



جامعة بوليتكنك فلسطين

كلية تكنولوجيا المعلومات وهندسة الحاسوب

دائرة علم الحاسوب وتكنولوجيا المعلومات

مشروع تخرج

بعنوان :

منصة إدارة العقارات والخدمات المهنية في فلسطين

إعداد الطلبة:

حازم عمايرة

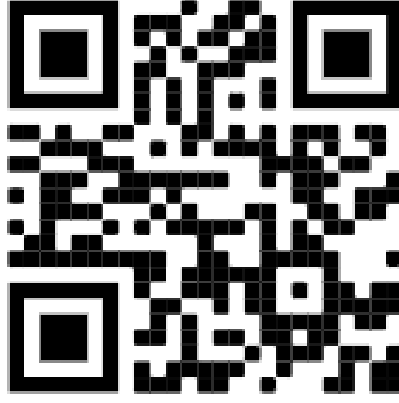
سيف الكومي

صامد سويطي

إشراف:

د.ازدهار جوابرة

2025-2024



لزيارة موقعنا اليكم QR CODE

ايضا الرابط :

aqaraty.netlify.app

بسم الله الرحمن الرحيم

الإهداء

إلى والدينا الأعمام

إلى من زرعوا فينا بذور الإيمان والثقة، إلى من حملونا على أكتافهم ودفعونا نحو النجاح دون كلل أو ملل، بكم وبفضل دعواتكم وصلنا إلى هذه اللحظة التي طالما حلمنا بها.

إلى إخواننا وأخواتنا

إلى من كانوا لنا شعلة الأمل ومصدر القوة، إلى من تقاسموا معنا الدرب بكل تفاصيله، وضحكاتهم كانت عزاءً لنا في لحظات التعب، كنتم الدافع الذي رفعنا كلما تعثرنا.

إلى أنفسنا

إلى أنفسنا التي لم تياس ولم تستسلم رغم الصعوبات، إلى كل لحظة اجتهدنا فيها وكل خطوة قطعناها على طريق تحقيق الحلم، تستحق أنفسنا منا الشكر والامتنان.

إلى أساتذتنا الأفاضل

إلى من كانوا منارةً نهتدي بها في مسيرتنا العلمية، إلى من أضأوا لنا الطريق بحروف العلم والمعرفة، وفتحوا لنا أبواباً من الأمل والنجاح.

إلى كل من دعمنا

إلى كل من كان جزءاً من رحلتنا، من دعوة صادقة، كلمة مشجعة، أو لحظة عطاء، نهدي هذا المشروع لكم جميعاً، شكراً لكم من القلب.

Certification and Anti-Plagiarism Declaration

This is to declare that the graduation project produced under the supervision of Ezdehar Jawabreh, having the title (منصة إدارة العقارات والخدمات المهنية في فلسطين), was prepared by the student/s below

In partial fulfilment of the requirements for the degree of Bachelor in Information Technology, and no part hereof has been reproduced illegally

(in particular: cut and paste), which can be considered as **Plagiarism**.

All referenced parts have been used to support and argue the idea and have been cited properly. I/We certify that I/we have not committed any plagiarism, cheating,

or any other academic integrity violation. We will be responsible and liable for any consequences if violation of this declaration is violated.

Any use of Artificial Intelligence (AI) tools in the preparation of this project has been clearly declared and documented in the related appendix at the end of the report, in accordance with the PPU guidelines.

Date: 2025/5/30

Graduation project group's student(s):

Name: Hazem Amayreh

Name: Saifaldin komi

Name: Samd Swaity

Abstract

The rapid growth of the real estate market and the increasing reliance on online platforms for conducting property transactions have introduced several challenges in managing property listings and related services. These challenges include difficulties in providing accurate and detailed information about properties, ensuring smooth communication between buyers and property owners, and connecting users with reliable service providers to meet their real estate needs efficiently.

To address these challenges, we developed an innovative digital platform dedicated to the real estate sector in Palestine. The platform simplifies the process of listing real estate ads, including land, apartments, and commercial spaces, where property owners can provide detailed information such as location, price, and images. Users can browse available properties, communicate directly with the owners, and submit inquiries or requests.

Additionally, the platform enables seamless integration with service providers such as electricians, plumbers, and engineers to deliver complete real estate solutions. It also includes features for submitting complaints and user reviews, along with a user-friendly interface that enhances transparency and trust, ultimately simplifying real estate transactions and improving the overall user experience. The system is also supported by an interactive map feature that accurately displays property locations and helps users explore and search more efficiently.

ملخص

أدى التطور الكبير في سوق العقارات وزيادة الاعتماد على المنصات الإلكترونية لإجراء المعاملات العقارية إلى ظهور تحديات عديدة في إدارة الإعلانات العقارية والخدمات المرتبطة بها. تشمل هذه التحديات صعوبة توفير معلومات دقيقة ومفصلة عن العقارات، وضمان التواصل السلس بين المشتريين وأصحاب العقارات، وربط المستخدمين بمقدمي الخدمات الموثوقين لتلبية احتياجاتهم العقارية بكفاءة.

لمعالجة هذه التحديات، قمنا بتطوير منصة إلكترونية مبتكرة مخصصة لقطاع العقارات في فلسطين. تسهّل المنصة إدراج الإعلانات العقارية بما يشمل الأراضي والشقق والمساحات التجارية، حيث يمكن لأصحاب العقارات تقديم معلومات مفصلة مثل الموقع والسعر والصور. يمكن للمستخدمين استعراض العقارات المتاحة والتواصل مباشرة مع المالكين وتقديم الاستفسارات أو الطلبات.

بالإضافة إلى ذلك، تتيح المنصة الربط السلس مع مقدمي الخدمات مثل الكهربائيين والسباكين والمهندسين لتوفير حلول عقارية متكاملة. وتشمل المنصة أيضاً ميزات لتقديم الشكاوى واستعراضات المستخدمين، مع واجهة استخدام سهلة تعزز من الشفافية والثقة، مما يساهم في تبسيط المعاملات العقارية وتحسين تجربة المستخدم بشكل عام. كما تم دعم النظام بميزة الخرائط التفاعلية لعرض مواقع العقارات بدقة وتسهيل عملية الاستكشاف والبحث.

جدول المحتويات

1	الفصل الأول: المقدمة
2	1.1 المقدمة
2	1.2 المشكلة
2	1.3 فكرة عامة عن المشروع
3	1.4 فكرة المشروع
3	1.5 الدوافع والمحفزات:
	1.6 الأهداف: 3
3	1.7 نطاق المشروع:
4	1.8 أهمية المشروع
4	1.9 مخطط السياق (Context Diagram)
5	1.10 الحلول
5	1.10.1 موقع إلكتروني (Web Application)
5	1.10.2 تطبيق هاتف (Mobile Application)
6	1.11 جدولة المهام والجدول الزمني للمهام
7	الفصل الثاني: تحليل متطلبات النظام
8	2.1 المقدمة
8	2.2 المتطلبات الوظيفية
8	2.2.1 متطلبات مستخدم النظام
9	2.3 متطلبات النظام غير الوظيفية:
10	2.4 مخطط حالة الاستخدام (Use Case Diagram)
12	2.5 وصف المتطلبات الوظيفية للنظام
20	الفصل الثالث: تصميم النظام
21	1.1 المقدمة :
21	1.2 نموذج معمارية النظام:
21	3.2.1 Model
22	3.2.2 View
22	3.2.3 Controller
22	1.2.4 سبب اختيار نموذج MVC
23	3.2.5 البدائل المعمارية (Alternative Architectures)
25	3.2.6 ER Model
26	3.4 جداول قواعد البيانات :

32 الفصل الرابع : تنفيذ النظام

33 4.1 المقدمة

33 Back-End Tools & Libraries

33 Front-End Tools & Libraries

34 Deployment & Tools

35 Code 4.2

42 الفصل الخامس : فحص النظام

43 5.1 المقدمة

43 5.2 التحقق (Validation)

44 5.3 اختبار التحقق

45 التحقق من جهة الواجهة (Front-End Validation)

45 التحقق من جهة الخلفية (Back-End Validation)

45 5.4 اختبار نقاط الاتصال (Endpoint Testing)

48 5.5.1 الاختبار اليدوي (QA Manual Testing)

48 السيناريو الأول: إرسال بيانات فارغة

49 السيناريو الثاني: إدخال إيميل بصيغة خاطئة

49 السيناريو الثالث: استخدام إيميل أو رقم هاتف مسجل مسبقاً

50 السيناريو الرابع: إدخال كلمة مرور ضعيفة

50 السيناريو الخامس: إدخال كود تأكيد خاطئ

51 السيناريو السادس: نسيان كلمة المرور

52 السيناريو السابع: محاولة تسجيل الدخول قبل تأكيد الحساب

53 السيناريو الثامن: محاولة اضافة مهنة وترك بعض الحقول فارغة

54 5.5.2 اختبار مدى توافق الشاشات وخاصة الهواتف (QA Responsive)

56 5.6 الإضافات

57 الفصل السادس الخاتمة والعمل المستقبلي

57 الخاتمة :

57 العمل المستقبلي:

58 References

قائمة الجداول

6.....	جدول : 1 الجدول الزمني للمهام على الأسابيع (Gantt Chart)
12.....	جدول : 2 انشاء حساب
12.....	جدول : 3 تسجيل الدخول
13.....	جدول : 4 استرجاع كلمة المرور
13.....	جدول : 5 البحث عن عقار / البحث عن مهنة
13.....	جدول : 6 استعراض التفاصيل
14.....	جدول : 7 استكشاف الاعلانات
14.....	جدول : 8 التواصل مع المزود
14.....	جدول : 9 تقييم العقارات والمهنيين
15.....	جدول : 10 إرسال الشكاوي
15.....	جدول : 11 ادارة الحساب
16.....	جدول : 12 إدارة العقارات
16.....	جدول : 13 ادارة المهنة
17.....	جدول : 14 ادارة الاعلانات
17.....	جدول : 15 إدارة الشكاوي
18.....	جدول : 16 إضافة المهنة
18.....	جدول : 17 حذف الحساب
19.....	جدول : 18 إرسال تحذيرات
26.....	جدول : 19 قائمة جداول فواعد البيانات
27.....	جدول : 20 وصف المستخدمين
27.....	جدول : 21 الصلاحيات
27.....	جدول : 22 وصف الجدول الخاص بالصلاحيات والمستخدمين
28.....	جدول : 23 وصف الجدول الخاص بالرسائل
28.....	جدول : 24 وصف الجدول الخاص بالاعلانات
28.....	جدول : 25 وصف جدول الخاص في الحفظ
29.....	جدول : 26 وصف الجدول الخاص بالعقارات
29.....	جدول : 27 جدول الخدمة
30.....	جدول : 28 وصف الجدول الخاص بالمراجعات
30.....	جدول : 29 وصف الجدول الخاص بالشكاوي
30.....	جدول : 30 وصف الجدول الخاص بنوع المهنة
31.....	جدول : 31 وصف جدول العقار
31.....	جدول : 32 وصف جدول العنوان
31.....	جدول : 33 وصف جدول الصور

قائمة الأشكال

4	شكل 1 مخطط السياق Context Diagram
11	شكل 2 مخطط حالة الاستخدام (use case model)
21	شكل 3 نموذج معمارية النظام
25	شكل 4 ER MODEL

قائمة الواجهات

35	واجهة 1 إنشاء حساب
36	واجهة 3 رمز التحقق عبر الأيميل
36	واجهة 2 ادخال كود التحقق
37	واجهة 4 تسجيل الدخول
37	واجهة 5 الرئيسية
38	واجهة 6 تصفح الاعلانات
38	واجهة 7 تصفح المهن
38	واجهة 8 اضافة عقار او مهنة
39	واجهة 9 صفحة تصفح الاعلانات
39	واجهة 10 المحفوظات
40	واجهة 11 الأعدادات
41	واجهة 12 صفحة الأيمن

الفصل الأول: المقدمة

المقدمة
المشكلة
فكرة عامة عن المشروع
فكرة المشروع
الدوافع والمحفزات
الأهداف
أهمية المشروع
نطاق المشروع
مخطط السياق

1.1 المقدمة

سنقوم في هذا القسم من المشروع بالتحدث عن المشكلة الرئيسية التي نهدف إلى حلها، وكيف يسعى المشروع إلى تقديم حلول مبتكرة تسهل عملية البحث عن العقارات والمهنيين في فلسطين.

ونستعرض فكرة المشروع بشكل عام، ونبين نطاق عمل النظام والأنظمة والخدمات المرتبطة به. سيتم تحديد الفئات المستفيدة من هذا النظام وإظهار أهميته في تسهيل الوصول إلى العقارات والخدمات المساندة، كما سيتم استعراض الوظائف الأساسية التي يوفرها النظام والآلية التي يتعامل بها مع المستخدمين.

1.2 المشكلة

أدى التطور السريع في السوق الفلسطيني وزيادة الطلب على العقارات والخدمات المهنية إلى ظهور العديد من التحديات في إيجاد حلول فعالة وسريعة للوصول إلى هذه الخدمات. بالنسبة للأفراد الذين يبحثون عن عقار مناسب أو خدمات مهنية موثوقة، غالبًا ما يواجهون صعوبة في الوصول إلى المعلومات الدقيقة حول المهنيين الأقرب إليهم أو العقارات المتاحة في مواقعهم المرغوبة. يتسبب نقص التواصل الفعال بين العملاء والمهنيين في تأخير إتمام المشاريع أو الحصول على خدمات موثوقة في الوقت المناسب.

بالإضافة إلى ذلك، يواجه المستخدمون صعوبة في تقييم جودة المهنيين أو العقارات نظرًا لغياب نظام تقييم موثوق يمكنهم من اتخاذ قرارات مبنية على تجارب الآخرين. كل هذه التحديات تعيق عملية الوصول إلى الخدمات المطلوبة وتبطئ من سرعة اتخاذ القرارات المناسبة.

1.3 فكرة عامة عن المشروع

المشروع عبارة عن منصة إلكترونية موجهة للأفراد والشركات في فلسطين تهدف إلى نشر العقارات والخدمات المهنية. توفر المنصة وسيلة فعالة لنشر العقارات المتاحة للبيع أو الإيجار مع إمكانية إضافة معلومات تفصيلية مثل الموقع، السعر، والصور، مما يسهل على المشترين أو المستأجرين البحث واتخاذ القرارات المناسبة.

بالإضافة إلى ذلك، تقدم المنصة خدمة البحث عن المهنيين المحليين مثل السباكين والكهربائيين والمهندسين المعماريين الذين يمكنهم المساعدة في إتمام أو صيانة العقارات. يقوم النظام بتحديد المهنيين الأقرب جغرافيًا للمستخدم، ويوفر وسيلة للتواصل المباشر معهم من خلال الموقع أو تطبيقات مثل تطبيقات التواصل الاجتماعي لتسهيل التفاعل السريع.

يتيح النظام أيضًا للمستخدمين تقديم تقييمات للعقارات والمهنيين بناءً على تجاربهم السابقة، مما يعزز من الشفافية والثقة في المنصة. كما يتم دمج خرائط تفاعلية تعرض المواقع الجغرافية للعقارات المتاحة، مما يسهل عملية التنقل واتخاذ القرارات بناءً على القرب الجغرافي.

1.4 فكرة المشروع

مع زيادة الطلب على العقارات والخدمات المهنية في فلسطين، يواجه الكثير من الأفراد والشركات تحديات في العثور على العقارات المناسبة أو المهنيين الأقرب جغرافياً لتقديم الخدمات المطلوبة. غالباً ما تؤدي هذه التحديات إلى تأخير في استكمال المشاريع أو اتخاذ القرارات المناسبة، مما يؤثر سلباً على كفاءة وسرعة العمل. "إدارة العقارات والخدمات المهنية في فلسطين" هي مشروع مبتكر يأتي كحل مقترح لهذه المشكلة الشائعة. يهدف المشروع إلى تحسين تجربة المستخدمين وتسهيل عملية البحث عن العقارات والخدمات المهنية في فلسطين. يتم ذلك من خلال توفير منصة رقمية شاملة تمكن المستخدمين من نشر عقاراتهم أو البحث عنها بسهولة، مع توفير معلومات تفصيلية وموثوقة حول كل عقار.

1.5 الدوافع والمحفزات:

فيما يلي أهم الدوافع والمحفزات لإنشاء منصة "إدارة العقارات والخدمات المهنية في فلسطين":

1. زيادة الطلب على العقارات وخدمات والمهنيين في فلسطين.
2. الحاجة إلى تسهيل الوصول إلى المهنيين المحليين والعقارات.
3. تعزيز التواصل المباشر بين الأفراد والمهنيين الموثوقين.
4. تحسين الكفاءة في إدارة الوقت والموارد للوصول إلى العقارات والخدمات بسرعة.

1.6 الأهداف:

يأتي مشروع منصة "إدارة العقارات والخدمات المهنية في فلسطين" لتحقيق عدة أهداف مهمة:

1. توفير منصة إلكترونية لنشر العقارات والخدمات المهنية.
2. تسهيل عملية البحث عن المهنيين المحليين بناءً على الموقع الجغرافي.
3. تقليل الوقت والجهد المبذول في العثور على عقار مناسب أو مهني.
4. تعزيز الشفافية من خلال نظام تقييمات المستخدمين للعقارات والمهنيين.
5. تقديم تجربة مستخدم سهلة وسريعة للتواصل وإتمام المعاملات.

