backlog، ومراجعة المتطلبات الوظيفية وغير الوظيفية المرتبطة بالمكون المستهدف.	وثيقة نطاق الدورة وقائمة المهام المحددة.
التصميم Design	نمذجة المعمارية الفنية للوكيل المستهدف، وتحديد هياكل البيانات والواجهات البرمجية APIs، وإعداد التصميم الأولي لهندسة المطالبات Prompt Engineering.	مخططات التدفق، وهياكل البيانات، ومخطط هندسة المطالبات.
التطوير Development	كتابة الشفرة البرمجية باستخدام Python وFastAPI وLangGraph للواجهة الخلفية، أو Next.js للواجهة الأمامية، وربط الوكلاء بمصادر البيانات PostgreSQL وQdrant.	مكون برمجي جاهز يعمل بصورة مستقلة، مثل وكيل متخصص أو واجهة النظام.
الاختبار Testing	تنفيذ اختبارات الوحدة Unit Testing واختبارات التكامل Integration Testing، ومعايرة دقة الاستعلامات، والحد من الهلوسة، وقياس مؤشرات الأداء المعتمدة.	تقرير اختبارات موثق يوضح نسب الدقة وزمن الاستجابة.
الإطلاق والدمج Deployment	بناء المكونات البرمجية وحزمها داخل حاويات Docker Compose، ودمجها في البيئة الموحدة، والتحقق من توافقها مع بقية الوكلاء.	نسخة محدثة من النظام، مدمجة ومجربة في بيئة التطوير.
المراجعة والتقييم Review	تقديم عرض توضيحي للمخرجات أمام مشرف المشروع، وتوثيق الدروس المستفادة، وتحديث الأولويات وقائمة مهام الدورة التالية.	قائمة تحسينات وتعديلات معتمدة للدورة المباشرة التالية.



شكل 3-1: خارطة الدورات التطويرية الأربع المعتمدة في مشروع «بيان».

الدورة الثانية: بناء وكيل البحث في المستندات واللوائح

المستهدف: معالجة البيانات غير المهيكلة، بما في ذلك اللوائح والسياسات والمستندات التنظيمية الخاصة بالجامعة والمؤسسات ذات الصلة.

الأنشطة: إعداد قاعدة البيانات المتجهة Qdrant Vector DB، وتطوير معالج للنصوص العربية وتجزئتها Text Chunking، وإنشاء المتجهات Embeddings باستخدام نماذج داعمة للغة العربية، وبناء آلية استرجاع معزز بالمصادر ترفق اسم المستند ورقم الصفحة مع الإجابة متى توفرت البيانات.

المخرج: وكيل RAG متكامل يجيب عن الأسئلة النصية بإجابات دقيقة ومدعومة بمراجع واضحة.

الدورة الثالثة: دمج الوكلاء وبناء الواجهة الأمامية

المستهدف: تحقيق التكامل بين المكونات البرمجية وإتاحة واجهة مستخدم عربية سلسة.

الأنشطة: بناء الوكيل المشرف Supervisor Agent باستخدام إطار LangGraph لتوجيه الاستفسارات Intent Routing إلى الوكيل المختص (وكيل SQL أو وكيل RAG أو كليهما)، وإدارة ذاكرة السياق Context Management، ثم تطوير واجهة محادثة تفاعلية باللغة العربية وبدعم الاتجاه من اليمين إلى اليسار RTL باستخدام Next.js وTailwind CSS.

المخرج: نظام موحد متعدد الوكلاء يعمل عبر واجهة ويب عربية حوارية متكاملة.

الدورة الرابعة: الاختبار الشامل وضبط الأداء والنشر

المستهدف: التحقق من موثوقية النظام، وإدارة المخاطر التقنية، وتجهيز البيئة التشغيلية للنشر والتقييم.

