بحث مقدَّم ضمن متطلبات مشروع التخرج — هندسة البرمجيات وتصميم الأنظمة

تصميم وتطوير منصة «Nawa360» كنواة برمجية قابلة لإعادة الاستخدام لبناء تطبيقات الأعمال

نواة واحدة... تطبيقات بلا حدود

إعداد الطالب: **بسام مسمار** إشراف: **د. أسامة أبو سحلة** العام الجامعي: **2025 / 2026**

من ثلاث منصات متفرقة... إلى فكرة واحدة

بدأت الفكرة من تجربة عملية حقيقية امتدت قرابة عام، تكررت خلالها البنية الأساسية نفسها في كل مشروع جديد.

1

منصة تعليمية

بناء نظام كامل من الصفر: مستخدمون، صلاحيات، لوحة تحكم.

2

منصة تجارة إلكترونية

إعادة بناء الأساس نفسه مرة أخرى مع اختلاف منطق العمل فقط.

3

نظام إدارة شحن

التكرار للمرة الثالثة: نفس المكونات، نفس سير العمل.

4

الملاحظة الجوهرية

الجزء الأكبر من العمل البرمجي يتكرر في كل مشروع.

مع ظهور **وكلاء الذكاء الاصطناعي** وانتشار نموذج **SaaS**، أصبح تحويل هذه الملاحظة إلى حل عام أمرًا ممكنًا — فوُلدت فكرة Nawa360.

مشكلة البحث

تطوير أي نظام إلكتروني جديد يتطلب إعادة بناء المكونات الأساسية نفسها في كل مرة، رغم أنها مشتركة بين معظم التطبيقات:

تسجيل الدخول | إدارة المستخدمين | الأدوار والصلاحيات | لوحة التحكم | التقارير | الإعدادات | الإشعارات

قواعد البيانات | سير العمل

ما الذي يترتب على هذا التكرار؟

- زيادة وقت التطوير وارتفاع التكلفة.
- تكرار الأخطاء نفسها في كل مشروع.
- صعوبة الصيانة وضعف الاستفادة من الخبرات السابقة.

سؤال البحث

كيف يمكن تصميم نواة برمجية موحّدة تحتوي المتطلبات المشتركة، وتُبنى فوقها تطبيقات أعمال مختلفة دون البدء من الصفر في كل مرة؟

الحل المقترح: منصة Nawa360

نواة برمجية موحّدة تحتوي المكونات المشتركة بين الأنظمة، تُبنى فوقها تطبيقات متخصصة، مع متجر تطبيقات وصانع تطبيقات مدعوم بالذكاء الاصطناعي.

النواة الجاهزة

كل الأساسيات المشتركة مبنية مسبقًا ومختبرة: مستخدمون، صلاحيات، لوحة تحكم، تقارير، إشعارات.

متجر التطبيقات

تطبيقات جاهزة تُفعَّل داخل مساحة المستخدم حسب حاجته: شحن، تجارة، تعليم، موارد بشرية.

صانع التطبيقات الذكي

يكتب المستخدم فكرته بلغته الطبيعية، فيحلّلها وكيل الذكاء الاصطناعي ويحوّلها إلى مواصفات تقنية منظمة.

الدور الدقيق للذكاء الاصطناعي: **محلّل ومساعد صياغة** — يحوّل الفكرة إلى متطلبات ومواصفات، ثم تُنشئ المنصة التطبيق اعتمادًا على النواة والقوالب والبنية المحددة، لا على كود حر.

أهداف البحث

الهدف العام

تصميم وتطوير منصة **Nawa360** كنواة برمجية قابلة لإعادة الاستخدام، تقلّل تكرار بناء المكونات الأساسية عند تطوير أنظمة الأعمال المختلفة.