1.7 نطاق المشروع:

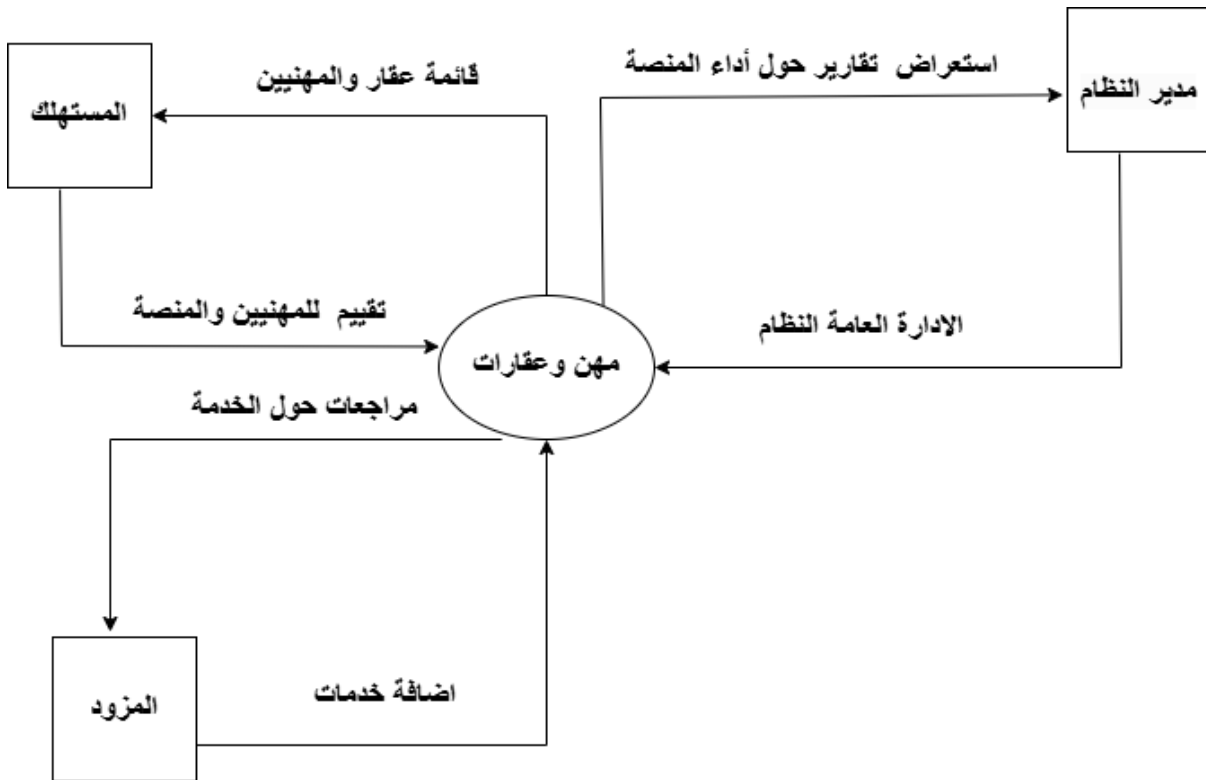
يستهدف المشروع الأشخاص والشركات في نطاق فلسطين، وهم:

1. الأفراد الراغبون في بيع أو شراء العقارات.
2. الشركات العقارية والمكاتب الهندسية.
3. المهنيون (السياكون، الكهربائيون، وغيرهم) الذين يقدمون خدمات مهنية.

1.8 أهمية المشروع

يمثل مشروع "منصة إدارة العقارات والخدمات المهنية في فلسطين" حلاً تقنيًا يساهم في تسهيل وصول الأفراد إلى العقارات والخدمات المهنية. من خلال توفير نظام متكامل لنشر العقارات وإيجاد المهنيين القريبين جغرافيًا، يعزز المشروع من سرعة اتخاذ القرارات ويساعد على تحسين الكفاءة والراحة للمستخدمين. كما يساهم المشروع في زيادة الوعي الرقمي ودمج التكنولوجيا في تلبية احتياجات المجتمع بطريقة أكثر مرونة وفعالية.

1.9 مخطط السياق (Context Diagram)



شكل 1 مخطط السياق Context Diagram

يوضح الشكل (1) مخطط السياق الخاص بمشروع "منصة إدارة العقارات والخدمات المهنية في فلسطين"، حيث يتم فيه عرض العناصر الرئيسية المتفاعلة مع النظام والمهام الأساسية التي تقوم بها.

1.10 الحلول

1.10.1 موقع إلكتروني (Web Application)

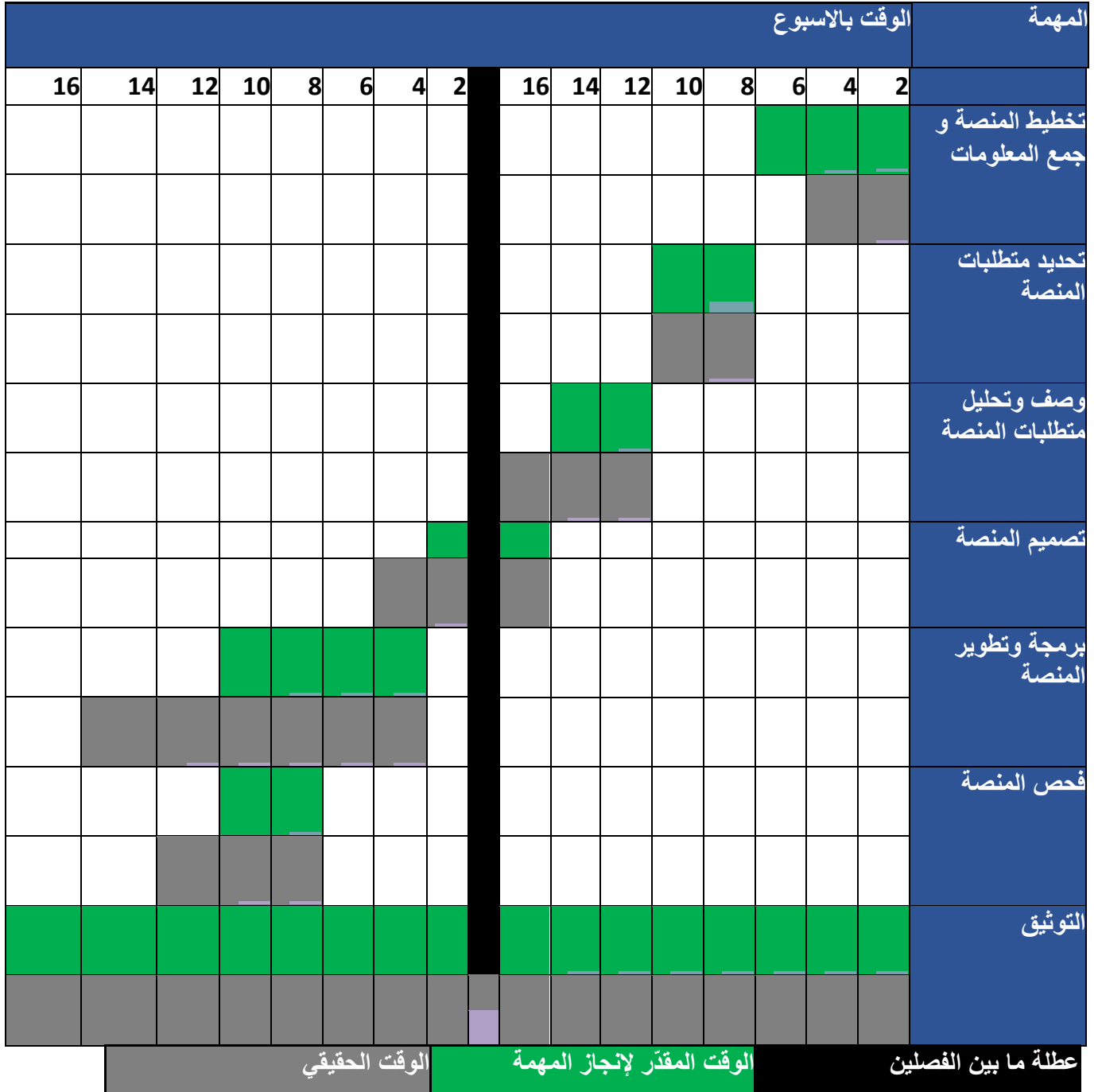
- **فكرة النظام:**
تصميم موقع إلكتروني يتضمن جميع الميزات التي تحدثنا عنها مثل إدارة العقارات، عرض الخدمات، والتواصل بين المستخدمين والمعلنين.
الموقع يعمل من خلال المتصفح على أي جهاز متصل بالإنترنت.
- **مميزات النظام:**
 - يعمل على أكثر من منصة (أجهزة كمبيوتر، هواتف ذكية، وأجهزة لوحية).
 - سهولة التحديث على النظام في أي وقت دون الحاجة إلى تثبيت تحديثات على الأجهزة.
 - قابلية الوصول من أي مكان وأي جهاز متصل بالإنترنت.
- **عيوب النظام:**
 - يتطلب اتصالاً مستمراً بالإنترنت للعمل.
 - لا يمكن تحميله كتطبيق من متاجر التطبيقات، مما قد يؤثر على قابلية الاكتشاف.
 - يكون أبطأ نسبياً في الاستجابة مقارنةً بتطبيقات الهاتف أو سطح المكتب.
- **لماذا هو الأفضل؟**
الموقع الإلكتروني يُعتبر الحل الأنسب لمشروعنا نظراً لقدرته على الوصول لشريحة واسعة من المستخدمين بسهولة عبر أي منصة ومناسب لكل المنصات ، مع إمكانية التطوير المستمر وإضافة ميزات مستقبلية دون تعقيد.

1.10.2 تطبيق هاتف (Mobile Application)

- **فكرة النظام:**
تصميم تطبيق يعمل على أنظمة الهواتف الذكية Android و iOS يقدم نفس الميزات التي يوفرها الموقع.
- **مميزات النظام:**
 - سهولة الوصول والاستخدام من خلال الهاتف المحمول، وهو الجهاز الأكثر استخداماً حالياً.
 - يمكن تحميله من متاجر التطبيقات مما يسهل اكتشافه من قبل المستخدمين.
 - يدعم الإشعارات الفورية لإبقاء المستخدمين على اطلاع دائم بالمستجدات.
 - إمكانية العمل في وضع عدم الاتصال لبعض الميزات، مع مزامنة البيانات لاحقاً عند الاتصال بالإنترنت.
- **عيوب النظام:**
 - قد يتطلب موارد إضافية لتطوير نسختين Android و iOS .
 - قد يواجه المستخدمون مشاكل إذا لم يكن لديهم مساحة تخزين كافية لتحميل التطبيق.

1.11 جدولة المهام والجدول الزمني للمهام

يقدم الجدول (1) رؤية لتوزيع مراحل المشروع على مدار الفصلين الدراسيين، مع توضيح المواعيد النهائية والتقديرية لإنجاز المهام. يتيح هذا التوزيع الزمني تصورًا لتقدم المشروع والمهام المتوقع إنجازها خلال كل فصل دراسي.



جدول : 1 الجدول الزمني للمهام على الأسابيع (Gantt Chart)

الفصل الثاني: تحليل متطلبات النظام

المقدمة
المتطلبات الوظيفية
متطلبات مستخدمي النظام
متطلبات المزودين
متطلبات المستهلكين
متطلبات مدير النظام
متطلبات النظام الغير وظيفية
مخطط حالة الاستخدام

2.1 المقدمة

يتناول هذا القسم تحليل المتطلبات الوظيفية للنظام بناءً على ممثلي النظام الأساسيين، وهم:

1. **مدير النظام:** المسؤول عن إدارة المحتوى ومتابعة التفاعلات على المنصة.
2. **المستهلك:** الذي يبحث عن العقارات والخدمات، ويقوم بتقديم الطلبات وتقييم الخدمات.
3. **المزود:** الذي يعرض العقارات والخدمات المهنية ويتفاعل مع المستهلكين.
4. **الزائر:** الذي يستعرض العقارات والخدمات المهنية بدون ان يرى التفاصيل وبدون التفاعلات وبدون تسجيل دخول الى الموقع.

كما يستعرض هذا القسم متطلبات النظام ونموذج الاستخدام (Use Case Diagram) لتوضيح كيفية تفاعل كل ممثل مع النظام.

2.2 المتطلبات الوظيفية

تشمل هذه المتطلبات احتياجات مستخدمي النظام الرئيسيين وهم: مدير النظام، المستهلكين، و المزودين، بالإضافة إلى المتطلبات التي يقوم بها النظام تلقائياً.

2.2.1 متطلبات مستخدمى النظام

2.2.1.1 متطلبات المزود:

1. إنشاء حساب باستخدام بياناته الشخصية والمهنية.
2. تسجيل الدخول باستخدام البريد الإلكتروني أو رقم الهاتف وكلمة المرور.
3. إضافة إعلان عن عقار أو مهنة يتضمن التفاصيل المهمة (مثل الوصف، الموقع، السعر، الصور، ووسائل التواصل).
4. تحديث الإعلانات المنشورة لتعديل المعلومات أو الصور حسب الحاجة.
5. حذف الإعلانات التي لم تعد متاحة أو لم تعد صالحة للعرض.
6. اضافة وسائل التواصل (مثل البريد الإلكتروني، رقم الهاتف، أو التواصل المباشر من خلال المحادثة داخل الموقع)

2.2.1.2 متطلبات المستهلك:

1. إنشاء حساب باستخدام البيانات الشخصية.
2. تسجيل الدخول باستخدام البريد الإلكتروني أو رقم الهاتف وكلمة المرور.
3. البحث عن الإعلانات المتاحة حسب الموقع أو نوع العقار أو الخدمة.
4. استعراض تفاصيل الإعلان (الوصف، الصور، الموقع، السعر).
5. التواصل مع المزودين من خلال وسائل التواصل الاجتماعي المدرجة في الإعلان ل طرح استفساراتهم أو طلب مزيد من التفاصيل.
6. إضافة إعلانات إلى قائمة المفضلة للرجوع إليها لاحقاً.
7. رؤية العقار عبر الخريطة التفاعلية.
8. تقييم الإعلانات بناءً على محتواها، و تقييم المزودين مباشرة على المنصة.
9. تعديل بيانات الحساب الشخصية.

2.2.1.3 متطلبات المستخدم الزائر:

1. تصفح الإعلانات دون الحاجة إلى تسجيل الدخول.
2. البحث عن العقارات والخدمات باستخدام الفلاتر المتاحة مثل الموقع، النوع، والسعر.
3. إنشاء حساب: عند الرغبة في الاستفادة من كامل خصائص النظام.

2.2.1.4 متطلبات مدير النظام:

1. مراجعة الإعلانات:
 - مراقبة الإعلانات المنشورة على المنصة للتحقق من توافقها مع سياسات الاستخدام والمعايير المعتمدة.
 - تعديل أو حذف الإعلانات التي تخالف السياسات.
2. مراجعة الشكاوى:
 - معالجة الشكاوى المقدمة من المستخدمين حول المحتوى غير المناسب أو الخدمات.
 - اتخاذ الإجراءات المناسبة بناءً على محتوى الشكاوى.
3. إرسال التحذيرات:
 - تنبيه المستخدمين المخالفين عبر إرسال تحذيرات رسمية لتجنب تكرار المخالفات.
4. تعطيل الحسابات عند تكرار المخالفات:
 - تعطيل حسابات المزودين أو المستهلكين المخالفين بعد تجاوز عدد معين من التحذيرات.
5. إدارة المهن :
 - اضافة وتعديل المهن المنسدة بقائمة المهن التي اقدمها.
6. الإشراف العام على جميع خدمات التطبيق:
 - مراقبة سير العمليات داخل النظام لضمان تقديم خدمات موثوقة وسلسة.
 - تحسين تجربة المستخدمين من خلال متابعة الأداء وإدارة الأنشطة اليومية.

2.3 متطلبات النظام غير الوظيفية:

1. واجهة مستخدم سهلة وسلسة:
 - يجب أن تتمتع واجهة المستخدم بتصميم بسيط ومنظم، بحيث لا يزيد الوقت اللازم لفهم أي ميزة جديدة في النظام عن 1 دقيقة.
 - يجب أن يتمكن المستخدمون من إكمال العمليات الأساسية (مثل البحث عن إعلان أو تسجيل الدخول) في غضون 3 خطوات أو أقل
 - يجب أن يتم اختباره على الشاشات الصغيرة (مثل الهواتف) والكبيرة (مثل الحواسيب) لضمان تناسق التصميم.

2. أداء عالي:
- يجب أن يكون زمن استجابة النظام لجميع العمليات أقل من 5 ثواني .
3. أمان البيانات والخصوصية:
- يجب حماية البيانات الشخصية للمهنيين والمستفيدين.
4. توافر النظام :
- يجب أن يكون النظام متاحًا على مدار الساعة طيلة الاسبوع.