الأنشطة: تنفيذ اختبارات شاملة من البداية إلى النهاية End-to-End Testing، وتقييم زمن الاستجابة وجودة الإجابات، وضبط أداء النموذج اللغوي المحلي Qwen 2.5 14B عبر Ollama، ثم حزم النظام بالكامل باستخدام Docker وDocker Compose.

المخرج: بيئة تشغيل موحدة قابلة للنشر السريع، تحتوي على مكونات نظام «بيان» في حاويات مستقلة وجاهزة للتسليم والتطبيق التجريبي.

3.1.3 ملخص الدورات ومخرجاتها

يلخص الجدول 3-2 العلاقة بين كل دورة تطوير والمكون المستهدف والأدوات المستخدمة والمخرج المباشر المتحقق منها.

جدول 3-2: الملخص التنفيذي للدورات الأربع في منهجية تطوير نظام «بيان».

الدورة	المكون المستهدف	الأنشطة والأدوات الرئيسية	المخرج النهائي المباشر
Sprint 1	وكيل SQL وطبقة Smart Views	PostgreSQL، وFastAPI، ومحرك NL2SQL، وطبقة الرؤى الذكية.	وكيل SQL آمن ومستقل لاسترجاع البيانات المهيكلة.
Sprint 2	وكيل RAG والبحث في المستندات	قاعدة Qdrant المتجهة، والمتجهات الدلالية العربية، وتجزئة المستندات.	وكيل RAG يجيب استناداً إلى السياسات واللوائح مع توثيق المصدر.
Sprint 3	دمج الوكلاء والواجهة الأمامية	الوكيل المشرف LangGraph، وNext.js، وTailwind CSS.	نظام هجين موحد يعمل عبر واجهة ويب عربية حوارية.
Sprint 4	الاختبار والتحسين والنشر	Docker Compose، وOllama (Qwen 2.5 14B)، واختبارات End-to-End.	نسخة تشغيلية محزمة وجاهزة للتقييم والتطبيق التجريبي.

وبذلك تحقق المنهجية تدرجاً عملياً يبدأ ببناء الوكلاء المتخصصين، ثم دمجهم في معمارية متعددة الوكلاء، وينتهي بالاختبار الشامل والتجهيز للنشر. كما تسمح المخرجات المرحلية بتقييم كل مكون بصورة مستقلة قبل اعتماده ضمن النظام الموحد.

2.3 دراسة الجدوى

3.3 تحليل المتطلبات

يُعد تحليل المتطلبات من أهم مراحل تطوير أي نظام برمجي، إذ يمثل الأساس الذي تُبنى عليه مراحل التصميم والتنفيذ اللاحقة. يهدف هذا القسم إلى تحديد المتطلبات الوظيفية وغير الوظيفية لنظام «المساعد الذكي المؤسسي (بيان)» بصورة دقيقة وقابلة للقياس، بحيث يمكن التحقق من تحقيقها عند اختبار النظام وتقييمه. وقد استُرشد في تصنيف المتطلبات بالمعايير الأكاديمية في هندسة المتطلبات البرمجية، وبمعيار الجودة الدولي ISO/IEC 25010 لتصنيف المتطلبات غير الوظيفية.

1.3.3 منهجية جمع المتطلبات

اعتمد فريق المشروع في جمع المتطلبات على أربعة مصادر رئيسية تكاملت لتشكيل صورة شاملة عن احتياجات النظام:

1. **تحليل النظام الحالي:** دراسة بنية قاعدة بيانات Odoo Education وجداولها وعلاقاتها ووظائفها المتاحة لفهم البيئة التي سيعمل فيها النظام المقترح.
2. **المشاورات مع المشرف وأصحاب المصلحة:** عقد مناقشات مع المشرف، وإجراء مقابلات غير رسمية مع مستخدمي نظام ERP، من إداريين ومحللين، لفهم احتياجاتهم اليومية ونقاط الضعف في النظام الحالي.
3. **التحليل المقارن للأنظمة السابقة:** استخلاص الفجوات والمتطلبات من مراجعة أنظمة مثل Vanna.ai وLangChain وChatGPT Edu، كما عُرضت في الفصل الثاني.