الأهداف الفرعية

1. تحليل المتطلبات المشتركة بين أنواع مختلفة من التطبيقات.
2. تصميم نواة موحّدة تضم المكونات الأساسية (المستخدمون، الصلاحيات، لوحة التحكم، التقارير...).
3. بناء صانع تطبيقات يحوّل فكرة المستخدم إلى مواصفات تقنية عبر الذكاء الاصطناعي.
4. توفير متجر تطبيقات لتفعيل الوحدات المتخصصة حسب حاجة المستخدم.
5. تطبيق ضوابط تعزل التطبيقات المولّدة عن نواة النظام وتحمي البيانات.
6. دعم نموذجَي التشغيل: SaaS والاستضافة الذاتية Self-hosted.

أهمية البحث

أهمية علمية

يربط بين هندسة البرمجيات، وتحليل المتطلبات، وإعادة استخدام المكونات، والأنظمة السحابية، والذكاء الاصطناعي في إطار واحد.

أهمية عملية

يقدّم نموذجًا حقيقيًا قابلاً للاستخدام والتجربة، مبنيًا على خبرة تطوير فعلية، لا فكرة نظرية مجردة.

أهمية اقتصادية

يقلّل وقت التطوير وتكلفة بناء الأنظمة وتكرار الأخطاء، ويغني عن بناء كل نظام من الصفر.

حدود البحث

نطاق واضح ومحدد يضمن نموذجًا أوليًا مكتملاً وقابلاً للعرض، دون توسّع يضر بجودة التنفيذ.

ضمن نطاق البحث

- ✓ نموذج أولي لمنصة Nawa360 والنواة الأساسية.
- ✓ صانع تطبيقات مبسّط مدعوم بالذكاء الاصطناعي.
- ✓ متجر تطبيقات كمفهوم عملي أولي.
- ✓ تطبيق أو تطبيقان نموذجيان (الشحن أو المتجر).
- ✓ الذكاء الاصطناعي كمساعد تحليل وتخصيص فقط.

خارج النطاق — يُذكر في التوصيات

- بيع التطبيقات بين المستخدمين وتقاسم الأرباح.
- محفظة مزودي الخدمات والدفع الإلكتروني.
- النشر العام وتقييم التطبيقات.
- وكلاء يكتبون كودًا كاملاً وينشرونه تلقائيًا.

الحد الموضوعي: النواة الأساسية وصانع التطبيقات ومتجر أولي فقط، دون الميزات المتقدمة.

الحد الزماني: نُفِّذ المشروع خلال العام الجامعي 2025 / 2026.

الحد التطبيقي: تطبيق أو تطبيقان نموذجيان (الشحن / المتجر) كإثبات مفهوم.

منهجية البحث

لأن المشروع ذو طبيعة **تطويرية** (بناء نظام) لا دراسة مسحية، اعتُمد **منهج تطوير النظم (SDLC)** بأسلوب تكراري تزايدي: تُبنى النواة أولاً، ثم تُركَّب التطبيقات فوقها.

1. تحليل المشكلة وجمع المتطلبات المشتركة.
2. تصميم بنية النواة وطبقة التطبيقات.
3. تنفيذ النواة ثم صانع التطبيقات والمتجر.
4. اختبار الوظائف وعزل التطبيقات عن النظام.
5. تقييم النتائج وتحديد التطوير المستقبلي.

أدوات البحث والتقنيات

- Django / Python
- قواعد بيانات معزولة لكل مستخدم (Multi-tenant)
- وكيل ذكاء اصطناعي للتحليل
- بيئة تشغيل معزولة (Sandbox)
- مواصفات منظمة (App Manifest)

هيكل البحث: ستة فصول

الجانب النظري

الفصل الأول ▪ الإطار العام: المقدمة، المشكلة، الأهداف، الأهمية، الحدود، المنهجية.

الفصل الثاني ▪ تصميم الأنظمة وهندسة البرمجيات، دورة حياة التطوير، والدراسات السابقة.

الفصل الثالث ▪ تحليل المتطلبات الوظيفية وغير الوظيفية وسير العمل.

الفصل الرابع ▪ نموذج SaaS، الذكاء الاصطناعي، ومفهوم متجر التطبيقات.

الجانب العملي

الفصل الخامس ▪ النموذج التطبيقي لمنصة Nawa360: النواة، الصانع، المتجر، التقنيات.

الفصل السادس ▪ النتائج، والتحديات، والتوصيات، والتطويرات المستقبلية.