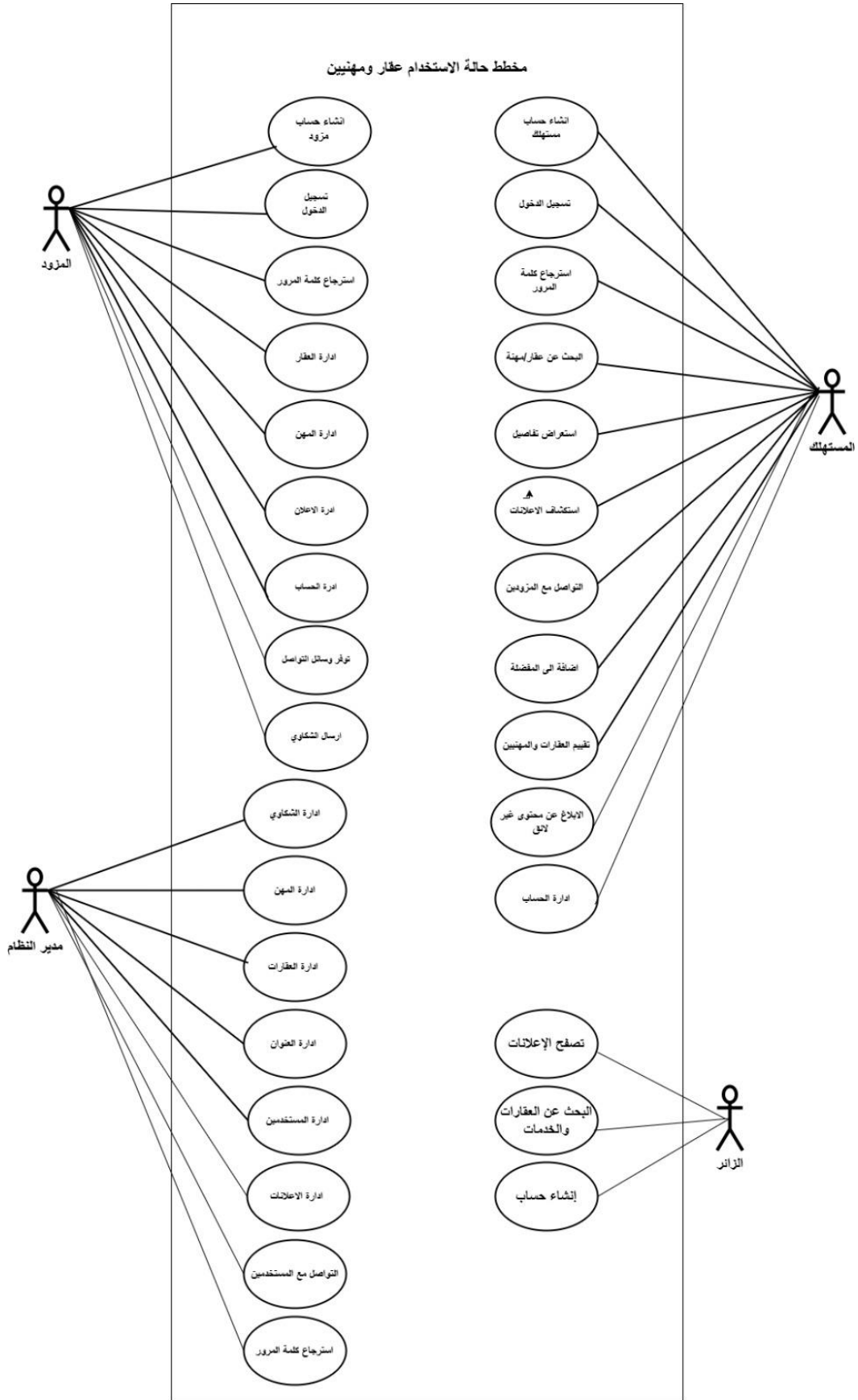
2.4 مخطط حالة الاستخدام (Use Case Diagram)

في نظام "منصة إدارة العقارات والخدمات المهنية في فلسطين"، يتمتع كل مستخدم بصلاحيات وخدمات مخصصة وفقاً لدوره. فعند دخول مدير النظام إلى المنصة باستخدام بياناته كمدير، يتمكن من الإشراف على جميع محتويات التطبيق وإجراء التعديلات اللازمة على الإعلانات والعقارات والخدمات المتاحة. كما يستطيع مراجعة وتأكيده حسابات المستخدمين، سواء كانوا مزودين أو مستهلكين، والتأكد من توافقهم مع سياسات النظام. بالإضافة إلى ذلك، يكون مسؤولاً عن تنظيم وإدارة الإعلانات داخل النظام ومتابعة محتواها، وكذلك الإشراف على التفاعلات المختلفة بين المستخدمين، وإرسال التحذيرات أو تعطيل الحسابات عند الضرورة، مما يضمن تقديم تجربة آمنة وموثوقة للجميع.

أما المستهلك، فيمكنه بعد تسجيل الدخول استعراض العقارات والخدمات المتاحة على المنصة، والبحث عنها حسب الموقع أو الفئة أو السعر. كما يمكنه عرض تفاصيل الإعلانات، والتواصل مع المزودين عبر وسائل التواصل الاجتماعي المدرجة، مما يتيح له طرح الأسئلة أو طلب المزيد من المعلومات. يتيح النظام أيضاً للمستهلكين حفظ الإعلانات المفضلة للرجوع إليها لاحقاً، بالإضافة إلى إمكانية تقييم الإعلانات وتحديث بياناتهم الشخصية عند الحاجة.

بالنسبة للمزود، فيتيح له النظام بعد تسجيل الدخول إضافة إعلانات عقارات أو خدمات جديدة وتحديث أو حذف إعلاناته عند الضرورة. ويستطيع المزود كذلك توفير وسائل للتواصل المباشر مع المستهلكين عبر تطبيقات التواصل الاجتماعي، ما يسهل عملية التواصل ويزيد من فرص وصوله إلى عملاء جدد.

أما المستخدم الزائر، فيمكنه تصفح الإعلانات المتاحة على المنصة دون الحاجة إلى تسجيل الدخول، حيث يمكنه رؤية العقارات والمهن المعروضة بشكل عام دون القدرة على عرض التفاصيل الكاملة أو التعليقات. كما يمكنه استخدام الفلاتر الأساسية مثل الموقع ونوع العقار أو الخدمة لتسهيل عملية التصفح. ومع ذلك، فإن الوصول إلى معلومات الإعلان التفصيلية، مثل الوصف الكامل، الصور، الأسعار، أو التفاعل مع الإعلانات (مثل الحفظ في المفضلة أو التواصل مع المزودين)، يتطلب من الزائر إنشاء حساب وتسجيل الدخول للاستفادة من هذه الميزات.



شكل 2 مخطط حالة الاستخدام (use case model)

2.5 وصف المتطلبات الوظيفية للنظام

في هذا القسم، يتم توضيح التفاصيل الخاصة بكل متطلب وظيفي مذكور في مخطط الاستخدام، لتزويد صورة واضحة عن كيفية بناء كل متطلب من المتطلبات المذكورة :

اسم الوظيفة	إنشاء حساب
الممثل الأساسي	المستهلك / المزود
الهدف	تمكين المستخدمين من إنشاء حساب في النظام
الشروط المسبقة	لا يمتلك المستخدمين حساباً سابقاً ويجب أن يقدم جميع البيانات المطلوبة
السيناريو	<ol style="list-style-type: none"> 1. يختار المستخدم "إنشاء حساب". 2. يحدد ما إذا كان يفضل التحقق عبر واتساب أو البريد الإلكتروني. 3. يُدخل بياناته (الاسم، البريد، كلمة المرور، الدور، المدينة). 4. ينشئ النظام الحساب ويحوّله إلى صفحة تأكيد البريد أو الهاتف.
الاستثناءات	<ol style="list-style-type: none"> 1. يقوم المستخدمون بإدخال بيانات غير صالحة. 2. يترك حقولاً مطلوبة فارغة. 3. الحساب موجود مسبقاً. 4. مشكلة بالاتصال.

جدول : 2 إنشاء حساب

اسم الوظيفة	تسجيل الدخول
الممثل الأساسي	المستهلك / المزود / مدير النظام
الهدف	تمكين المستخدمين من الوصول إلى حسابهم في النظام
الشروط المسبقة	يمتلك المستخدم حساباً نشطاً وبيانات تسجيل الدخول الصحيحة (اسم مستخدم أو بريد إلكتروني أو رقم الهاتف و كلمة المرور).
السيناريو	<ol style="list-style-type: none"> 1. يختار المستخدم "تسجيل الدخول". 2. يطلب النظام إدخال اسم المستخدم أو البريد أو رقم الهاتف مع كلمة المرور. 3. يُدخل المستخدم البيانات المطلوبة. 4. يتحقق النظام من صحتها ويمنح الوصول عند التحقق.
الاستثناءات	يقوم المستخدمون بإدخال اسم مستخدم أو كلمة مرور غير صحيحة

جدول : 3 تسجيل الدخول

اسم الوظيفة	استرجاع كلمة المرور
الممثل	المستهلك / المزود / مدير النظام
الهدف	تمكين المستخدمين من استعادة الحساب في حال نسي كلمة المرور
الشروط المسبقة	يجب أن يكون لدى المستخدمين بريد إلكتروني أو رقم هاتف أو اسم المستخدم مسجل بالنظام
السيناريو	<ol style="list-style-type: none"> 1. يضغط " استرجاع كلمة المرور." 2. يُدخل بريده الإلكتروني أو اسم المستخدم أو رقم الهاتف . 3. يتلقى المستخدمين كود تحقق من البريد الإلكتروني أو الواتساب . 4. يدخل الكود , ويعيد تعيين كلمة مرور جديدة .
الاستثناءات	بريد أو رقم هاتف أو اسم غير موجود في النظام

جدول : 4 استرجاع كلمة المرور

اسم الوظيفة	البحث عن عقار / البحث عن مهنة
الممثل	المستهلك / المزود / مدير النظام / الزائر
الهدف	تمكين المستخدمين من البحث عن معلومات حول عقار أو مهنة من خلال النظام.
الشروط المسبقة	يجب أن يكون لدى المستخدمين إمكانية الوصول إلى النظام ومعرفة نوع العقار أو المهنة التي يريد البحث عنها.
السيناريو	<ol style="list-style-type: none"> 1. يختار المستخدمين خيار البحث عن عقار/ مهنة 2. يبحث المستخدمين باستخدام الكلمات المفتاحية أو نظام الفلاتر . 3. يعرض النظام قائمة بالعقارات أو المهن التي تطابق معايير البحث. 4. يختار المستخدمين العقار أو المهنة المطلوبة للحصول على تفاصيل إضافية.
الاستثناءات	عرض رسالة تفيد بعدم توفر نتائج مطابقة مع إمكانية تعديل معايير البحث أو تصفح الإعلانات المتاحة.

جدول : 5 البحث عن عقار / البحث عن مهنة

اسم الوظيفة	استعراض التفاصيل
الممثل	المستهلك / المزود / مدير النظام
الهدف	تمكين المستخدمين من رؤية تفاصيل الإعلان أو العقار أو المهنة بشكل كامل بعد اختياره
الشروط المسبقة	يجب أن يكون المستخدمين قد وصل إلى قائمة الإعلانات أو العقارات أو المهن إما عبر البحث أو من خلال الصفحة الرئيسية
السيناريو	<ol style="list-style-type: none"> 1. يضغط المستخدمين على الإعلان أو العقار أو المهنة في نتائج البحث أو الصفحة الرئيسية 2. يعرض النظام تفاصيل الإعلان أو العقار أو المهنة المحدد 3. يستعرض المستخدمين معلومات الإعلان أو عقار أو مهنة
الاستثناءات	1. يواجه المستخدم خطأ في الشبكة عند تحميل التفاصيل

جدول : 6 استعراض التفاصيل

اسم الوظيفة	استكشاف الاعلانات
الممثل	المستهلك / المزود / مدير النظام / الزائر
الهدف	تصفح الإعلانات المعروضة في المنصة والتفاعل معها
الشروط المسبقة	يجب أن يكون المستخدم قد وصل إلى قائمة الإعلانات إما عبر البحث أو من خلال الصفحة الرئيسية
السيناريو	1. يدخل المستخدم الصفحة الرئيسية أو قسم الإعلانات. 2. يعرض النظام قائمة الإعلانات. 3. ينتقل المستخدم بين الاعلانات المعروضة
الاستثناءات	عدم وجود إعلانات حاليًا.

جدول : 7 استكشاف الاعلانات

اسم الوظيفة	التواصل مع المزود
الممثل	المستهلك / المزود / مدير النظام
الهدف	تمكين المستخدم من التواصل مع المستخدم الآخر عبر المحادثة الفورية التي يسمح بها الموقع.
الشروط المسبقة	يجب أن يكون المستخدم قد سجل الدخول و قد استعرض الإعلان
السيناريو	1. يختار المستخدم المحادثة الفورية المدرجة في تفاصيل الإعلان 2. يقوم النظام بفتح قائمة للمحادثة بين الطرفين. 3. يطرح المستخدم استفساراته .
الاستثناءات	في حال انتهاء جلسة تسجيل الدخول.

جدول : 8 التواصل مع المزود

اسم الوظيفة	تقييم العقارات والمهنيين
الممثل	المستهلك / المزود
الهدف	تمكين المستخدم من تقييم العقارات أو المهنيين لتحسين تجربة المستخدم وتعزيز الشفافية
الشروط المسبقة	يمتلك المستخدم حسابًا نشطًا ومسجل الدخول إلى النظام، وأن يكون قد دخل الى صفحة التفاصيل.
السيناريو	1. يفتح المستخدم صفحة التفاصيل لإضافة التقييم . 2. يختار المستخدم خيار "إضافة تقييم" 3. يقوم المستخدم بتحديد التقييم (عدد النجوم) وإضافة تعليق. 4. يقوم النظام بحفظ التقييم وربطه بالعقار أو المهني 5. يعرض النظام إشعارًا بتأكيد نجاح عملية التقييم
الاستثناءات	في حال انتهاء جلسة تسجيل الدخول.

جدول : 9 تقييم العقارات والمهنيين

اسم الوظيفة	إرسال الشكاوي
الممثل	المستهلك / المزود
الهدف	تمكين المستخدم من الإبلاغ عن محتوى غير لائق أو مخالف لسياسات الموقع
الشروط المسبقة	يملك المستهلك حسابًا نشطًا ومسجل الدخول إلى النظام، وأن يكون المحتوى موجودًا على الموقع
السيناريو	<ol style="list-style-type: none"> 1. يفتح المستخدم صفحة التفاصيل لتقديم شكوى 2. يختار المستخدم خيار " تقديم شكوى " 3. يظهر النظام نموذجًا يطلب من المستخدم تحديد سبب الشكوى (مثل محتوى مسيء، غير لائق، أو مخالف) وإضافة صورة إذا اراد 4. يقوم المستهلك بتعبئة النموذج وإرسال البلاغ 5. يعرض النظام إشعارًا بتأكيد استلام البلاغ
الاستثناءات	في حال انتهاء جلسة تسجيل الدخول.

جدول : 10 إرسال الشكاوي

اسم الوظيفة	ادارة الحساب
الممثل	المستهلك / المزود / مدير النظام
الهدف	تمكين المستخدم من تعديل بيانات حسابه الشخصية بسهولة ومرونة
الشروط المسبقة	يملك المستخدم حسابًا نشطًا ومسجل الدخول إلى النظام
السيناريو	<ol style="list-style-type: none"> 1. يدخل المستخدم إلى صفحة "إعدادات الحساب." 2. يختار المستخدم قسم البيانات المراد تعديلها (مثل الاسم أو البريد الإلكتروني). 3. يقوم المستخدم بتحديث البيانات المطلوبة. 4. يضغط المستخدم على خيار "حفظ التغييرات" 5. يقوم النظام بحفظ التعديلات وتحديث البيانات في الحساب.
الاستثناءات	<ol style="list-style-type: none"> 1. إدخال بيانات غير صالحة (مثل بريد إلكتروني غير صحيح). 2. في حال انتهاء جلسة تسجيل الدخول.

جدول : 11 ادارة الحساب

اسم الوظيفة	إدارة العقارات
الممثل	المزود/ مدير النظام
الهدف	تمكين المستخدم من إضافة أو تعديل أو حذف عقار.
الشروط المسبقة	يجب أن يكون لدى المزود صلاحية الوصول إلى النظام وان يكون هو من اضاف العقار او مدير النظام في حال اراد التعديل عليه .
السيناريو	<ol style="list-style-type: none"> 1. يختار المزود قسم العقارات الخاصة به . 2. يختار اضافة عقار جديد او يختار عقار قديم للتعديل عليه او حذفه 3. يتحقق النظام من صحة البيانات المدخلة. 4. يحفظ النظام البيانات ويؤكد نجاح العملية.
الاستثناءات	<ol style="list-style-type: none"> 1. بيانات ناقصة أو صلاحيات غير كافية أو مشاكل في الحفظ 2. في حال انتهاء جلسة تسجيل الدخول.

جدول : 12 إدارة العقارات

اسم الوظيفة	ادارة المهن
الممثل	المزود/ مدير النظام
الهدف	تمكين المستخدم من إضافة أو تعديل أو حذف عقار.
الشروط المسبقة	يجب أن يكون لدى المزود صلاحية الوصول إلى النظام وان يكون هو من اضاف العقار او مدير النظام في حال اراد التعديل عليه .
المحفز	يرغب المزود في عرض مهاراته المهنية ضمن النظام او التعديل على مهنة سابقة او حذف مهنة معينة
السيناريو	<ol style="list-style-type: none"> 1. يختار المزود قسم المهن الخاصة به . 2. يختار اضافة مهنة جديدة او يختار مهنة قديمة للتعديل عليها او حذفها 3. يتحقق النظام من صحة البيانات المدخلة. 4. يحفظ النظام البيانات ويؤكد نجاح العملية.
الاستثناءات	<ol style="list-style-type: none"> 1. بيانات ناقصة أو صلاحيات غير كافية أو مشاكل في الحفظ 2. في حال انتهاء جلسة تسجيل الدخول.

جدول : 13 ادارة المهن

اسم الوظيفة	ادارة الاعلانات
الممثل	المزود/ مدير النظام
الهدف	تمكين المستخدم من التحكم الكامل في الإعلانات المرتبطة بالعقارات أو الخدمات الخاصة به .
الشروط المسبقة	يجب أن يكون لدى المزود صلاحية الوصول إلى النظام، ويجب أن يكون مدير النظام أو المزود هو من قام بإضافة عقار أو مهنة في حال اراد التعديل على اعلان .
السيناريو	<ol style="list-style-type: none"> 1. يفتح المزود قائمة عقاراته او خددماته . 2. يضغط المزود على العقار او الخدمة التي يريد الاعلان عنها أو التعديل على اعلانها أو حذف اعلانها . 3. يؤكد المزود العملية . 4. يقوم النظام بحفظ التغييرات التي قام بها المزود .
الاستثناءات	<ol style="list-style-type: none"> 1. إذا كانت هناك مشكلة في النظام أو البيانات غير مكتملة، يعرض النظام إشعارًا بفشل العملية. 2. في حال انتهاء جلسة تسجيل الدخول.