4. **مراجعة المعايير والأدبيات:** الاسترشاد بمعايير IEEE وISO/IEC 25010 عند صياغة المتطلبات وتصنيفها، لضمان التوافق مع الممارسات الأكاديمية المعتمدة.

2.3.3 أصحاب المصلحة والمستخدمون المستهدفون

يستهدف نظام «بيان» خمس فئات رئيسية من مستخدمي أنظمة ERP المؤسسية، لكل منها دور واهتمامات مختلفة. وقد استُخدم نظام Odoo Education الجامعي بوصفه بيئة تطبيق واقعية ومثالاً توضيحياً، مع بقاء التصميم عاماً وقابلاً للتطبيق على أنظمة ERP المؤسسية الأخرى.

جدول 3-3: أصحاب المصلحة في نظام «بيان».

صاحب المصلحة	الدور	الاهتمامات الرئيسية
صانع القرار المؤسسي	اتخاذ القرارات الاستراتيجية المبنية على البيانات	الحصول على رؤى استراتيجية سريعة وموثوقة من بيانات ERP.
محلل بيانات ERP	تحليل البيانات المؤسسية وإعداد التقارير	تنفيذ استعلامات معقدة وإعداد تقارير مفصلة من نظام ERP.
موظف العمليات	تنفيذ العمليات اليومية في نظام ERP	الوصول السريع والسهل إلى المعلومات المؤسسية.
مدير النظام	الإشراف على تشغيل النظام وأمانه	مراقبة الأداء وإدارة المستخدمين والصلاحيات.
مشغل التهيئة	الإعداد الأولي وربط النظام بنظام ERP	أتمتة التهيئة وفهرسة المستندات المؤسسية.

3.3.3 وصف النظام المقترح

يُعرّف نظام «بيان» بأنه منصة محادثة ذكية قائمة على معمارية الأنظمة متعددة الوكلاء Multi-Agent System، تمكن المستخدمين من التفاعل باللغة العربية الطبيعية مع قواعد البيانات والمستندات المؤسسية. ويتكون النظام تشغيلياً من المكونات الرئيسية الآتية:

- **واجهة المحادثة العربية:** تتيح للمستخدم طرح الأسئلة واستقبال الردود باللغة العربية الفصحى أو العامية، مع دعم الاتجاه من اليمين إلى اليسار RTL.
- **لوحة تحكم الويب:** منصة إدارية لمراقبة أداء النظام، وتتبع حالة الوكلاء، وإدارة المستخدمين والصلاحيات والأمان.
- **وكلاء الذكاء الاصطناعي:** شبكة من خمسة وكلاء متخصصين هي: الوكيل المشرف Supervisor، ووكيل SQL، ووكيل RAG، والوكيل المحلل Analyst، ووكيل التقارير Reporter، وتعمل بتنسيق عبر إطار LangGraph.
- **طبقة الرؤى الذكية:** طبقة وسيطة تبسط بنية قاعدة البيانات العلائقية لرفع دقة تحويل اللغة الطبيعية إلى استعلامات SQL.

ويرتبط النظام ببيئة Odoo Education بوصفها بيئة تطبيق واقعية، مع اعتماد نموذج لغوي مزدوج قابل للتبديل بين التشغيل المحلي عبر Ollama والتشغيل السحابي عبر API. ويركز هذا القسم على ما يفعله النظام، بينما تُترك تفاصيل كيفية عمله للفصل الرابع.

4.3.3 المتطلبات الوظيفية

تم تحديد ثلاثين متطلباً وظيفياً، وصُنفت في خمس فئات. وتحدد الأولوية باستخدام منهجية MoSCoW: عالية Must Have للوظائف الأساسية، ومتوسطة Should Have للوظائف المهمة القابلة للتأجيل مؤقتاً، ومنخفضة Could Have للوظائف التحسينية.