محاور الجانب النظري

تصميم الأنظمة ودورة حياة التطوير

من تحليل المشكلة وجمع المتطلبات، إلى التصميم والبرمجة والاختبار، ثم النشر والصيانة والتطوير.

هندسة البرمجيات وإعادة الاستخدام

التخطيط قبل البرمجة، تقليل الأخطاء، قابلية الصيانة والتوسع، وإعادة استخدام المكونات كأساس علمي للمشروع.

نموذج SaaS مقابل الاستضافة الذاتية

SaaS: النظام خدمة عبر المتصفح تديرها المنصة.
Self-hosted: تثبيت على خادم خاص بالعميل. المنصة تدعم النموذجين.

وكلاء الذكاء الاصطناعي

لم يعد الذكاء الاصطناعي مجيبًا عن الأسئلة فحسب؛ بل مساعدًا في تحليل المتطلبات، واقتراح الواجهات، وبناء سير العمل.

الدراسات والأنظمة السابقة

اطّلع البحث على أبرز منصات بناء التطبيقات القائمة لتحديد موقع Nawa360 منها والفجوة التي يسدّها.

Bubble / Adalo

بناء تطبيقات بصريًا دون كود، لكنها منصات مغلقة تُنتج تطبيقات لا يملك المستخدم بنيتها، ومحدودة في التخصيص العميق.

OutSystems / Mendix

منصات Low-Code مؤسسية قوية، لكنها مكلفة ومعقّدة وموجّهة للمطورين المحترفين لا للمستخدم غير التقني.

Odoo

نواة أعمال مفتوحة بوحدات جاهزة، لكن إضافة تطبيق جديد تتطلب خبرة برمجية، ولا يوجد توليد ذكي من فكرة نصية.

Zoho Creator

إنشاء تطبيقات أعمال بسرعة، لكنها خدمة مغلقة بلا استضافة ذاتية ولا وكيل ذكاء اصطناعي يحوّل الفكرة إلى مواصفات.

الفجوة البحثية: تجمع Nawa360 في منصة واحدة بين النواة القابلة لإعادة الاستخدام، والتوليد الذكي من فكرة بلغة طبيعية، وعزل التطبيقات عن النظام، ودعم SaaS والاستضافة الذاتية معًا — وهو ما لا يتوفّر مجتمِعًا في المنصات السابقة.

تحليل المتطلبات

المتطلبات الوظيفية

ماذا يفعل النظام؟

- ✓ تسجيل الدخول وإدارة المستخدمين والصلاحيات.
- ✓ إنشاء تطبيق جديد عبر صانع التطبيقات.
- ✓ تفعيل تطبيق جاهز من متجر التطبيقات.
- ✓ لوحة تحكم وتقارير وإشعارات لكل مساحة.
- ✓ تعدد العملاء: مساحة وقاعدة بيانات مستقلة لكل مستخدم.
- ✓ دعم نسختي SaaS وSelf-hosted.

المتطلبات غير الوظيفية

كيف يعمل النظام وبأي جودة؟

- ✓ الأمان وحماية البيانات وعزل مساحات المستخدمين.
- ✓ الأداء وسرعة الاستجابة.
- ✓ سهولة الاستخدام لغير التقنيين.
- ✓ قابلية التوسع والصيانة.
- ✓ الاعتمادية والنسخ الاحتياطي.
- ✓ قابلية النشر على خوادم مختلفة.

البنية العامة للمنصة

طبقة التطبيقات — تُبنى فوق النواة حسب النشاط

إدارة الشحن | المتجر الإلكتروني | المنصة التعليمية | الموارد البشرية | المحاسبة والفواتير | إدارة العملاء CRM

▲ تعتمد جميعها على ▼

النواة الأساسية Core — المكونات المشتركة بين كل الأنظمة

المستخدمون | الأدوار والصلاحيات | المصادقة | لوحة التحكم | الإعدادات | الإشعارات | التقارير | سجلات التدقيق

متجر التطبيقات

تفعيل تطبيقات جاهزة داخل مساحة المستخدم، مع مراجعة قبل النشر.