جدول : 14 ادارة الاعلانات

اسم الوظيفة	إدارة الشكاوي
الممثل	مدير النظام
الهدف	تمكين مدير النظام من مراجعة الشكاوي المقدمة من المزودين واتخاذ الإجراءات المناسبة
الشروط المسبقة	يجب أن تكون هناك شكاوي مقدمة من المستخدمين في النظام
السيناريو	<ol style="list-style-type: none"> 1. يقوم مدير النظام بتسجيل الدخول إلى المنصة. 2. يدخل مدير النظام إلى صفحة "إدارة الشكاوي". 1. يختار الشكوى التي يريد مراجعتها من قائمة الشكاوي المقدمة. 2. يقرأ تفاصيل الشكوى المرفقة، مثل المشكلة المبلغ عنها . 3. يقرر مدير النظام ما إذا كان سيحتاج إلى اتخاذ إجراء مثل الرد على الشكوى أو حذفها.
الاستثناءات	<ol style="list-style-type: none"> 1. عدم وجود شكاوي جديدة للمراجعة. في حال انتهاء جلسة تسجيل الدخول.

جدول : 15 إدارة الشكاوي

اسم الوظيفة	إضافة المهنة
الممثل	مدير النظام
الهدف	تمكين مدير النظام من إضافة مهنة جديدة إلى النظام لتكون متاحة للمزودين.
الشروط المسبقة	يجب أن يكون لدى مدير النظام صلاحية الوصول ومسجل الدخول إلى النظام.
السيناريو	<ol style="list-style-type: none"> 1. يدخل مدير النظام إلى واجهة النظام ويسجل الدخول إلى حسابه. 2. ينتقل إلى قسم "إضافة مهنة جديدة". 3. يقوم بإدخال بيانات المهنة المطلوبة (اسم المهنة). 4. يضغط على زر "إضافة" لتأكيد إدخال المهنة الجديدة. 5. يقوم النظام بحفظ بيانات المهنة وإضافتها إلى قائمة المهن المتاحة. 6. يعرض النظام إشعارًا بتأكيد نجاح عملية الإضافة.
الاستثناءات	1. في حال انتهاء جلسة تسجيل الدخول.

جدول : 16 إضافة المهنة

اسم الوظيفة	حذف الحساب
الممثل	مدير النظام
الهدف	تمكين مدير النظام من حذف الحسابات المخالفة لسياسات المنصة أو غير النشطة لفترة طويلة من الزمن
الشروط المسبقة	يجب أن يكون هناك حسابات مخالفة أو غير نشطة في النظام
السيناريو	<ol style="list-style-type: none"> 1. يقوم مدير النظام بتسجيل الدخول إلى المنصة. 2. يدخل إلى صفحة "المستخدمين". 3. يضغط على خيار "حذف الحساب". 4. يقوم النظام بحذف الحساب ويعرض إشعارًا بتأكيد العملية. 5. يمكن لمدير النظام إرسال إشعار للمستخدم المعني (إذا كان الحساب غير نشط لفترة طويلة) ليخبره بإجراء الحذف.
الاستثناءات	في حال انتهاء جلسة تسجيل الدخول.

جدول : 17 حذف الحساب

اسم الوظيفة	إرسال تحذيرات
الممثل	مدير النظام
الهدف	تمكين مدير النظام من إرسال تحذيرات للمزودين أو المستهلكين عند ارتكاب مخالفة ما.
الشروط المسبقة	يجب ان يكون مدير النظام مسجل الدخول
المحفز	قيام المزود أو المستهلك بمخالفة قواعد النظام أو الشروط المحددة.
السيناريو	<ol style="list-style-type: none"> 1. يدخل مدير النظام إلى واجهة النظام ويسجل الدخول إلى حسابه. 2. ينتقل إلى قسم "مراقبة المخالفات". 3. يحدد المزود أو المستهلك الذي قام بالمخالفة. 4. يختار نوع المخالفة من قائمة المخالفات المحددة. 5. يقوم بإدخال تفاصيل التحذير (وصف المخالفة، الإجراء المطلوب). 6. يضغط على زر "إرسال تحذير". 7. يقوم النظام بإرسال إشعار تحذير إلى حساب المزود أو المستهلك. 8. يعرض النظام تأكيدًا بنجاح إرسال التحذير.
الاستثناءات	<ol style="list-style-type: none"> 1. إذا لم يتم العثور على حساب المزود أو المستهلك، يعرض النظام رسالة "الحساب غير موجود". 2. إذا كانت البيانات المدخلة غير مكتملة، يعرض النظام رسالة خطأ تطلب إكمال البيانات. 3. إذا حدثت مشكلة تقنية أثناء العملية، يعرض النظام إشعارًا بفشل إرسال التحذير

جدول : 18 إرسال تحذيرات

الفصل الثالث: تصميم النظام

المقدمة

نموذج معمارية النظام

Class Diagram

جداول قواعد البيانات

الواجهات الرسومية

1.1 المقدمة :

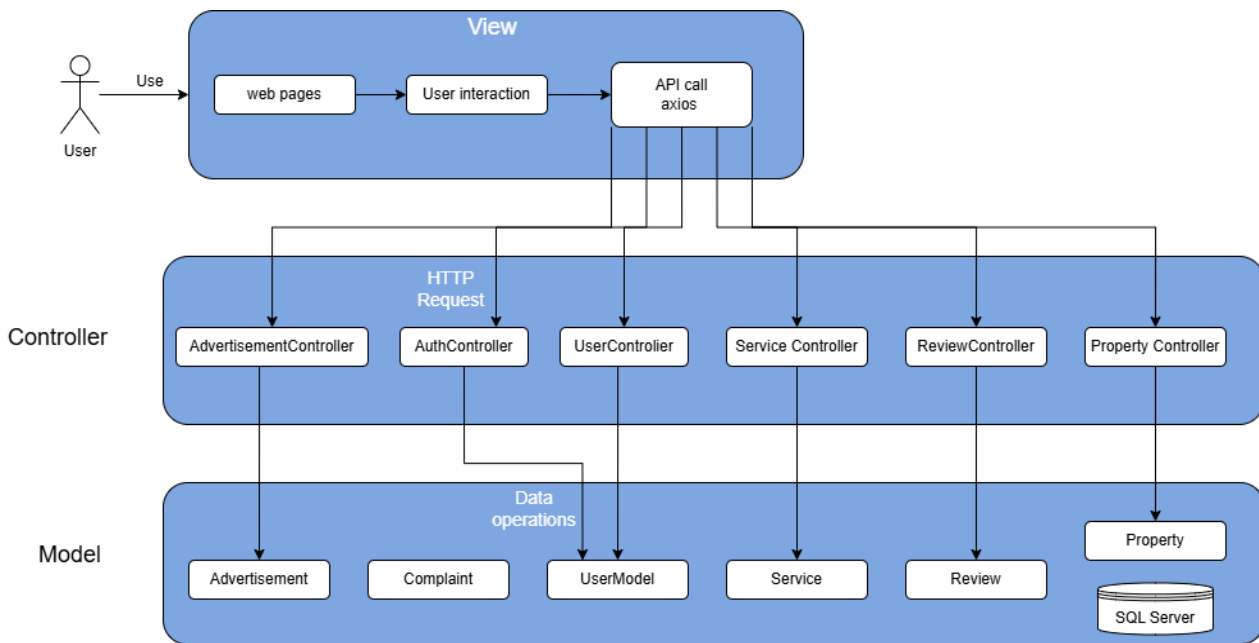
يتضمن هذا الفصل شرح تصميم وهيكلية المشروع، حيث سيتم تفصيل مكونات وأجزاء النظام، حتى نعطي فكرة كاملة عن جميع أجزاء النظام. من ناحية التصميم والمكونات الداخلية لها.

1.2 نموذج معمارية النظام:

سيتم اعتماد نموذج **Model-View-Controller (MVC)** لبناء النظام، حيث يتم فصل النظام إلى ثلاثة أجزاء أساسية:

Model, View, Controller

هذا النموذج يتيح فصل المهام بشكل واضح لكل مكون، مما يسهل التطوير والصيانة، ويعزز من قابلية التوسع في النظام. اختيار هذا النموذج مبني على دراسة دقيقة لاحتياجات المشروع ومتطلبات الأداء الفعال.



شكل 3 نموذج معمارية النظام

Model 3.2.1

الطبقة الخاصة بالمودل تمثل الجزء المسؤول عن تعريف الهيكل البنوي للبيانات في النظام، حيث تحتوي على الكيانات (Entities) التي تُجسد الجداول في قاعدة البيانات، مثل المستخدمين، الإعلانات، الخدمات، العقارات، المهن، التقييمات، والشكاوى. يتم ربط هذه الكيانات فيما بينها باستخدام العلاقات المناسبة مثل (One-to-One) و (One-to-Many)، مع ضمان تكامل البيانات من خلال استخدام قيود المفاتيح الأساسية والأجنبية (Primary/Foreign Keys). يوضح الشكل (4) في التقرير مخطط العلاقات بين هذه الكيانات (ER Model)، حيث يبين كيفية ترابطها معاً لدعم العمليات المتنوعة في النظام بشكل منظم ومتسق.

View 3.2.2

تمثل هذه الطبقة واجهة المستخدم التي يتم من خلالها تفاعل المستخدمين مع النظام. تتضمن صفحات تفاعلية مثل الصفحة الرئيسية، تسجيل الدخول، التسجيل، عرض الإعلانات، تفاصيل الإعلان، الخدمات، العقارات، إدارة الحساب، التقييمات، الشكاوى، المهن، ولوحة التحكم. تم تصميم هذه الواجهة لتكون مرنة وسريعة الاستجابة، مما يضمن تجربة استخدام سلسة عبر مختلف الأجهزة، ويسهل على المستخدمين التصفح، إدارة حساباتهم، تقديم التقييمات، ورفع الشكاوى بكفاءة. يوضح **القسم 4.3** تفاصيل هذا الجزء.

Controller 3.2.3

تمثل طبقة الـ **Controllers** الجزء المسؤول عن تنفيذ منطق العمليات في النظام، حيث تعمل كوسيط بين واجهة المستخدم وطبقة البيانات. عند حدوث أي تفاعل من المستخدم، مثل إرسال أو تعديل أو حذف بيانات، يقوم الـ **Controller** باستقبال الطلب وتنفيذ العمليات المطلوبة مثل الإدخال أو التعديل أو الحذف، ومن ثم إرجاع استجابة مناسبة إلى الواجهة. تحتوي هذه الطبقة على وحدات متعددة، كل منها مسؤول عن جزء محدد من النظام، مثل إدارة الإعلانات، المستخدمين، العقارات، التقييمات، وغيرها، مما يساهم في تنظيم منطق العمل وتسهيل التكامل بين واجهات النظام ومكوناته الداخلية.

1.2.4 سبب اختيار نموذج MVC

تم اختيار نموذج MVC (Model-View-Controller) كعمارة أساسية في بناء النظام نظرًا لما يقدمه من مزايا واضحة في تنظيم الكود، وفصل المسؤوليات، وتحسين قابلية الصيانة والتوسع. في مشروعنا، تم تطبيق هذا النموذج بطريقة مرنة، حيث:

- **Model** يمثل طبقة البيانات (Entities) والتعامل مع قاعدة البيانات باستخدام Entity Framework Core في ASP.NET Core.
- **Controller** يمثل الطبقة الوسيطة التي تعالج الطلبات الواردة من الواجهة، وتنفذ المهام وتعيد الاستجابة المناسبة.
- **View** تم تمثيلها من خلال واجهة المستخدم المبنية باستخدام React.js، والتي تتعامل مع البيانات من خلال استدعاء API والتفاعل مع المستخدم بشكل ديناميكي وسريع.

أسباب اختيار MVC بالتحديد:

- **سهولة الصيانة والتطوير:** يمكن تعديل الواجهة (React) دون التأثير على الواجهة الخلفية (backend)، والعكس صحيح.
- **قابلية التوسع:** يدعم النموذج إضافة ميزات جديدة بسهولة دون إرباك بنية النظام.
- **دعم ممتاز من ASP.NET Core:** الإطار يدعم نموذج MVC بشكل مدمج، مما يسهل إنشاء واجهات برمجية (APIs) منظمة وسريعة.
- **كفاءة الأداء:** بفضل الفصل بين المكونات، يتم تحميل الموارد بكفاءة ويمكن النظام من الاستجابة للطلبات بشكل سريع ومرن.
- **فصل الاهتمامات (Separation of Concerns):** كل جزء من النظام مسؤول عن مهمة محددة، مما يجعل الكود أكثر وضوحًا وأسهل في التتبع.

3.2.5 البدائل المعمارية (Alternative Architectures)

عند تصميم البنية المعمارية للنظام، تم دراسة عدة نماذج معمارية شائعة، وتم تقييمها بناءً على متطلبات المشروع من حيث الأداء، القابلية للتوسعة، سهولة الصيانة، ووضوح الفصل بين المكونات. ومن بين النماذج التي تم النظر فيها:

• 1. Monolithic Architecture

الوصف: تعتمد على تجميع جميع أجزاء النظام في تطبيق واحد، حيث تتكامل الواجهة، منطق الأعمال، وطبقة البيانات في نفس المشروع.

• الميزات:

- سهل في الإعداد والنشر.
- مناسب للمشاريع الصغيرة أو النماذج الأولية.

• القيود:

- صعوبة التوسع على المدى الطويل.
- التعديلات في جزء من النظام قد تؤثر على الأجزاء الأخرى.

سبب الاستبعاد: لا يتماشى مع الحاجة إلى قابلية التوسع واستقلالية الواجهة (React) عن الخادم. [50]

• 2. Microservices Architecture

الوصف: يتم تقسيم النظام إلى خدمات صغيرة مستقلة، كل منها مسؤولة عن جزء محدد من الوظائف.

• الميزات:

- مرونة عالية في التطوير والنشر.
- قابلية توسعة أفقية ممتازة.

• القيود:

- تعقيد في الإدارة والربط بين الخدمات.
- يتطلب بنية تحتية متقدمة (مثل Docker, Kubernetes، إدارة Gateway...).

سبب الاستبعاد: معقد جدًا بالنسبة لحجم المشروع الحالي، وقد يؤدي إلى زيادة العبء التقني دون حاجة حقيقية [51].

• 3. MVVM (Model–View–ViewModel)

الوصف: يُستخدم بشكل شائع في تطبيقات سطح المكتب أو التطبيقات الهجينة مثل WPF أو Angular ، حيث يتم ربط الواجهة بالبيانات بشكل تلقائي عبر ViewModel.

- الميزات:

- دعم قوي لربط البيانات. (Data Binding)
- يفصل الواجهة عن منطق العرض.

- القيود:

- لا يتماشى مباشرة مع React ، حيث لا يوجد مفهوم ViewModel بشكل صريح.
- سبب الاستبعاد: لا يتناسب مع طبيعة React ولا مع تصميم Web API في ASP.NET Core و [52]

3.4 جداول قواعد البيانات :

Table	Name of the table in the database	Table description
Users	User	Storing user information.
Permission	Role	Define user permissions.
Permissions and Users	UserRole	Associate users with permissions.
Messages	Messages	Storing user messages.
Ads	Advertisement	Storing advertisement data.
Properties	Property	Storing property data.
Services	Service	Storing service data.
Reviews	Review	User comments on services.
Save	Save	Storing user advertisements.
Complaints	Complaint	Storing complaints data.
Address	Address	Storing property or service addresses.
Service Type	TypeService	Specifying the type of service (plumbing, electricity, etc.).
Property Type	TypeProperty	Specifying the property type (apartment, villa, etc).

جدول : 19 قائمة جداول قواعد البيانات

Field Name	Field Type	Field Length	Null	Field Description
ID	Int	-	No	The user's primary key.
username	Varchar	256	No	The user's full name.
email	Varchar	256	Yes	The user's email address.
PhoneNumber	Varchar	256	Yes	The user's phone number.
password	Varchar	256	No	The user's password.
AddressId	Int	-	No	The foreign key refers to the address table.
CreateAt	Datetime	7	No	Used to store the creation date and time.
notification	Bool	1	No	Specifies whether the user wants to receive notifications.

جدول : 20 وصف المستخدمين

Field Name	Field Type	Field Length	Null	Field Description
ID	Int	-	No	Primary key for the permissions.
Name	Varchar	256	No	Permission name

جدول : 21 الصلاحيات

Field Name	Field Type	Field Length	Null	Field Description
UserId	No	-	Int	A foreign key that refers to the users table
RoleId	No	-	Int	A foreign key that refers to the permissions table

جدول : 22 وصف الجدول الخاص الصلاحيات والمستخدمين

Field Name	Field Type	Field Length	Null	Field Description
ID	Int	-	No	The primary key for messages
SenderId	Int	-	No	Indicates the user who sent the message
ReceiverId	Int	-	No	Foreign key refers to the User table
Message	No	256	No	The text of the message sent between users
Timestamp	Datetime	7	No	The date and time the message was sent

جدول : 23 وصف الجدول الخاص بالرسائل

Field Name	Field Type	Field Length	Null	Field Description
ID	Int	-	No	As a primary key.
startAt	Date	7	No	Ad start date.
endAt	Date	7	No	Ad end date.