يوضح الشكل 3-2 توزيع المتطلبات الوظيفية حسب الأولوية. وتضم القائمة 21 متطلباً ذا أولوية عالية، و7 متطلبات متوسطة، ومتطلبين منخفضي الأولوية، بما يعكس تركيز الإصدار الحالي على الوظائف الأساسية.

للشكل مؤقت إطار

توزيع المتطلبات الوظيفية حسب الأولوية

Must Have: 21 Should Have: 7 Could Have: 2

الرسم توليد سيُعاد النهائي لاحقاً.

شكل 3-2: توزيع المتطلبات الوظيفية لنظام «بيان» حسب الأولوية.

LT

جدول 3-4: المتطلبات الوظيفية لنظام «بيان».

المعرّف	وصف المتطلب	الأولوية
	الفئة الأولى: المصادقة وإدارة المستخدمين	
FR-01	تسجيل الدخول باستخدام بيانات الاعتماد المؤسسية واسم المستخدم وكلمة المرور.	عالية
FR-02	إنشاء حسابات جديدة وتعيين الأدوار من قبل مدير النظام.	عالية
FR-03	تغيير كلمة المرور.	متوسطة
FR-04	تطبيق صلاحيات قائمة على الأدوار RBAC بأربعة مستويات أمنية.	عالية

جدول 3-4 -- تتمة

المعرّف	وصف المتطلب	الأولوية
FR-05	الاحتفاظ بسجل تدقيق شامل Audit Log لجميع العمليات.	عالية
الفئة الثانية: المحادثة وتحويل اللغة الطبيعية إلى SQL		
FR-06	توفير واجهة محادثة عربية تدعم الاتجاه RTL.	عالية
FR-07	دعم العربية الفصحى والعامية في فهم الأسئلة.	عالية
FR-08	توجيه السؤال إلى الوكيل المتخصص المناسب عبر الوكيل المشرف.	عالية
FR-09	تحويل السؤال العربي إلى استعلام SQL عبر طبقة الرؤى الذكية.	عالية
FR-10	تبسيط مخطط قاعدة البيانات قبل تمريره إلى النموذج اللغوي.	عالية
FR-11	تنفيذ استعلامات SQL على PostgreSQL وإرجاع النتائج.	عالية
FR-12	عرض مسار التنفيذ Agent Trace للمستخدم.	متوسطة
FR-13	التبديل بين النموذج اللغوي المحلي Ollama والسحابي API.	متوسطة
FR-14	معالجة أخطاء SQL النحوية بإعادة المحاولة باستخدام أمثلة Few-Shot.	عالية
FR-15	رفض الاستعلامات المخالفة لمستوى الأمان وتسجيلها.	عالية
الفئة الثالثة: الاسترجاع المعزز بالتوليد		
FR-16	البحث في المستندات المفهرسة في قاعدة Qdrant.	عالية
FR-17	تضمين المستندات المسترجعة في سياق النموذج لتوليد الإجابة.	عالية
FR-18	ربط الإجابة بمصادرها الأصلية Citation.	عالية
FR-19	فهرسة مستندات Odoo تلقائياً عند التهيئة.	متوسطة
FR-20	استخدام نموذج تضمين متعدد اللغات داعم للعربية multilingual-e5-large.	عالية
الفئة الرابعة: التحليل والتقارير		
FR-21	استخراج المؤشرات الإحصائية والاتجاهات.	عالية
FR-22	اكتشاف القيم الشاذة Outliers وتنبيه المستخدم.	متوسطة

جدول 3-4 -- تتمة

المعرّف	وصف المتطلب	الأولوية
FR-23	تنسيق المخرجات في تقارير نصية وجدولية ورسوم بيانية.	عالية
FR-24	تصدير التقارير بصيغ PDF وCSV وExcel.	متوسطة
FR-25	توليد رسوم بيانية تفاعلية عند الطلب.	منخفضة
الفئة الخامسة: الإدارة والتهيئة		
FR-26	توفير لوحة تحكم ويب لمراقبة أداء النظام.	عالية
FR-27	مراقبة نشاط الوكلاء في الوقت الفعلي.	متوسطة
FR-28	تنفيذ التهيئة الذكية لربط النظام بـ Odoo خلال 8 ساعات.	عالية
FR-29	تصنيف جداول قاعدة البيانات إلى مستويات أمنية.	متوسطة
FR-30	إرسال إشعارات داخلية عبر لوحة التحكم للأحداث المهمة.	منخفضة