مساعد الذكاء الاصطناعي

تحليل الطلبات، اقتراح الوحدات، بناء سير العمل، والمساعدة في التقارير.

نماذج النشر

نسخة SaaS سحابية | نسخة Self-hosted مستقلة

الشكل (1): البنية العامة لمنصة Nawa360 — النواة، طبقة التطبيقات، خدمات المنصة، ونماذج النشر.

النواة الأساسية: كل الأساسيات جاهزة مسبقًا

مكونات مبنية مرة واحدة ومختبرة، يرثها كل تطبيق جديد تلقائيًا:

المستخدمون والأدوار · حسابات وصلاحيات دقيقة لكل دور في النظام.

الإعدادات · إعدادات النظام واللغة قابلة للتخصيص.

سجلات التدقيق · توثيق كل عملية: مَن فعل ماذا ومتى.

الحماية والأمان · عزل البيانات وضبط الوصول على مستوى المنصة.

المصادقة · تسجيل دخول آمن موحّد لجميع التطبيقات.

الإشعارات · تنبيهات داخل النظام لكل الأحداث المهمة.

إدارة الملفات · رفع الملفات وتنظيمها داخل المساحة.

لوحة التحكم · واجهة إدارة مركزية لكل مساحة عمل.

التقارير · استخراج تقارير جاهزة لكل تطبيق.

تعدد العملاء · مساحة وقاعدة بيانات مستقلة لكل مستخدم.

صانع التطبيقات: اكتب فكرتك... ودع المنصة تفهمها

1. يكتب المستخدم فكرته بلغته الطبيعية.
2. تطرح المنصة أسئلة توضيحية بسيطة.
3. يحلّل وكيل الذكاء الاصطناعي الإجابات ويعيد الصياغة.
4. تُولَّد مواصفات منظمة (App Manifest): جداول، حقول، أدوار، تقارير.
5. يراجع المستخدم ملخصًا مفهومًا ويعتمد الإنشاء.
6. يُنشأ التطبيق داخل مساحته المستقلة.

«أريد نظامًا لإدارة الشحنات والعملاء والسائقين والفواتير.»

المنصة تسأل: هل تريد تتبع حالة الشحنة؟ هل توجد صلاحيات مختلفة؟ هل تريد تقارير شهرية؟

المواصفات الناتجة: وحدات العملاء والسائقين والشحنات والفواتير · حالات الشحنة · أدوار (مدير، محاسب، عمليات) · تقرير شهري.

لا يحتاج المستخدم إلى أي معرفة تقنية، ولا يكتب الوكيل كودًا حرًّا؛ بل يمرّ كل شيء عبر مواصفات منظمة تتحول إلى تطبيق وفق بنية المنصة وقوالبها — بأمان وقابلية للمراجعة.

متجر التطبيقات

تطبيقات جاهزة تُفعَّل داخل مساحة المستخدم بنقرة، بدل بنائها من الصفر —
ومستقبلاً يعرض المطورون ومزودو الخدمات تطبيقاتهم فيه.

إدارة الشحن | المتجر الإلكتروني | المنصة التعليمية | الموارد البشرية | المحاسبة والفواتير | إدارة علاقات العملاء

لا يصل تطبيق إلى المتجر إلا بعد المراجعة

1. فحص الأمان وخصوصية البيانات.
2. فحص الجودة والتوافق مع بنية المنصة.
3. اعتماد إدارة المنصة قبل النشر.

الضوابط والإجراءات الوقائية

القاعدة الذهبية: **النواة الأساسية لا يلمسها وكيل الذكاء الاصطناعي** — يعمل الوكيل داخل مساحة معزولة ومحددة فقط.

- ✓ فصل تام بين كود النواة وكود التطبيقات المولَّدة.
- ✓ الوكيل يُنتج مواصفات (App Manifest) لا كودًا حرًّا.
- ✓ لا تنفيذ قبل مراجعة المستخدم واعتماده الصريح.
- ✓ منع الأوامر الخطرة: حذف قواعد البيانات، تعديل النظام، الوصول لبيانات مستخدم آخر.
- ✓ الإنشاء داخل بيئة معزولة (Sandbox) لا في النسخة الحية.
- ✓ سجل تدقيق يوثّق كل خطوة: الطلب، التحليل، الإنشاء، الاعتماد.