جدول : 24 وصف الجدول الخاص بالإعلانات

Field Name	Field Type	Field Length	Null	Field Description
ID	No	-	Int	Primary key for storage
UserId	No	-	Int	A foreign key that refers to the User table.
PropertyId	No	-	Int	A foreign key that refers to the Property table if the storage is for real estate.
ServiceId	No	-	Int	A foreign key that refers to the Service table if the storage is for a service.

جدول : 25 وصف جدول الخاص في الحفظ

Field Name	Field Type	Field Length	Null	Field Description
ID	Int	-	No	The primary key for the property.
Description	Navchar	Max	No	Property description
StartAt	Datetime	7	No	Property creation time
updateAt	Datetime	7	No	Property data update time
Price	Decimal	400	No	Property price
Lat	Decimal	400	No	Latitude (specifies the location north or south).
Lng	Decimal	400	No	Longitude (specifies the location east or west).
UsersID	Int	-	No	Foreign key referring to the User table.
AdvertisementID	Int	-	No	Foreign key referring to the Advertisement table.
TypeId	Int	-	No	Foreign key referring to the Type table.
AddressId	Int	-	No	Foreign key referring to the Address table.

جدول : 26 وصف الجدول الخاص بالعقارات

Field Name	Field Type	Field Length	Null	Field Description
ID	Int	-	No	Primary key for the service.
Description	Varchar	256	No	Service description.
PriceRange	Decimal	400	No	Service price.
UsersID	Int	-	No	Foreign key referring to the User table.
AdvertisementID	Int	-	Yes	Foreign key referring to the Advertisement table.
TypeId	Int	-	No	Foreign key referring to the Type table.
AddressId	Int	-	No	Foreign key referring to the Address table.

جدول : 27 جدول الخدمة

Field Name	Field Type	Field Length	Null	Field Description
ID	Int	-	No	Primary key for the review.
Description	Varchar	256	No	Write a comment
UsersID	Int	-	No	Foreign key that refers to the User table.
PropertyId	Int	-	Yes	Foreign key that refers to the Property table if the review is for real estate.
ServiceId	Int	-	Yes	Foreign key that refers to the Service table if the review is for a service.
Rating	Decimal	400	No	Review rating (1-5)
CreateAt	Datetime	7	No	Reference creation time.
UpdateAt	Datetime	7	No	Reference update time

جدول : 28 وصف الجدول الخاص بالمراجعات

Field Name	Field Type	Field Length	Null	Field Description
ID	Int	-	No	Primary key for complaints.
Name	Int	-	No	Complaint title.
Description	Varchar	255	No	Complaint content.
Status	Varchar	255	No	Complaint status (open/closed/under review).
UsersID	Int	-	No	Foreign key referring to the User table.
CreatedDate	Datetime	7	No	Complaint creation time.

جدول : 29 وصف الجدول الخاص بالشكاوي

Field Name	Field Type	Field Length	Null	Field Description
ID	No	-	Int	Occupation is the primary key.
Name	No	255	Varchar	Occupation name.

جدول : 30 وصف الجدول الخاص بنوع المهنة

Field Name	Field Type	Field Length	Null	Field Description
ID	No	-	Int	Property master key
Name	No	255	Varchar	Property type name

جدول : 31 وصف جدول العقار

Field Name	Field Type	Field Length	Null	Field Description
ID	Int	-	No	Primary key title.
Name	Varchar	255	No	Title

جدول : 32 وصف جدول العنوان

Field Name	Field Type	Field Length	Null	Field Description
ID	Int	-	No	Primary key for the Images table.
image	Varchar	255	No	Image link
PropertyId	Int	-	Yes	Foreign key referring to the Property table
ServiceId	Int	-	Yes	Foreign key referring to the Service table
ComplaintId	Int	-	Yes	Foreign key referring to the Complaint table

جدول : 33 وصف جدول الصور

الفصل الرابع : تنفيذ النظام

المقدمة

code

واجهات النظام (الشاشات)

4.1 المقدمة

تعتبر مرحلة تصميم النظام من المراحل المهمة في عملية بناء النظام وتطويره، لأنها تعطي فكرة كاملة عن جميع أجزاء النظام بالرسومات التوضيحية كما تسهل على المبرمجين عملية بناء النظام بالشكل الصحيح. ويجب مراعاة أذواق المستخدمين للنظام عند التصميم ورغباتهم وضمان سهولة استخدام النظام ومراعاة جميع الفئات المستخدمة للنظام على اختلاف مستوياتهم.

Back-End Tools & Libraries

- **ASP.NET Core:** A powerful framework for building web APIs and applications. [1]
- **Microsoft.EntityFrameworkCore:** An ORM to manage databases using C# objects. [2]
- **Microsoft.EntityFrameworkCore.SqlServer:** EF Core provider for SQL Server database. [2]
- **Microsoft.AspNetCore.Identity.EntityFrameworkCore:** User account and role management system. [3]
- **Microsoft.AspNetCore.Authentication.JwtBearer:** JWT-based authentication middleware. [4]
- **Azure.Storage.Blobs:** Library to manage file storage in Azure Blob Storage. [5]
- **Mapster:** A simple and fast object mapping library. [6]
- **RestSharp:** A library for making HTTP requests from the server. [7]
- **Serilog.AspNetCore:** Logging library for structured and flexible logs. [8]
- **Swashbuckle.AspNetCore:** Automatically generates Swagger documentation for APIs. [9]
- **Microsoft.AspNetCore.SignalR:** Library for real-time communication (chat, notifications). [10]

Front-End Tools & Libraries

- **ReactJS:** A front-end library for building interactive UIs with components. [11]
- **React Router DOM:** Routing library to navigate between pages in React. [12]
- **React Hook Form:** Manage and validate forms easily in React. [13]
- **Yup:** JavaScript schema validation library for forms. [14]
- **Formik:** Makes working with forms easier in React. [15]
- **Lucide React:** Beautiful and open-source icon library. [18]
- **React Toastify:** Library to display toast notifications. [19]
- **Recharts:** A React-based library to render charts and graphs. [20]
- **Swiper:** Powerful slider library for carousels and swipers. [21]
- **Leaflet + React-Leaflet:** Map libraries to create interactive maps. [22]

- **Leaflet Routing Machine:** Adds routing and direction features to Leaflet maps. [23]
- **i18next + React-i18next:** Internationalization library to support multiple languages. [24]
- **Emoji Picker React:** Component to select and insert emojis. [25]
- **Dayjs + Date-fns:** Libraries for manipulating and formatting dates. [26]
- **Axios:** Promise-based HTTP client for making API requests. [27]
- **jwt-decode:** Decode JWT tokens in the frontend. [28]
- **Lodash:** Utility functions for arrays, objects, strings, etc. [29]

Deployment & Tools

- **Postman:** Tool to test and debug REST APIs. [30]
- **Swagger:** Auto-generates and documents API endpoints in-browser. [9]
- **Netlify:** Hosting service for front-end apps (React, HTML, static sites). [31]
- **Azure:** Cloud platform for hosting, storage, and services. [5]
- **UltraMsg:** WhatsApp API service for sending automated messages. [33]
- **SQL Server:** Relational database management system by Microsoft. [34]
- **GitHub:** Code hosting and version control platform. [35]
- **Trello:** Kanban-style tool for task management. [42]
- **Visual Studio Code (VS Code):** Lightweight, powerful code editor developed by Microsoft, used primarily for front-end development. [44]
- **Visual Studio 2022:** An integrated development environment (IDE) used mainly for ASP.NET Core and backend development. [45]
- **draw.io:** Free diagramming tool used to create flowcharts, ER diagrams, and system architecture designs. [46]
- **Microsoft Word:** Word processing software used for writing reports and documentation. [47]
- **Microsoft PowerPoint:** Presentation software used for creating slides to present project ideas and progress. [48]
- **Google Maps:** Web-based map service used in the project for location services, distances, and map visualization. [49]

Code 4.2

في هذا الجزء من المشروع، ركزنا على مرحلة مهمة وهي "المصادقة" وتسجيل حساب المستخدم. عملية التسجيل تم تصميمها بحيث تمر بواحدة من طريقتين:

- عبر رقم الواتساب.
- عبر البريد الإلكتروني.

وبعد اختبار وسيلة التسجيل، يُطلب من المستخدم إكمال باقي البيانات الأساسية مثل: الاسم، رقم الهاتف، العنوان، الدور، وكلمة المرور، وغيرها من المتطلبات.

اهتمنا بشكل خاص بجعل نظام المصادقة قوي ومرن، بحيث يخدم جميع أنواع المستخدمين سواءً المسجلين عن طريق الواتساب أو عن طريق الإيميل، ويضمن أمان عالي وسهولة في الاستخدام.

```
1 export const registerUser = async (userData) => {
2   try {
3     const { whatsapp, email, ...baseData } = userData;
4     const endpoint = "/Auth/Register" + (whatsapp ? "?WhatsApp=true" : "");
5
6     const payload = whatsapp
7       ? baseData
8       : { ...baseData, email };
9
10    const response = await api.post(endpoint, payload);
11    return response.data;
12  } catch (error) {
13    throw handleApiError(error);
14  }
15  };
```

إنشاء حساب


عقاراتي

التسجيل عبر واتساب

أدخل الاسم

أدخل البريد الإلكتروني

أدخل 9 أرقام بدون صفر في البداية

اختر العنوان

اختر نوع الحساب

أدخل كلمة المرور

هل تمتلك حساب بالفعل؟

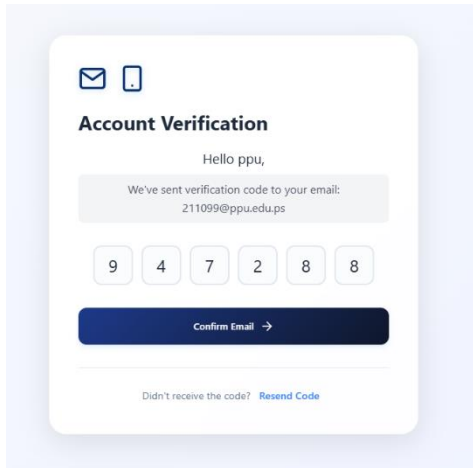
واجهة 1 إنشاء حساب

الكود هذا ، وظيفته تسجيل المستخدم في النظام. يعطي الشخص خيارين: يسجل يا إما برقم الواتساب أو بالإيميل. حسب الخيار اللي اختاره، الكود يجهز البيانات المناسبة ويرسلها للسيرفر عن طريق رابط التسجيل. ولو صار أي خطأ أثناء العملية، فيه طريقة جاهزة تتعامل مع الخطأ بدون ما تخرب التجربة على المستخدم.

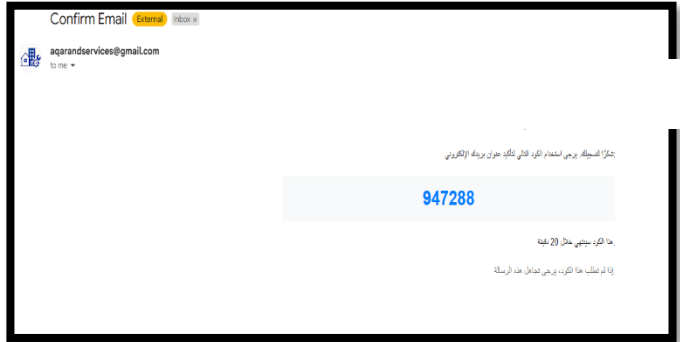
قسم الواجهة الخلفية (backend) المسؤول عن انشاء الحساب :

```
1
2 [HttpPost("Register")]
3 public async Task<IActionResult> Register(AuthRegisterDTOS request, bool? WhatsApp)
4 {
5     if (ModelState.IsValid)
6     {
7
8         if (request.role == "admin")
9         {
10            return BadRequest(new { message = "you cannot create an account Admin" });
11        }
12
13        if (request.Email is not null || WhatsApp is false)
14        {
15
16            ApplicationUser applicationEmail = await userManager.FindByEmailAsync(request.Email);
17
18            if (applicationEmail is not null)
19            {
20                if (applicationEmail.EmailConfirmed == false && applicationEmail.PhoneNumberConfirmed == false)
21                {
22                    await userManager.DeleteAsync(applicationEmail);
23                }
24                else
25                {
26                    return BadRequest(new { message = "Email already registered" });
27                }
28            }
29        }
30    }
31 }
32
```

جزء مبسّط من عملية إنشاء API لتسجيل الحساب، حيث يتم توجيه المستخدم بعد التسجيل إلى صفحة التوثيق في حال كانت البيانات المدخلة صحيحة.



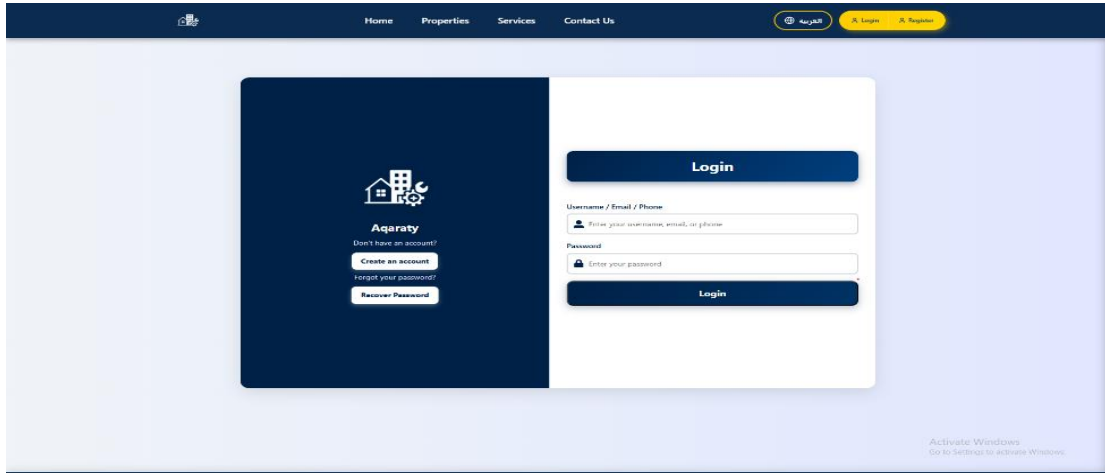
واجهة 3 ادخال كود التحقق



واجهة 2 رمز التحقق عبر الأيميل

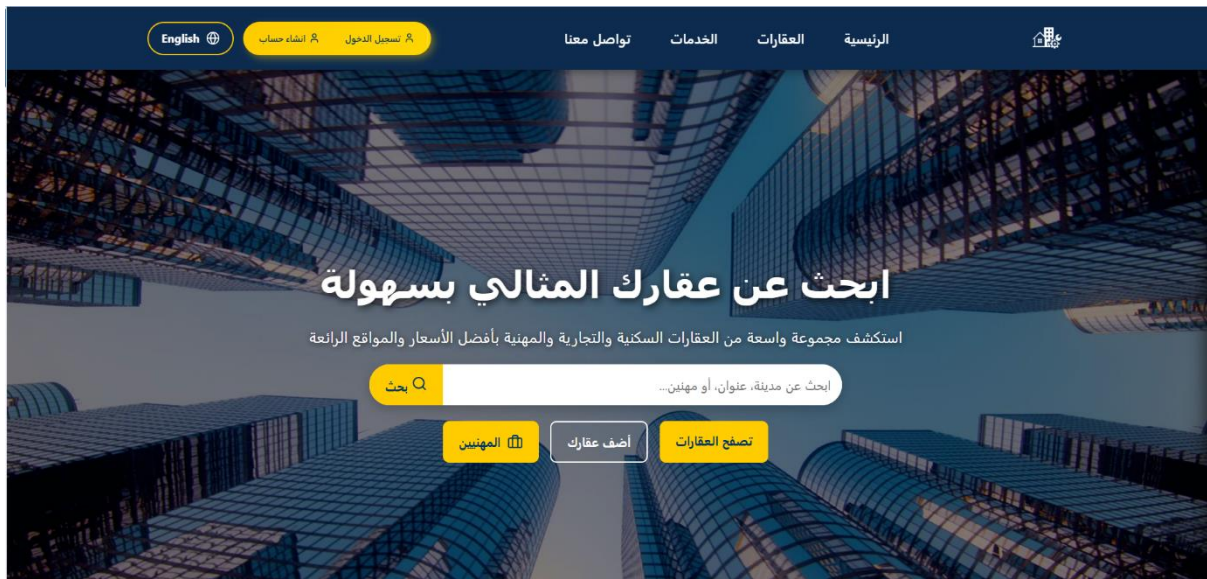
كمستخدم، قمت بتسجيل حسابي باستخدام بريدي الجامعي وليس عن طريق تطبيق واتساب. بعد إتمام عملية التسجيل، ظهرت لي الواجهة رقم (3) التي تحتوي على رسالة تفيد بأنه تم إرسال كود التحقق إلى بريدي الإلكتروني. بعد ذلك، انتقلت إلى الواجهة رقم (2)، حيث طُلب مني إدخال كود التحقق الذي وصلني من أجل تأكيد الحساب وإتمام عملية التحقق بنجاح.