5.3.3 المتطلبات غير الوظيفية

تمثل المتطلبات غير الوظيفية معايير الجودة والأداء التي تحدد كيفية تنفيذ النظام، لا الوظائف التي يقدمها. صُنفت المتطلبات وفق خصائص مختارة من معيار ISO/IEC 25010، وهي الأداء، والأمان، والموثوقية، وقابلية الاستخدام، وقابلية الصيانة والنقل. يوضح الشكل 3-3 توزيعها حسب خصائص الجودة.

شكل 3-3: توزيع المتطلبات غير الوظيفية حسب خصائص الجودة في معيار ISO/IEC 25010.

LT

جدول 3-5: المتطلبات غير الوظيفية وطرق اختبارها.

المعرّف	وصف المتطلب	طريقة الاختبار
الأداء		
NFR-01	زمن الاستجابة لا يتجاوز 3 ثوانٍ في 95% من الحالات.	اختبار الحمل لخمسة مستخدمين.
NFR-02	زمن التهيئة الذكية لا يتجاوز 8 ساعات لبيانات خمس سنوات.	قياس زمن التنفيذ.
NFR-03	دعم 10 مستخدمين متزامنين دون تدهور ملحوظ في الأداء.	اختبار الحمل.
NFR-04	استهلاك الذاكرة لا يتجاوز 32 GB RAM عند تشغيل النموذج المحلي.	مراقبة الموارد.
الأمان		
NFR-05	تشفير كلمات المرور باستخدام bcrypt بمعامل كلفة 12.	فحص قيمة التجزئة.
NFR-06	تطبيق التحكم القائم على الأدوار بأربعة مستويات أمنية.	اختبار محاولات الوصول.
NFR-07	تسجيل العمليات في سجل تدقيق شامل.	فحص السجل.
NFR-08	نقل البيانات عبر HTTPS/TLS 1.2+.	فحص الشهادة.
NFR-09	الحماية من حقن SQL باستخدام الاستعلامات ذات المعاملات.	مراجعة الشفرة.
الموثوقية		
NFR-10	نسبة توفر النظام لا تقل عن 95% خلال فترة الاختبار.	مراقبة الاستجابة.
NFR-11	التعافي من الأعطال خلال 60 ثانية.	محاكاة الانقطاع.
NFR-12	الاحتفاظ بحالة المحادثة عند فشل أحد الوكلاء.	محاكاة فشل وكيل.
قابلية الاستخدام		
NFR-13	تحقيق درجة SUS لا تقل عن 70 من 100.	استبيان SUS.
NFR-14	إتمام 90% من المستخدمين للمهمة خلال أقل من دقيقتين.	اختبار المستخدم.

جدول 3-5 -- تتمة

المعرّف	وصف المتطلب	طريقة الاختبار
NFR-15	توفير واجهة عربية داعمة للاتجاه RTL.	مراجعة بصرية.
قابلية الصيانة والنقل		
NFR-16	توثيق ما لا يقل عن 70% من الشفرة البرمجية.	أداة pydocstyle.
NFR-17	اتباع معايير تنسيق PEP 8.	أداة flake8.
NFR-18	النشر باستخدام Docker Compose في خطوة واحدة.	اختبار النشر.
NFR-19	التوافق مع بيئتي Linux وWindows.	اختبار الحاويات.
NFR-20	تبديل النموذج اللغوي دون تعديل الشفرة.	اختبار التبديل.