سيناريو العرض العملي أمام اللجنة

عرض حي يمر بدورة كاملة: من فكرة مكتوبة بلغة طبيعية إلى تطبيق يعمل داخل مساحة مستقلة.

1. إنشاء حساب جديد على المنصة.
2. تجهيز مساحة عمل وقاعدة بيانات مستقلة تلقائيًا.
3. كتابة فكرة تطبيق إدارة الشحن بلغة بسيطة.
4. الإجابة عن أسئلة صانع التطبيقات.
5. مراجعة المواصفات التي حلّلها الوكيل الذكي.
6. اعتماد الإنشاء ومشاهدة التطبيق يتكوّن.
7. فتح التطبيق وتجربته: لوحة تحكم، بيانات، صلاحيات، تقارير.
8. جولة في متجر التطبيقات وتفعيل تطبيق جاهز.

النتائج والمناقشة

حقّق النموذج الأولي أهداف البحث الأساسية، وأثبت جدوى فكرة النواة القابلة لإعادة الاستخدام عمليًا.

ما تحقّق في النموذج الأولي

- ✓ نواة أساسية عاملة: مستخدمون، صلاحيات، لوحة تحكم، تقارير.
- ✓ عزل مساحة وقاعدة بيانات مستقلة لكل مستخدم.
- ✓ صانع تطبيقات يحوّل فكرة نصية إلى مواصفات منظمة.
- ✓ متجر تطبيقات أولي لتفعيل الوحدات.
- ✓ فصل تام بين كود التطبيقات وكود النواة.

مناقشة النتائج

- تقليل ملموس في زمن التأسيس؛ الأساس جاهز بدل بنائه من الصفر.
- تمكين غير التقنيين من وصف أنظمتهم بلغتهم الطبيعية.
- أبرز تحدٍّ: ضبط مخرجات الوكيل الذكي ومنع خلط الأكواد.
- تتوافق النتائج مع مبدأ إعادة الاستخدام في أدبيات هندسة البرمجيات.

خارطة التطوير المستقبلي

ميزات خارج نطاق البحث الحالي، تُذكر في فصل التوصيات كامتداد طبيعي للمنصة:

- بيع التطبيقات بين المستخدمين وتقاسم الأرباح مع منشئيها.
- محفظة مالية لمزودي الخدمات ونظام دفع إلكتروني.
- تقييم التطبيقات ومراجعات المستخدمين داخل المتجر.
- توسيع قدرات وكلاء الذكاء الاصطناعي ضمن الضوابط الأمنية نفسها.
- لوحة إدارة متقدمة للمنصة ومراجعي التطبيقات.

المراجع

وفق نظام التوثيق الأكاديمي APA.

Chong, F., & Carraro, G. (2006). *Architecture strategies for catching the long tail*. Microsoft Corporation.

Sahay, A., Indamutsa, A., Di Ruscio, D., & Pierantonio, A. (2020). Supporting the understanding and comparison of low-code development platforms. In *46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)* (pp. 171–178). IEEE.

Brown, T. B., et al. (2020). Language models are few-shot learners. In *Advances in Neural Information Processing Systems* (Vol. 33, pp. 1877–1901).

Sommerville, I. (2016). *Software engineering* (10th ed.). Pearson.

Pressman, R. S., & Maxim, B. R. (2020). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill.

Krueger, C. W. (1992). Software reuse. *ACM Computing Surveys, 24*(2), 131–183.

Mell, P., & Grance, T. (2011). *The NIST definition of cloud computing* (Special Publication 800-145). National Institute of Standards and Technology.

لا تبني Nawa360 تطبيقًا واحدًا،
بل **نواة** تُبنى فوقها تطبيقات متعددة،
بطريقة منظمة وآمنة وقابلة للتوسع.

شكرًا لحسن استماعكم — ونرحّب بأسئلة اللجنة الموقّرة.