4.3 واجهات النظام (الشاشات)



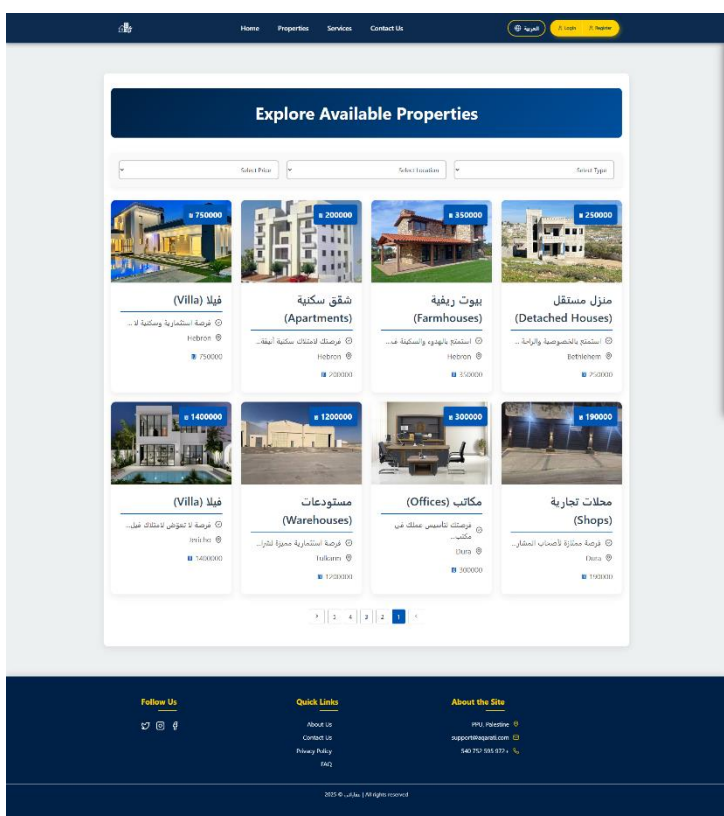
واجهة 4 تسجيل الدخول

واجهة تسجيل دخول بسيطة تحتوي على حقل لإدخال البريد الإلكتروني أو رقم الهاتف أو اسم المستخدم، وحقل لكلمة المرور، بالإضافة إلى زر لإنشاء حساب جديد وآخر لاسترجاع كلمة المرور.

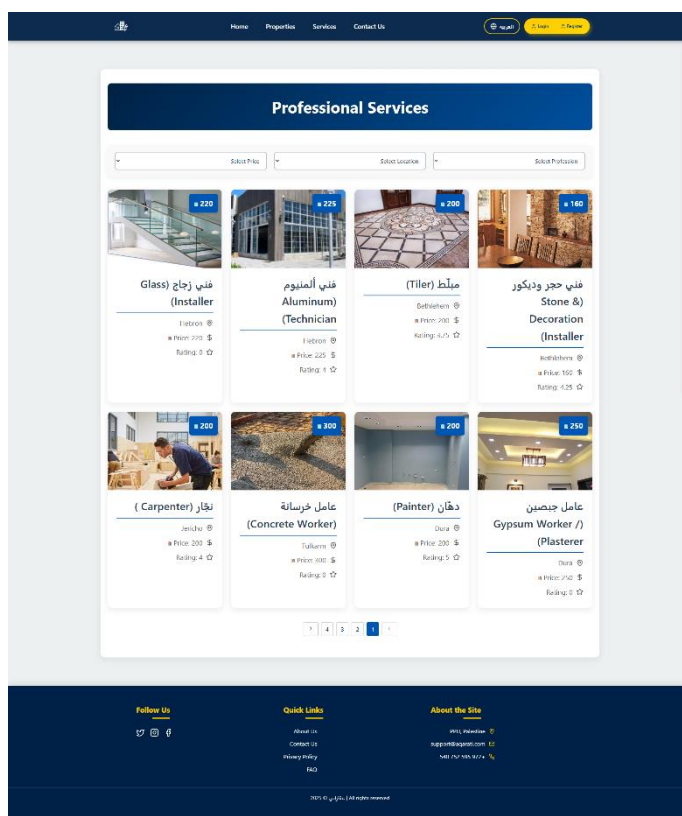


واجهة 5 الرئيسية

الواجهة الرئيسية تتضمن في أعلاها شريط تنقل (Navbar) يتيح للمستخدم التنقل بين صفحة الإعلانات و صفحة الخدمات و صفحة العقارات، وفي منتصف الصفحة يوجد شريط بحث يتيح البحث باستخدام العنوان أو اسم المهنة أو اسم الخدمة أو حتى اسم مزود الخدمة، مما يساعد المستخدم في الوصول إلى ما يبحث عنه بسهولة ووضوح.

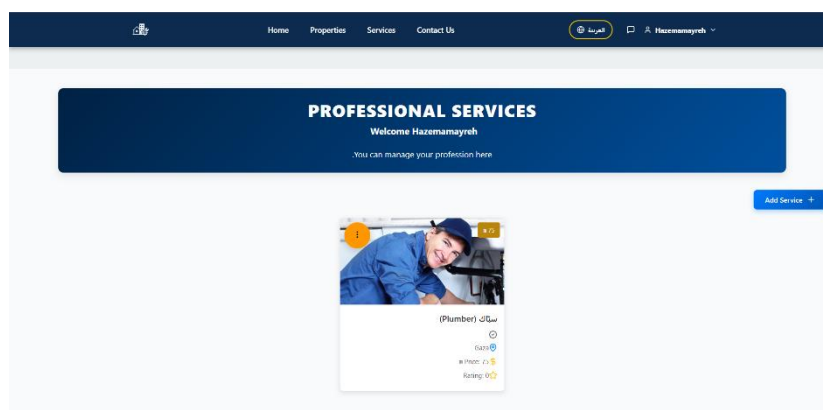


واجهة 6 تصفح الاعلانات



واجهة 7 تصفح المهن

تحتوي صفحات العقارات والخدمات على تصميم موحّد يبدأ بشرط فلتر في أعلى الصفحة يمكّن المستخدم من اختيار نوع العقار أو نوع الخدمة، وتحديد الموقع والسعر لتسهيل عملية البحث. بعد ذلك، تُعرض قائمة بالعناصر، حيث تحتوي صفحة العقارات على جميع العقارات المتوفرة مثل الفلل، الشقق، البيوت الريفية وغيرها، مع عرض تفاصيل مثل العنوان، وصف مختصر، والسعر. أما صفحة الخدمات، فتعرض جميع الخدمات المتاحة بمهن مختلفة مثل الفنيين، الدهانين، البلاطيين وغيرهم، مع تفاصيل تشمل نوع الخدمة، اسم المهنة، الموقع، وسعر الخدمة، مما يساعد المستخدم على استعراض الخيارات المتوفرة بسهولة ودقة.



واجهة 8 اضافة عقار او مهنة

الصفحة الخاصة في اضافة المهنة وبماثلها صفحة اضافة العقار بحيث يتحكم بكل المنشورات الخاصة به من هذه الصفحات

جميع الإعلانات

اكتشف أفضل العروض والإعلانات

نوع الإعلان: الكل

الإعلان رقم 183

31/05/2025 - 29/05/2025

عقار

HHHHH

أراض صناعية (industrial lands)

TEST

العنوان: Qalqiya

السعر: 0

التقييم: noRating

تقييم 0

الإعلان رقم 182

31/05/2025 - 29/05/2025

خدمة

saifaldin

عامل بناء (Construction Worker)

assdsad

العنوان: Hebron

نطاق السعر: 0

التقييم: noRating

تقييم 0

الإعلان رقم 177

31/05/2025 - 15/03/2025

خدمة

soso

عامل بناء (Construction Worker)

service two

العنوان: Tubas

نطاق السعر: 0

التقييم: noRating

تقييم 0

واجهة 9 صفحة تصفح

تعرض الصفحة الرئيسية جميع الإعلانات المتوفرة مع إمكانية فلترتها حسب النوع (عقار أو خدمة)، حيث تُعرض كل بطاقة إعلان بتفاصيل واضحة تشمل رقم وتاريخ الإعلان، نوعه، العنوان، نطاق السعر، التقييم، والموقع، والصورة مما يتيح للمستخدم تصفح العروض بسهولة وسرعة.

محفوظاتي

استعرض كافة العناصر المحفوظة لديك

الخدمات

عامل بناء (Construction Worker)

اسم المستخدم: saifaldin

الوصف: assdsad

نطاق السعر: 122

احذف من المفضلة

واجهة 10 المحفوظات

تعرض واجهة المحفوظات العقارات أو الخدمات التي قام المستخدم بحفظها مسبقاً، مع عرض تفاصيلها الأساسية لتسهيل الرجوع إليها لاحقاً.

الأعدادات

إدارة معلومات حسابك وإعداداته

المعلومات الشخصية

اسم المستخدم:	admin
تأكيد البريد الإلكتروني:	نعم
تأكيد الهاتف:	نعم
عضو منذ:	0001-01-01
الدور:	admin

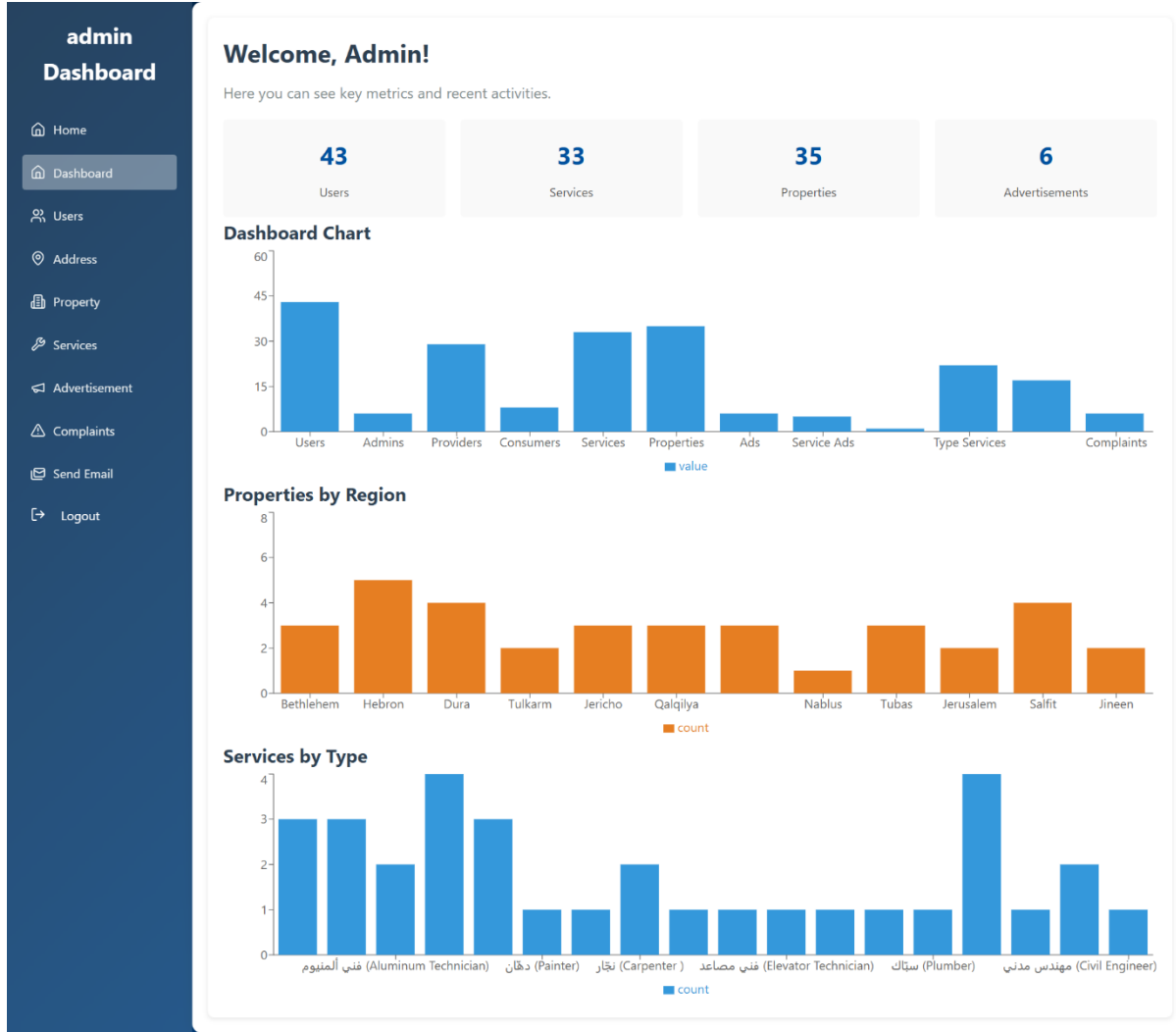
معلومات الاتصال

البريد الإلكتروني:	admin@aqar.com	تعديل
رقم الهاتف:	0020259575254	تعديل
العنوان:	Tubas	تعديل
الإشعارات:	مفعل	تغيير الحالة

تغيير كلمة المرور

واجهة 11 الأعدادات

تتيح صفحة الإعدادات للمستخدم عرض معلوماته الشخصية مثل الاسم، حالة تأكيد البريد والهاتف، تاريخ الانضمام، والدور في الموقع. كما يمكنه تعديل بياناته الأساسية وتفعيل أو إيقاف إشعارات الرسائل أو التقييمات سواء عبر البريد الإلكتروني أو الواتساب. في نهاية الصفحة، يمكن للمستخدم تغيير كلمة المرور الخاصة به بكل سهولة.



واجهة 12 صفحة الأدمن

توفر صفحة لوحة التحكم (Dashboard) للإدارة نظرة شاملة على أداء النظام، حيث تعرض إحصائيات رئيسية مثل عدد المستخدمين، الخدمات، العقارات، والإعلانات. تشمل الصفحة رسوماً بيانية تفاعلية توضح التوزيع الجغرافي والنوعي للمحتوى، مع إمكانية التنقل بين أقسام الإدارة المختلفة مثل المستخدمين والعناوين والشكاوى. كما يمكن للإدارة إرسال رسائل للمستخدمين عبر البريد الإلكتروني أو تطبيق واتساب.

الفصل الخامس : فحص النظام

المقدمة

التحقق

اختبار التحقق

اختبار نقاط الاتصال

QA Manual Testing

5.1 المقدمة

في هذه المرحلة، يتم اختبار النظام للتأكد من أنه يعمل بالشكل الصحيح وبدون مشاكل. الهدف من الاختبار هو التأكد من أن كل جزء من النظام يؤدي وظيفته كما هو متوقع، وأنه يحقق المتطلبات التي تم تحديدها في بداية المشروع، سواء كانت وظيفية (Functional) أو غير وظيفية (Non-functional). يتم هذا الفحص بعد الانتهاء من تصميم النظام وتنفيذه، ويُعتبر خطوة أساسية لضمان جودة المنتج قبل إطلاقه للمستخدمين. في هذا الفصل، سنستعرض كيف تم اختبار النظام، وما هي الإجراءات التي اتُبعت للتأكد من جاهزيته وكفاءته.

5.2 التحقق (Validation)

جزء أساسي من اختبار النظام هو التحقق من البيانات التي يدخلها المستخدم. الهدف من عملية التحقق هو التأكد من أن كل المعلومات المدخلة صحيحة، ومطابقة للشروط المطلوبة، وتساعد في حماية النظام من الاستخدام الخاطئ أو المتعمد. التحقق يتم في جميع النماذج داخل الموقع، ويشمل عدة خطوات منطقية، وهي كالتالي:

- 1. تحديد نوع المدخلات حسب الحقل**
كل حقل في النظام مصمم لاستقبال نوع معين من البيانات (مثل أرقام، نصوص، بريد إلكتروني). يتم رفض أي بيانات لا تتطابق مع نوع الحقل، لضمان دقة المعلومات.
- 2. رفض البيانات الخاطئة أو غير المتوافقة**
مثلاً، إذا تم إدخال بريد إلكتروني بدون رمز "@" أو رقم هاتف بدون كود الدولة، يعرض النظام رسالة تنبيه ويمنع التقديم حتى يتم تصحيح الخطأ.
- 3. منع ترك الحقول المطلوبة فارغة**
لا يمكن للمستخدم متابعة العملية إذا ترك أي حقل مطلوب بدون تعبئته. تظهر رسالة توضح له الحقول الناقصة أو التي يجب تصحيحها.
- 4. التحقق من وجود المستخدم في قاعدة البيانات**
في عمليات مثل تسجيل الدخول أو تأكيد الحساب، يتم فحص قاعدة البيانات للتأكد من أن الحساب موجود فعلاً، ولا يُسمح بتسجيل دخول لحساب غير مُسجل.
- 5. التحقق من التكرار**
إذا حاول المستخدم إدخال بريد إلكتروني أو رقم هاتف سبق استخدامه من قبل، يتم رفضه برسالة توضح أن البيانات مستخدمة مسبقاً، لتجنب التكرار.
- 6. التأكد من تنسيق المدخلات (Formatting)**
يتم التحقق من أن البيانات المدخلة، مثل كلمة المرور، تطابق معايير الأمان (مثل احتواءها على أحرف كبيرة، أرقام، ورموز)، وأنها ضمن الطول المسموح.
- 7. إعطاء ملاحظات واضحة للمستخدم**
النظام مصمم ليرد على المستخدم بشكل فوري عند وجود أي خطأ، ويعطيه رسالة واضحة عن الخطأ وطريقة إصلاحه، مما يسهل تجربة الاستخدام ويقلل من الإرباك.
- 8. اختبار سيناريوهات متعددة**
تم تجربة النظام باستخدام عدة حالات، مثل إدخال بيانات غير متوقعة، أو ترك الحقول فارغة، أو استخدام تنسيقات مختلفة، لضمان أن النظام قادر على التعامل مع كل الظروف.