6.3.3 متطلبات بيئة التشغيل التقنية

تختلف متطلبات بيئة التشغيل الفعلية عن أدوات التطوير، ولذلك قُسمت إلى متطلبات مادية وبرمجية كما يأتي.

جدول 3-6: متطلبات الخادم المادية.

المكون	المواصفات	الغرض والاستخدام
المعالج	Intel Core i7 أو ما يعادله، 8 أنوية على الأقل	تشغيل خدمات الواجهة الخلفية وإدارة الطلبات المتزامنة.
الذاكرة	32 GB كحد أدنى	تشغيل PostgreSQL وQdrant وRedis والنموذج المحلي عند الحاجة.
بطاقة الرسوميات	NVIDIA RTX 3060 بذاكرة 12 GB، اختيارية	تسريع النموذج المحلي عبر Ollama.
التخزين	1 TB SSD	تخزين قاعدة بيانات Odoo والمستندات المفهرسة والنماذج.
الشبكة	اتصال مستقر بسرعة لا تقل عن 25 Mbps	الاتصال بالواجهة السحابية ورفع المستندات.

يوضح الشكل 3-4 معمارية النشر المستهدفة عبر Docker Compose. ويُدرج الشكل حالياً بوصفه إطاراً مؤقتاً إلى حين إعادة توليد الرسم النهائي.

جدول 3-7: متطلبات بيئة التشغيل البرمجية.

المكون	التقنية المعتمدة	الغرض والاستخدام
نظام التشغيل	Ubuntu 22.04 LTS	الاستقرار والدعم طويل الأمد في الإنتاج.
منصة الحاويات	Docker Compose	تشغيل الخدمات في بيئة معزولة وموحدة.
قاعدة البيانات العلائقية	PostgreSQL 16 + pgvector	تخزين بيانات Odoo والمتجهات.
قاعدة البيانات المتجهة	Qdrant	تخزين واسترجاع تضمينات المستندات.
التخزين المؤقت	Redis	إدارة الجلسات وطابور رسائل تنسيق الوكلاء.
الواجهة الخلفية	Python 3.12 + FastAPI	تطوير واجهات API وخدمات الوكلاء.
الواجهة الأمامية	Node.js 18+ + Next.js	واجهة المحادثة العربية ولوحة التحكم.
النماذج اللغوية	Ollama محلياً وDeepSeek/GPT-4o API سحابياً	إتاحة التبديل بين التشغيل المحلي والسحابي.

شكل 3-4: معمارية نشر نظام «بيان» عبر Docker Compose (إطار مؤقت).

7.3.3 قيود ومحددات النظام

تمثل القيود الحدود التقنية والتشغيلية التي يعمل ضمنها النظام في نسخته الحالية، ويجب أخذها في الاعتبار عند تفسير النتائج وتحديد نطاق الاستخدام.

جدول 3-8: قيود ومحددات نظام «بيان».

نوع القيد	وصف القيد
اللغة	يدعم النظام العربية الفصحى والعامية، إلا أن دقة فهم اللهجات المحلية تعتمد على النموذج اللغوي وقد تتطلب إعادة صياغة السؤال.
البيانات	تعتمد جودة الإجابات على جودة واكتمال بيانات Odoo، ولا يصحح النظام أخطاء البيانات المصدرية.
الأمان	لا يدعم الإصدار الحالي المصادقة الثنائية، وتُطبق الصلاحيات على مستوى الجداول لا الصفوف.
بيئة البيانات	استُخدم Odoo Education ببيانات اصطناعية تعادل خمس سنوات بسبب صعوبة الوصول إلى قواعد ERP الحية، حفاظاً على الخصوصية.
الأداء	صُمم النظام لدعم 10 مستخدمين متزامنين في بيئة الاختبار، وقد يتدهور الأداء عند الأحمال الأعلى، خاصة مع النموذج المحلي.
النموذج اللغوي	تعتمد دقة NL2SQL على جودة المطالبات وطبقة الرؤى الذكية؛ وقد تفشل الأسئلة الغامضة أو المعقدة جداً.
التهيئة	تتطلب التهيئة وصولاً مباشراً إلى قاعدة Odoo وصلاحيات قراءة كافية، وقد تستغرق حتى 8 ساعات للبيانات الكبيرة.
المستندات	تدعم الفهرسة صيغ PDF وDOCX وTXT، بينما قد لا تُفهرس المستندات الممسوحة دون OCR بصورة صحيحة.