5.3 اختبار التحقق

```
if (request.Email is not null || WhatsApp is false)
{
    ApplicationUser applicationEmail = await userManager.FindByEmailAsync(request.Email);

    if (applicationEmail is not null)
    {
        if (applicationEmail.EmailConfirmed == false && applicationEmail.PhoneNumberConfirmed == false)
        {
            await userManager.DeleteAsync(applicationEmail);
        }
        else
        {
            return BadRequest(new { message = "Email already registered" });
        }
    }

    if (applicationPhone.EmailConfirmed == false && applicationPhone.PhoneNumberConfirmed == false)
    {
        await userManager.DeleteAsync(applicationPhone);
    }
    else
    {
        return BadRequest(new { message = "Phone number already registered" });
    }

    if (applicationName.EmailConfirmed == false && applicationName.PhoneNumberConfirmed == false)
    {
        await userManager.DeleteAsync(applicationName);
    }
    else
    {
        return BadRequest(new { message = "Username already registered" });
    }
}
```

```
1 const schema = yup.object().shape({
2   name: yup
3     .string()
4     .min(3, t("errors.nameMin"))
5     .max(19, t("errors.nameMax"))
6     .required(t("errors.nameRequired"))
7     .matches(/^[^S+$/], t("errors.nameNoSpaces")),
8   email: yup
9     .string()
10    .email(t("errors.emailInvalid"))
11    .when("whatsApp", {
12      is: false,
13      then: (schema) => schema.required(t("errors.emailRequired")),
14    }),
15   phone: yup
16     .string()
17     .required(t("errors.phoneRequired"))
18     .matches(/^[1-9]\d{8}$/ , t("errors.phoneInvalid")),
19   addressId: yup
20     .number()
21     .typeError(t("errors.addressRequired"))
22     .required(t("errors.addressRequired")),
23   role: yup
24     .string()
25     .oneOf(["consumer", "provider"], t("errors.roleRequired"))
26     .required(t("errors.roleRequired")),
27   password: yup
28     .string()
29     .min(6, t("errors.passwordMin"))
30     .max(19, t("errors.passwordMax"))
31     .matches(
32       /^(?=.*[a-z])(?=.*[A-Z])(?=.*[!@#_]).+$/,
33       t("errors.passwordComplexity")
34     )
35     .required(t("errors.passwordRequired")),
36   whatsApp: yup.boolean(),
37 });
38
```

في النظام، عملية التحقق (Validation) تصبح على مرحلتين:

المرحلة الأولى تكون من جهة الواجهة (Front-End)، وهي الجزء الذي يتفاعل معه المستخدم بشكل مباشر. في هذه المرحلة، يتم التحقق من البيانات قبل إرسالها لل خادم، مثل التأكد من أن الحقول ممتلئة، وأن الإيميل بصيغة صحيحة، أو أن رقم الجوال يبدأ برمز الدولة. الهدف من التحقق هنا هو تحسين تجربة المستخدم وتقليل الضغط على السيرفر عن طريق منع الأخطاء من البداية.

المرحلة الثانية تكون من جهة الخادم (Back-End)، وهي الأهم، حيث يكون التحقق بشكل أساسي ونهائي من قبل الواجهة الخلفية حيث أنها تضمن دخول بيانات صحيحة إلى قاعدة البيانات. في هذه المرحلة، يتم فحص أمور أكثر دقة، مثل التأكد إذا كان البريد الإلكتروني أو رقم الجوال مستخدم من قبل، أو إذا المستخدم موجود فعلاً في قاعدة البيانات.

بالتالي، وجود التحقق في الجهتين مهم. الواجهة تقدّم تجربة استخدام مرنة وسريعة، و الخادم يحمي النظام من التلاعب أو الأخطاء التي يمكن تتجاوز الواجهة

التحقق من جهة الواجهة (Front-End Validation)

قبل ما يتم إرسال البيانات إلى السيرفر، يتم التحقق منها مباشرة داخل المتصفح باستخدام مكتبة **yup** الهدف هو إعطاء المستخدم ملاحظات فورية لتعديل أخطائه بدون ما ينتظر رد من السيرفر.

بعض الأمثلة:

- الاسم: **(name)** لازم يكون بين 3 إلى 19 حرف، وما يحتوي على مسافات.
- الإيميل: **(email)** يتم التحقق من الصيغة، ويكون مطلوب فقط إذا لم يكن التسجيل عبر واتساب.
- رقم الهاتف: **(phone)** يجب أن يكون مكون من 9 أرقام، ويبدأ برقم من 1 إلى 9.
- العنوان: **(addressId)** لا يمكن تركه فاضي.
- الدور: **(role)** لازم يكون إما "مستهلك" أو "مزود خدمة".
- كلمة المرور: **(password)** يجب أن تحتوي على حرف كبير، حرف صغير، ورمز، وبطول لا يقل عن 8 ولا يزيد عن 19.

لو صار أي خطأ، النظام يعرض رسالة توضح للمستخدم إيش هو الخطأ تحديداً، زي "كلمة المرور ضعيفة" أو "الإيميل غير صالح".

التحقق من جهة الخلفية (Back-End Validation)

بعد ما يتم التأكد من صحة البيانات في الواجهة، ترسل إلى الخادم، وهناك يتم التحقق مرة ثانية لكن بشكل أدق . الهدف هو التأكد من الأمور الحساسة التي لا يمكن ان نعتمد عليها فقط من قبل واجهة المستخدم.

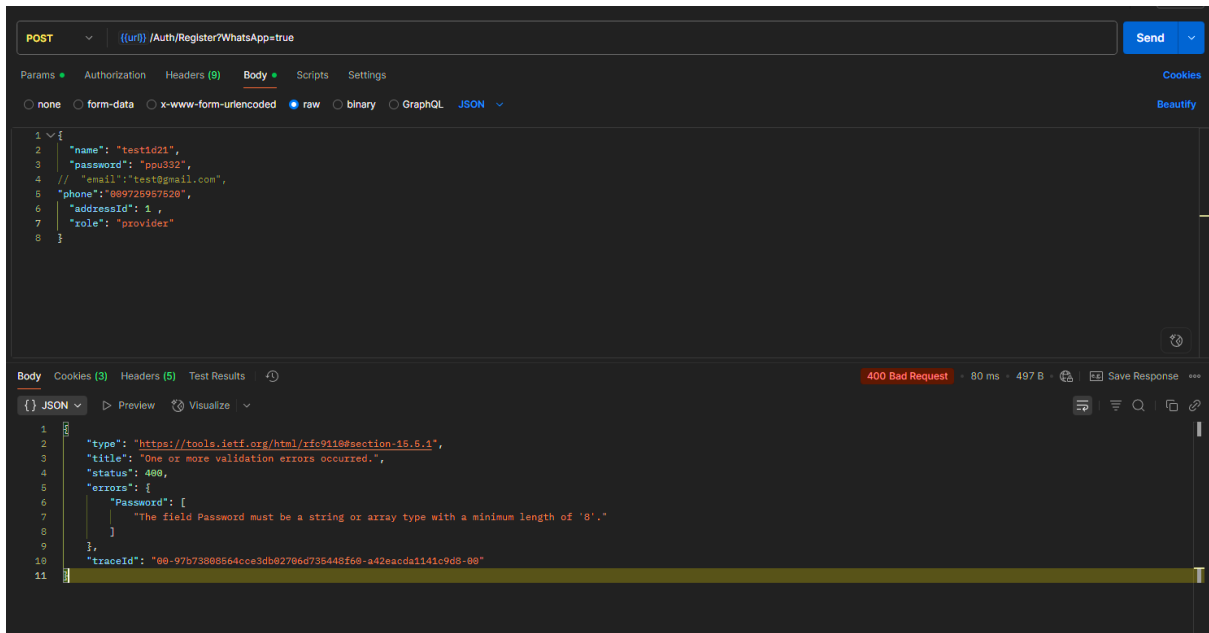
بعض الأمثلة من رسائل الخطأ اللي ممكن ترجع من الواجهة الخلفية (backend):

- "You cannot create an account Admin" → لو حاول يسجل كـ أدمن
- "Email already registered" → لو الإيميل مستخدم مسبقاً
- "Phone number already registered" → لو رقم الجوال مسجل من قبل
- "Username already registered" → لو الاسم موجود مسبقاً

5.4 اختبار نقاط الاتصال (Endpoint Testing)

في هذه المرحلة، تم استخدام أداة Postman لاختبار نقاط الاتصال الخاصة بواجهة الـ API في النظام. الهدف من هذا النوع من الاختبار هو التأكد من أن كل مسار (Endpoint) يتفاعل بشكل صحيح عند إرسال الطلبات المختلفة إليه، سواء كانت طلبات تسجيل، تسجيل دخول، تأكيد بريد، أو إعادة تعيين كلمة المرور. تم تنفيذ هذه الاختبارات للتأكد من دقة الردود، ومعالجة الأخطاء بشكل مناسب، وسلامة البيانات المتبادلة بين العميل والخادم.

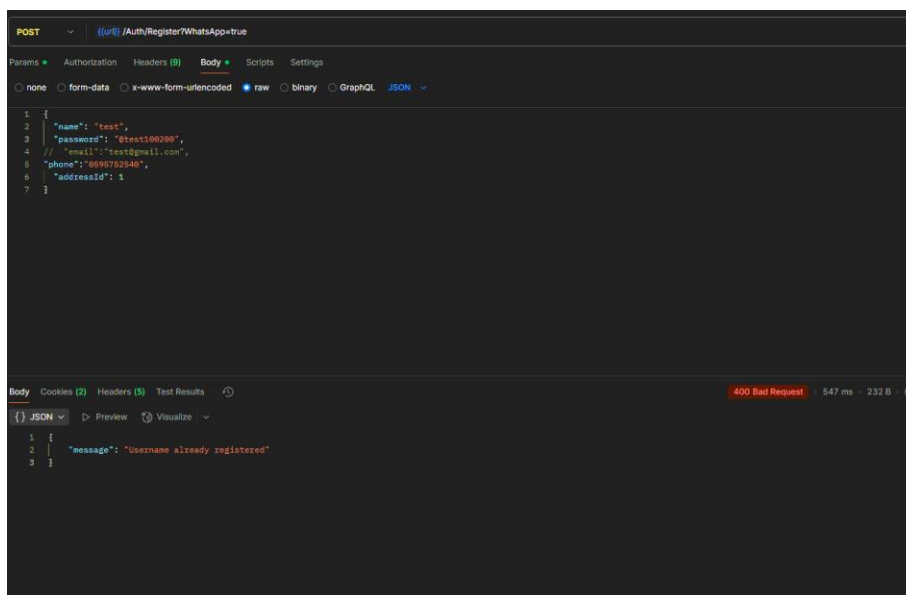
1. انشاء حساب بكلمة مرور ضعيفة



Back-end reply

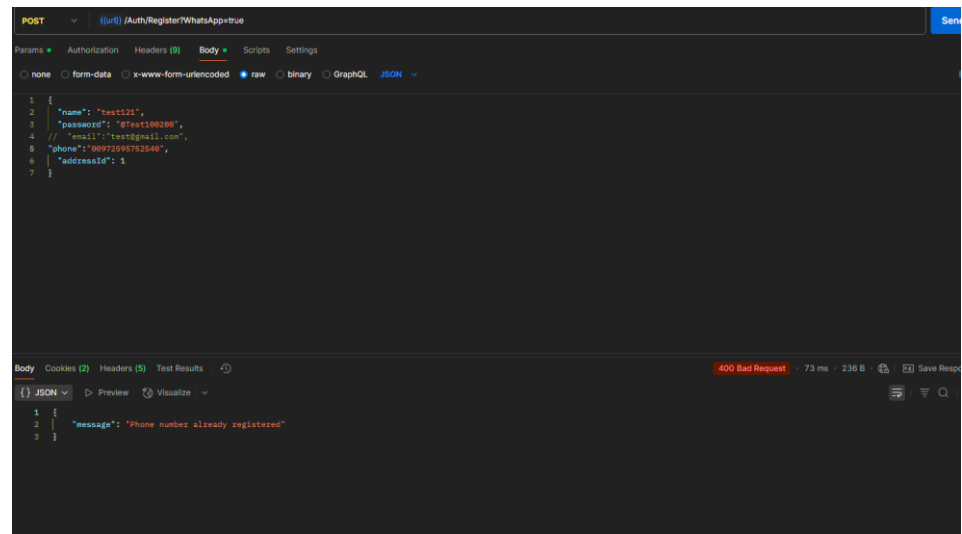
```
Password": [  
    "The field Password must be a string or array type with a minimum  
length of '8'."  
]
```

2. اسم مستخدم موجود من قبل :



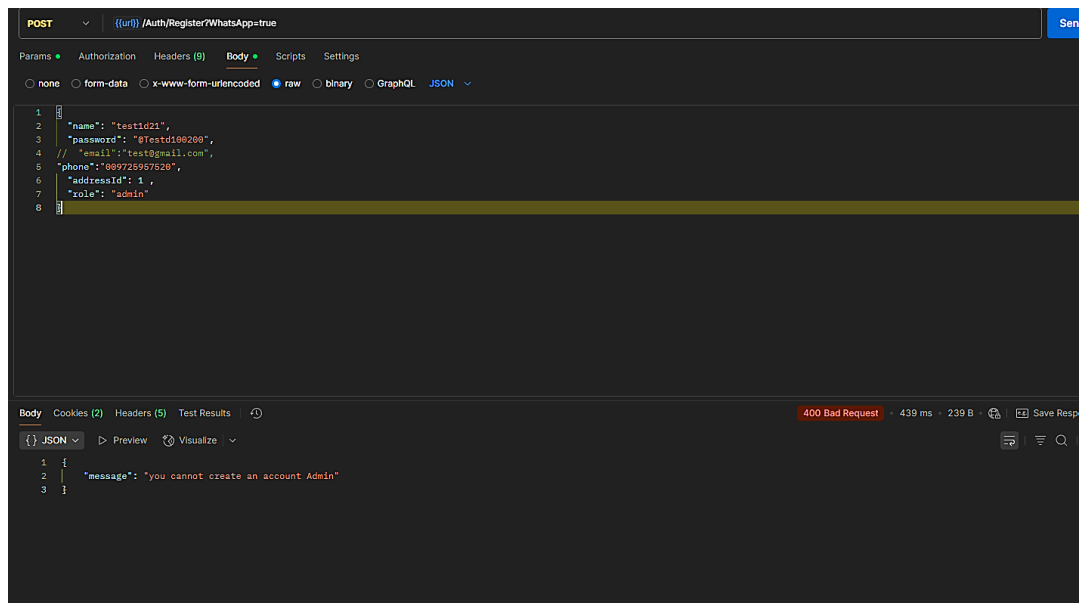
"message": "Username already registered"

3. التسجيل برقم هاتف موجود من قبل :



"message": "Phone number already registered"

4. التسجيل كمدير نظام :



"message": "you cannot create an account Admin"

هذه بعض الأمثلة للرسائل التي يرجعها (backend) في حال أرسل المستخدم طلب غير صحيح (خطأ في البيانات أو مخالفة لشروط النظام). الهدف من هذه الرسائل هو توضيح للمستخدم أو المطور السبب الحقيقي لفشل العملية، من أجل تصحيح الخطأ بسهولة .

5.5.1 الاختبار اليدوي (QA Manual Testing)

في هذه المرحلة، تم اختبار النظام يدويًا من خلال استخدام الواجهات كما يفعل المستخدم . تم التنقل بين الصفحات وتجربة عمليات التسجيل، تسجيل الدخول، التحقق، وإرسال البيانات، بهدف التأكد من أن جميع الوظائف تعمل بشكل سليم، وأن الأزرار والنماذج تؤدي دورها كما هو متوقع دون حدوث أية مشكلات.

السيناريو الأول: إرسال بيانات فارغة

الوصف: المستخدم يضغط على زر انشاء حساب بدون تعبئة أي حقل.
النتيجة المتوقعة: النظام يظهر رسائل خطأ توضح أن الحقول مطلوبة (مثل "الاسم مطلوب"، "رقم الهاتف مطلوب"، "كلمة المرور مطلوبة" إلخ)، ولا يتم إرسال الطلب للسيرفر.

إنشاء حساب

التسجيل عبر واتساب

الاسم الكامل
أدخل الاسم
⚠ يجب أن يكون الاسم 3 أحرف على الأقل.

البريد الإلكتروني
أدخل البريد الإلكتروني
⚠ البريد الإلكتروني مطلوب.

رقم الهاتف
أدخل 9 أرقام بدون صفر في البداية (00972)
⚠ رقم الهاتف مطلوب.

العنوان
اختر العنوان
⚠ يجب اختيار عنوان.

حدد دورك
اختر نوع الحساب
⚠ يجب اختيار دور (مستهلك أو مزود خدمة).

كلمة المرور
أدخل كلمة المرور
⚠ يجب أن تكون كلمة المرور 6 أحرف على الأقل.

إنشاء حساب

عقاراتي

هل تمتلك حساب بالفعل؟
تسجيل الدخول

السيناريو الثاني: إدخال إيميل بصيغة خاطئة

الوصف: المستخدم يكتب إيميل غير صالح (مثلاً hazem.com) ويحاول التسجيل.
النتيجة المتوقعة: تظهر رسالة تنبيه أن الإيميل غير صحيح، ولا يُسمح بإرسال البيانات إلا بعد تصحيحه.

The screenshot shows a registration form titled "إنشاء حساب" (Create Account) on the left and a dark blue sidebar with a house icon and the text "عقاراتي" (My Real Estate) on the right. The form fields are: "الاسم الكامل" (Full Name) with the value "HazemAgil", and "البريد الإلكتروني" (Email) with the value "hazemamayrehgamil.com". A red error message at the bottom states: "Please include an '@' in the email address. 'hazemamayrehgamil.com' is missing an '@'." There is also a "رقم الهاتف" (Phone Number) field.

السيناريو الثالث: استخدام إيميل أو رقم هاتف مسبقاً

الوصف: المستخدم يدخل إيميل أو رقم هاتف سبق وتم استخدامه في تسجيل حساب.
النتيجة المتوقعة: النظام يرجع رسالة من (backend) مثل "Email already registered" أو "Phone number already registered"، ولا يتم إنشاء حساب جديد.

The image contains two screenshots of the registration form. The left screenshot shows a red error message "Phone number already registered" above the "رقم الهاتف" (Phone Number) field, which contains "595752540". The right screenshot shows a red error message "Email already registered" above the "البريد الإلكتروني" (Email) field, which contains "ppu.edu.ps@211099". Both screenshots show the same form fields as the previous one, including "الاسم الكامل" (Full Name) and "العنوان" (Address).

السيناريو الرابع: إدخال كلمة مرور ضعيفة

الوصف: المستخدم يكتب كلمة مرور قصيرة أو بدون رموز.
النتيجة المتوقعة: تظهر رسالة تنبيه أن كلمة المرور غير قوية أو لا تحقق الشروط (مثلاً: لازم تحتوي على حرف كبير، صغير، ورمز).

استهتت نحو اسمين الذي ينبغي الحداد او السجيات.

كلمة المرور

⚠ يجب أن تكون كلمة المرور 6 أحرف على الأقل

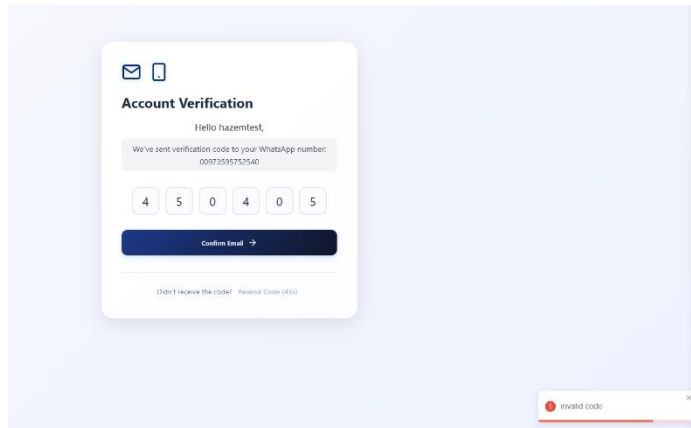
إنشاء حساب

كلمة المرور

⚠ يجب أن تحتوي كلمة المرور على أحرف كبيرة وصغيرة ورموز خاصة

السيناريو الخامس: إدخال كود تأكيد خاطئ

الوصف: بعد التسجيل، المستخدم يدخل كود تأكيد غير صحيح.
النتيجة المتوقعة: النظام يظهر رسالة مثل "رمز التحقق غير صحيح" ولا يتم تفعيل الحساب.



السيناريو السادس: نسيان كلمة المرور
الوصف: المستخدم يضغط "نسيت كلمة المرور" ويدخل بريده أو رقمه.
النتيجة المتوقعة: يتم إرسال كود أو رابط إعادة تعيين كلمة المرور ويكمل الخطوات.

The image shows a 'Reset Password' form with the following fields and buttons:

- Reset Password** (Title)
- Enter the verification code sent to your email and your new password.
- Verification Code:** Six empty input boxes.
- New Password:** One input box with the placeholder text 'Enter new password'.
- Reset Password** (Button)

Below the form is a snippet of a WhatsApp chat with 'Aqar And Services'. The chat shows a message from 'test121' asking for a password reset link, and a reply from 'Aqar And Services' providing the link.

This is a screenshot of the 'Reset Password' form with the verification code '961311' entered in the six boxes. The 'New Password' field is empty and masked with dots. The 'Reset Password' button is visible at the bottom.



Reset Password

Enter the verification code sent to your email and your new password.

Verification Code:

New Password:

Updating...

✔ Password successfully updated!

السيناريو السابع: محاولة تسجيل الدخول قبل تأكيد الحساب الوصف: المستخدم يسجل حساب جديد لكنه ما فعل الكود، ويحاول تسجيل الدخول. النتيجة المتوقعة: تظهر له رسالة توضح أن الحساب غير مفعّل بعد.



Aqaraty

Don't have an account?

Create an account

Forgot your password?

Recover Password

Login

Username / Email / Phone

confirmtest


Password

.....

Login

! Verification required

السيناريو الثامن: محاولة اضافة مهنة وترك بعض الحقول فارغة
الوصف: المستخدم يضيف مهنة ولكن يترك بعض الحقول فارغة
النتيجة المتوقعة: تظهر له رسالة توضح أن يجب تعبئة الحقول الفارغة .

**Add Service**

Service Description

Description is required.

Price Range

100

Address

Gaza ▼

Service Type

Select Service Type ▼

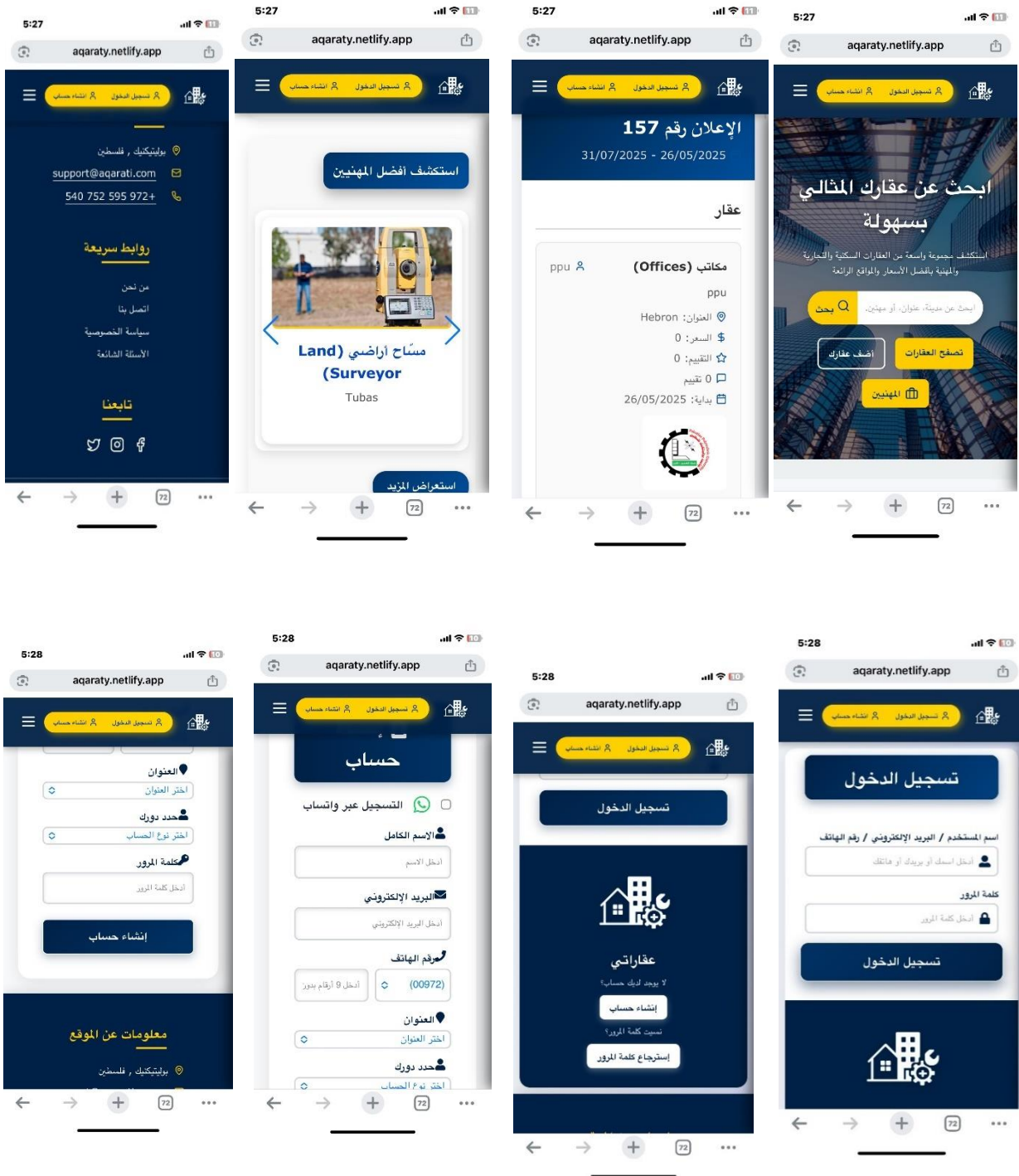
Service type is required.

Save Cancel

السيناريو التاسع: محاولة بعث رسالة الى شخص اخر ولكن تم انتهاء جلسة الدخول
الوصف: المستخدم يرسل رسالة الى شخص اخر
النتيجة المتوقعة: لن يتم ظهور الرسائل .

هناك العديد من السيناريوهات ولكن حاولنا نذكر اشهر الامثلة .

5.5.2 اختبار مدى توافق الشاشات وخاصة الهواتف (QA Responsive)





5.6 الإضافات

خلال تطوير المشروع، تمت إضافة مجموعة من التحسينات والخصائص الجديدة التي ساهمت في رفع جودة النظام وتجربة المستخدم، وأهم هذه الإضافات:

- **دعم الترجمة الكاملة للموقع:** تم ترجمة النظام بالكامل ليعمل باللغتين العربية والإنجليزية، بحيث يمكن للمستخدم التبديل بين اللغتين بسهولة في جميع الصفحات والمكونات.
- **تطوير الخرائط التفاعلية:** تم تحسين استخدام الخرائط داخل الموقع، لتكون أكثر وضوحًا وسهولة في عرض المواقع الجغرافية والبحث عنها.
- **مساعد ذكي عبر واتساب:** تمت إضافة مساعد ذكي في واتساب الخاص بمنصتنا، حيث يمكنه الرد على المستخدمين بطريقة تفاعلية من خلال خيارات جاهزة (List)، مما يسهل الوصول للمعلومة بسرعة وسلاسة.
- **رفع النظام على بيئة عمل:**
 - تم رفع الواجهة الخلفية (Back-end) على خوادم Microsoft Azure.
 - وتم نشر الواجهة الأمامية (Front-end) على منصة Netlify لتكون متاحة للمستخدمين بشكل دائم ومستقر.
- **إضافة المحادثة الفورية:** تم دمج نظام محادثة مباشرة (Live Chat) داخل الموقع، يتيح للمستخدمين التواصل السريع مع فريق الدعم أو الاستفسار عن الخدمات المتوفرة.

الفصل السادس الخاتمة والعمل المستقبلي

الحمد لله الذي أوصلنا إلى هذه المرحلة من العمل، ومنّ علينا بالصبر والعزيمة لإتمام هذا المشروع وتحقيق أهدافه بنجاح، ونسأله التوفيق في الخطوات القادمة نحو تطويره وتحسينه.

الخاتمة :

مع انتهاء هذا المشروع، تمكنا من بناء منصة إلكترونية فعّالة ومخصصة لاحتياجات السوق الفلسطيني في مجالي العقارات والخدمات المهنية. وقرّ النظام حلاً متكاملًا يجمع بين سهولة الاستخدام والوظائف المتقدمة، حيث يمكن للمستخدمين استعراض العقارات، التواصل مع المهنيين المحليين، والاعتماد على التقييمات والمراجعات لاتخاذ قرارات مدروسة. من خلال دمج ميزات مهمة مثل الترجمة الثنائية، الخرائط التفاعلية، الدردشة الفورية، والتكامل مع واتساب، تمكنا من تبسيط التفاعل بين الأطراف وزيادة فعالية التواصل، مما انعكس بشكل إيجابي على تجربة المستخدم ورفع من مستوى الثقة والمصداقية في المنصة.

العمل المستقبلي:

في المرحلة القادمة من المشروع، نطمح إلى توسيع نطاق المنصة وتحسين قدراتها بشكل أكبر. واحدة من أبرز الإضافات المخططة هي إدراج نظام تجارة إلكترونية (E-commerce) مخصص لبيع مواد البناء، بحيث يتمكن المستخدمون من شراء مستلزمات البناء مباشرة عبر المنصة، ما يوفر عليهم الوقت والجهد. كما نعمل على تطوير تطبيق موبايل متكامل يدعم جميع مزايا النظام لتوفير تجربة أكثر مرونة على الهواتف الذكية. من جانب آخر، نخطط لتوظيف تقنيات الذكاء الاصطناعي في تحليل مدى تقدم المستخدم في بناء منزله بالإضافة إلى توظيفها كعامل اضافي للموثوقية وسيتم دراستها لاحقاً، واقتراح المهنيين والخدمات أو حتى المواد المناسبة بناءً على المرحلة التي وصل إليها، مما يجعل المنصة أكثر ذكاءً وتفاعلاً مع احتياجات المستخدمين بشكل شخصي ودقيق.

References

- [1] Microsoft, "ASP.NET Core Documentation," [Online]. Available: <https://learn.microsoft.com/en-us/aspnet/core/>
- [2] Microsoft, "Entity Framework Core Documentation," [Online]. Available: <https://learn.microsoft.com/en-us/ef/core/>
- [3] Microsoft, "ASP.NET Core Identity," [Online]. Available: <https://learn.microsoft.com/en-us/aspnet/core/security/authentication/identity>
- [4] Microsoft, "JWT Authentication," [Online]. Available: <https://learn.microsoft.com/en-us/aspnet/core/security/authentication/jwt>
- [5] Microsoft Azure, "Blob Storage Documentation," [Online]. Available: <https://learn.microsoft.com/en-us/azure/storage/blobs/>
- [6] Mapster, "Mapster - Object Mapping Library," [Online]. Available: <https://github.com/MapsterMapper/Mapster>
- [7] RestSharp, "RestSharp - .NET HTTP Client Library," [Online]. Available: <https://restsharp.dev/>
- [8] Serilog, "Structured Logging for .NET," [Online]. Available: <https://serilog.net/>
- [9] Swashbuckle, "Swagger for .NET," [Online]. Available: <https://github.com/domaindrivendev/Swashbuckle.AspNetCore>
- [10] Microsoft, "SignalR Documentation," [Online]. Available: <https://learn.microsoft.com/en-us/aspnet/core/signalr/>
- [11] React, "React Official Documentation," [Online]. Available: <https://react.dev/>
- [12] React Router, "React Router Documentation," [Online]. Available: <https://reactrouter.com/>
- [13] React Hook Form, "Documentation," [Online]. Available: <https://react-hook-form.com/>
- [14] Yup, "Validation Library," [Online]. Available: <https://github.com/jquense/yup>
- [15] Formik, "Forms in React," [Online]. Available: <https://formik.org/>
- [16] Tailwind CSS, "Official Documentation," [Online]. Available: <https://tailwindcss.com/>
- [17] Framer, "Framer Motion Documentation," [Online]. Available: <https://www.framer.com/motion/>
- [18] Lucide, "Lucide Icons," [Online]. Available: <https://lucide.dev/>
- [19] React Toastify, "GitHub Repository," [Online]. Available: <https://github.com/fkhdra/react-toastify>
- [20] Recharts, "React Chart Library," [Online]. Available: <https://recharts.org/>
- [21] Swiper, "Touch Slider," [Online]. Available: <https://swiperjs.com/>

- [22] React-Leaflet, "Documentation," [Online]. Available: <https://react-leaflet.js.org/>
- [23] Leaflet Routing Machine, "GitHub Repository," [Online]. Available: <https://github.com/perliedman/leaflet-routing-machine>
- [24] i18next, "Internationalization Framework," [Online]. Available: <https://www.i18next.com/>
- [25] Emoji Picker, "Emoji Picker for React," [Online]. Available: <https://github.com/ealush/emoji-picker-react>
- [26] Day.js, "Documentation," [Online]. Available: <https://day.js.org/>
- [27] date-fns, "Date Utility Library," [Online]. Available: <https://date-fns.org/>
- [28] Axios, "GitHub Repository," [Online]. Available: <https://github.com/axios/axios>
- [29] jwt-decode, "GitHub Repository," [Online]. Available: <https://github.com/auth0/jwt-decode>
- [30] Postman, "API Testing Tool," [Online]. Available: <https://www.postman.com/>
- [31] Netlify, "Frontend Hosting Platform," [Online]. Available: <https://docs.netlify.com/>
- [33] UltraMsg, "WhatsApp API," [Online]. Available: <https://docs.ultramsg.com/>
- [34] Microsoft, "SQL Server Documentation," [Online]. Available: <https://learn.microsoft.com/en-us/sql/sql-server/>
- [35] GitHub, "GitHub Docs," [Online]. Available: <https://docs.github.com/>
- [42] Trello, "Trello Documentation," [Online]. Available: <https://support.atlassian.com/trello/>
- [44] Microsoft, "Visual Studio Code," [Online]. Available: <https://code.visualstudio.com/>
- [45] Microsoft, "Visual Studio 2022," [Online]. Available: <https://visualstudio.microsoft.com/vs/>
- [46] diagrams.net, "formerly draw.io," [Online]. Available: <https://www.diagrams.net/>
- [47] Microsoft, "Word - Office Application," [Online]. Available: <https://www.microsoft.com/en-us/microsoft-365/word>
- [48] Microsoft, "PowerPoint - Office Application," [Online]. Available: <https://www.microsoft.com/en-us/microsoft-365/powerpoint>
- [49] Google, "Google Maps," [Online]. Available: <https://maps.google.com/>
- [50] M. Fowler, "Patterns of Enterprise Application Architecture," [Online]. Available: <https://martinfowler.com/books/ea.html>
- [51] N. Dragoni et al., "Microservices: yesterday, today, and tomorrow," *Present and Ulterior Software Engineering*, vol. 100, pp. 195-216, 2017. [Online]. Available: <https://arxiv.org/pdf/1606.04036.pdf>
- [52] Microsoft, "Model-View-ViewModel (MVVM) pattern," [Online]. Available: <https://learn.microsoft.com/en-us/azure/architecture/patterns/mvvm>