8.3.3 افتراضات النظام

تمثل الافتراضات الشروط المسبقة التي بُني عليها تحليل النظام وتصميمه، ويتطلب التشغيل الصحيح تحققها. ويعرض الجدول 3-9 هذه الافتراضات بصورة موجزة.

جدول 3-9: افتراضات نظام «بيان».

نوع الافتراض	الوصف
المستخدمون	يُفترض أن المستخدم قادر على صياغة أسئلة واضحة بالعربية ويمتلك معرفة أساسية بواجهات المحادثة.
البيانات	يُفترض أن خادم Odoo يعمل بكفاءة وأن بياناته سليمة ومحدثة؛ إذ يعتمد النظام عليها كما هي.
الشبكة	يُفترض استقرار الاتصال بالإنترنت عند استخدام الواجهة السحابية؛ فقد يؤثر الانقطاع المتكرر في الاستجابة.
الأمان	يُفترض أن يصنف مدير النظام الجداول إلى مستويات Public, Masked, Restricted, Blacklist بدقة، وأن يمنح الصلاحيات المناسبة.
النموذج	يُفترض أن النموذج اللغوي المحلي أو السحابي يدعم العربية بكفاءة كافية لفهم الأسئلة وتوليد الاستعلامات.
التهيئة	يُفترض توفير بيانات اعتماد Odoo الصحيحة، بما في ذلك عنوان الخادم وقاعدة البيانات واسم المستخدم وكلمة المرور.

9.3.3 خلاصة تحليل المتطلبات

حدد هذا القسم نطاق نظام «بيان» من منظور أصحاب المصلحة والوظائف والجودة وبيئة التشغيل والقيود والافتراضات. وتشكل المتطلبات الوظيفية الثلاثون وغير الوظيفية العشرون أساساً قابلاً للتحقق لمراحل التصميم والتنفيذ والاختبار، في حين تساعد متطلبات البيئة والقيود والافتراضات على ضبط التوقعات وتفسير نتائج التقييم. وبذلك ينتقل العمل في القسم التالي إلى نمذجة تفاعل المستخدمين مع النظام من خلال حالات الاستخدام.

4.3 حالات الاستخدام

5.3 تحليل المخاطر

6.3 ملخص الفصل

الفصل

الرابع

النظام تصميم

1.4 المعمارية العامة للنظام

2.4 تصميم قواعد البيانات

3.4 تصميم واجهات المستخدم

4.4 مخططات النظام

1.4.4 مخططات التدفق

2.4.4 مخططات التتابع

3.4.4 مخططات ER

5.4 تصميم واجهات برمجة التطبيقات (APIs)

6.4 ملخص الفصل

الفصل

الخامس

05

النظام واختبار تنفيذ

1.5 بيئة التطوير والأدوات

2.5 تنفيذ النظام

3.5 النشر والتشغيل

4.5 اختبار النظام

1.4.5 اختبارات الوحدة (Unit Tests)

2.4.5 اختبارات التكامل (Integration Tests)

3.4.5 اختبارات الأداء

4.4.5 اختبار قبول المستخدم (UAT)

5.5 الخطة الزمنية للتنفيذ

6.5 ملخص الفصل

الفصل

السادس

والمناقشة النتائج

1.6 النتائج

1.1.6 النتائج الكمية

2.1.6 النتائج النوعية

2.6 واجهات النظام النهائية

3.6 مناقشة النتائج

4.6 التحديات والحلول

5.6 الدروس المستفادة

6.6 ملخص الفصل

الفصل

السابع

المستقبلية والأعمال الخاتمة

1.7 الخاتمة

2.7 ملخص تحقيق الأهداف

3.7 مساهمات المشروع

4.7 قيود النظام

5.7 التوصيات

6.7 الأعمال المستقبلية

7.7 ملخص الفصل

ملحق أ: قاموس المصطلحات

ملحق ب: قائمة المتطلبات الكاملة

ملحق ج: كود مصدري مختار

ملحق د: لقطات شاشة إضافية

ملحق هـ: استبيان تقييم المستخدمين

ملحق و: توزيع أدوار الفريق

المراجع

المصادر

[1] United Nations, ``E-Government Survey 2024: Accelerating Digital Transformation for Sustainable Development," *United Nations Department of Economic and Social Affairs*, 2024. [Online]. Available: https://publicadministration.un.org/egovkb/

[2] Vanna.ai, ``Vanna: Open-Source Python RAG Framework for SQL Generation," 2024. [Online]. Available: https://vanna.ai

[3] Snowflake AI Research Team, ``Arctic-Text2SQL-R1: A Reinforcement-Learned Text-to-SQL Model," 2025. [Online]. Available: https://github.com/Snowflake-Labs/arctic

[4] T. Yu, R. Zhang, K. Yang, M. Yasunaga, D. Wang, Z. Li, J. Ma, I. Li, Q. Yao, S. Yu, T. Zhang, and D. Radev, ``Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task," in *Proc. EMNLP*, 2018, pp. 3911--3921.

[5] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, ``Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 9459--9474.

[6] M. El-Haj, ``Arabic Natural Language Processing: Challenges and Opportunities," *arXiv preprint* arXiv:2307.10543, 2023.

[7] LangChain, ``LangGraph: Building Stateful Multi-Actor Applications with LLMs," 2024. [Online]. Available: https://github.com/langchain-ai/langgraph

[8] Qdrant Team, ``Qdrant: Vector Similarity Search Engine," 2024. [Online]. Available: https://qdrant.tech

[9] LangChain, ``LangChain: Building Applications with LLMs through Composability," 2023. [Online]. Available: https://github.com/langchain-ai/langchain

[10] LlamaIndex, ``LlamaIndex: Data Framework for LLM Applications," 2023. [Online]. Available: https://www.llamaindex.ai

[11] Q. Wu, G. Bansal, J. Zhang, Y. Wu, B. Li, E. Zhu, L. Jiang, X. Zhang, S. Zhang, J. Liu, *et al.*, ``AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation," *arXiv preprint* arXiv:2308.08155, 2023.

[12] CrewAI Inc., ``CrewAI: Framework for Orchestrating Role-Playing Autonomous AI Agents," 2024. [Online]. Available: https://github.com/joaomdmoura/crewAI

[13] S. Hong, M. Zhuge, J. Chen, X. Zheng, Y. Cheng, C. Zhang, J. Wang, Z. Wang, S. K. S. Yau, Z. Lin, *et al.*, ``MetaGPT: Meta Programming for A Multi-Agent Collaborative Framework," *arXiv preprint* arXiv:2308.00352, 2023.

[14] H. S. Al-Khalifa and M. Al-Razgan, ``Arabic Educational Chatbots: A Review of Current Status and Future Directions," *Journal of Educational Technology & Society*, vol. 26, no. 2, 2023.

[15] W. Magdy, K. Darwish, *et al.*, ``Arabic Information Retrieval: A Survey of the State of the Art," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 23, no. 3, pp. 1--38, 2024.

[16] OpenAI, ``GPT-4 Technical Report," *arXiv preprint* arXiv:2303.08774, 2023.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, ``Attention Is All You Need," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017, pp. 5998--6008.

[18] G. Poesia, A. Cambronero, S. Hou, L. Lambert, et al., ``Synthesizing Text-to-SQL with Relational Reasoning and Self-Correction," in *Proc. International Conference on Learning Representations (ICLR)*, 2024.