



جامعة الشهيد حمه لخضر - الوادي

كلية العلوم الدقيقة - قسم الاعلام الألي

اكاديمية ليسانس

مقدم من :

كير عماد الدين

عبد الحكيم بريم

أسامة خلف

تحت إشراف : بليلة خولة

عنوان :

تصميم موقع ويب للإدارة من وكالة بيع السيارات

رئيسا:

مقررا

الشكر الأول لمشرفنا بليلة خولة

نودّ أن نعرب عن امتناننا العميق لدعمكم وتوجيهكم القيّمين في مشروعنا. لقد أحدث تفانيكم واستعدادكم للمساعدة فرقاً كبيراً، ولم نكن لننجح لولاكم. شكراً لكم على صبركم وتشجيعكم، وعلى بذلكم قصارى جهدكم دائماً لضمان نجاحنا. نحن ممتنون حقاً لكل ما قدمتموه. كما نشكركم، أعضاء لجنة الامتحانات، على حضوركم. يشرفنا حضوركم.

ملخص

في العصر الرقمي الذي نعيشه اليوم، أصبحت التجارة الإلكترونية جزءًا لا يتجزأ من حياتنا اليومية، حيث يسعى العديد من الأشخاص لإتمام معاملاتهم التجارية عبر الإنترنت بسبب سهولة الوصول، الراحة، وتوفير الوقت, يهدف هذا المشروع إلى إنشاء موقع إلكتروني متخصص في بيع السيارات، بحيث يوفر تجربة تسوق مبتكرة وآمنة للعملاء من خلال هذا الموقع.

الكلمات المفتاحية

القوة الحصانية
نظام الدفع الرباعي
بوابة الدفع
شراء سيارات

Summary

In the digital era we live in today, e-commerce has become an inseparable part of our daily lives, as many individuals seek to complete their commercial transactions online due to easy access, convenience, and time savings. This project aims to establish a specialized website for car sales, providing customers with an innovative and secure shopping experience through this platform.

Key Features :

Horse Power
Four-wheel
Car Purchase
Payment Gateway

1	المقدمة العامة
2	الفصل الاول : التحليل و الدراسة التحليلية
3	1.2 تحليل السوق و المنافسين
4	2.2 الجمهور المستهدف و احتياجات المستخدمين
8	3.2 تحليل عملية الشراء و البيع الحالية
10	3. الفصل الثاني : التصميم
11	1.3 مقدمة
11	2.3 منهجية التصميم
11	1.2.3 تعريف منهجية TUP 2
11	2.2.3 مبدأ عمل المنهجية
12	3.2.3 المراحل الاساسية TUP 2
12	3.3 لغة النمذجة الموحدة (UML)
12	1.3.3 تعريف UML
13	2.3.3 اقسام المخططات
41	4.3 البرنامج المستعمل (Draw.IO)
42	4. الفصل الثالث : التنفيذ
43	1.4 مقدمة
43	2.4 الادوات و لغات البرمجية
43	1.2.4 لغات الواجهة الامامية (Front-end)
44	2.2.4 لغات الواجهة الخلفية (Back-end)
45	3.2.4 الادوات المستعملة
45	4.2.4 التحكم في الاصدارات باستخدام Git/GitHub
46	5.2.4 الاستضافات و الخدمات السحابية
46	3.4 تهيئة بيئة التطوير و التقنيات المستخدمة
47	1.3.4 بيئة العمل الواجهة الامامية (Front-end)
48	2.3.4 بيئة العمل الواجهة الخلفية (Back-end)

4.4 تطوير الواجهة الخلفية	51
1.4.4 اعداد الخادم و قاعدة البيانات	51
2.4.4 تنفيذ عمليات CRUD	52
3.4.4 تطوير التوجيه Routes	58
4.4.4 ادارة حسابات المستخدمين و الصلاحيات	60
5.4.4 ربط الواجهة الخلفية بقاعدة البيانات	63
5.4 تطوير الواجهة الامامية	64
1.5.4 الاجزاء الرئيسية للصفحات	64
2.5.4 انشاء الصفحات الرئيسية	66
3.5.4 انشاء بوابة الدفع	71
4.5.4 انشاء Dashboard	72
5.5.4 الربط بين الواجهة الامامية و الخلفية	80
6.4 واجهات البرمجية	84
1.6.4 واجهات تسجيل الدخول/تسجيل	84
2.6.4 واجهات Dashboard	86
3.6.4 واجهة الصفحة الرئيسية	91
4.6.4 واجهات عرض السيارة	93
الخاتمة	94
المراجع و المصادر	94

فهرس الاشكال

رقم	عنوان الشكل	الصفحة
1	مخطط حالت استخدام العميل Customer	13
2	مخطط حالت استخدام المسؤول Admin	14
3	مخطط حالت استخدام وكيل الدفع Payment	16
4	مخطط حالت استخدام وكيل المبيعات Agent Sales	17
5	مخطط الفئات Classes Diagram	18
6	مخطط النشاط تسجيل المستخدمين Register User	21
7	مخطط النشاط تسجيل دخول المستخدمين Login Users	23
8	مخطط النشاط اعادة تعيين كلمة مرور Password Reset	25
9	مخطط النشاط ادارة السيارات Listings Car Manage	27
10	مخطط النشاط ادارة المستخدمين User Manage	29
11	الرد على استفسارات العمالء Inquiries Customer to Replay	31
12	مخطط النشاط انشاء الطلبات Orders Create	33
13	مخطط النشاط تسجيل الدفع Payment Register	35
14	مخطط النشاط تتبع المدفوعات Payment Track	37
15	مخطط النشاط عرض ملخص المبيعات Summary Sales View	39
16	هيكل الملفات الخاص بالواجهة الامامية Front-end	47
17	هيكل الملفات الخاص بالواجهة الخلفية Back-end	49
18	هيكل الملفات الخاص بالواجهة الخلفية Back-end	50
19	تهيئة و اعداد الخادم Server بواسطة js.Express	52
20	اظهار كافة السيارات في صفحة عرض السيارات	54
21	اظهار عملية حذف السيارة من قاعدة البيانات	55
22	اظهار السيارة المختارة	56
23	عملية تحديث بيانات السيارة	57
24	واجهة التوجيه الرئيسية في الواجهة الخلفية	59
25	واجهة التوجيه للاستفسارات في الواجهة الخلفية	59
26	واجهة توضح صالحة Admin	60
27	واجهة توضح صالحة Payment	61
28	واجهة توضح صالحة Sales	62
29	مخطط لل Models في ملف prisma.schema	64
30	كود البرمجي للواجهة الرئيسية Page Home	66

31	كود البرمج ي للواجهة Us About	67
32	كود البرمجي للواجهة انشاء حساب	68
33	كود البرمجي للواجهة Login	69
34	كود البرمجي للواجهة Contact	74
35	كود البرمجي بوابة الدفع	71
36	كود البرمجي للواجهة Dashboard	72
37	كود البرمج ي للواجهة الدارة السيارات	73
38	كود البرمجي للواجهة Client Dashboard	74
39	كود البرمجي للواجهة Client Purchase	75
40	كود البرمجي للواجهة Client Dashboard	76
41	كود البرمجي للواجهة Payments Dashboard	76
42	كود البرمجي للواجهة Sales Dashboard	77
43	كود البرمجي للواجهة inquiries Dashboard	78
44	كود البرمجي للواجهة Sale Dashboard	78
45	كود البرمجي للواجهة Sales Dashboard	79
46	معمارية ملفات الربط بين واجهة الامامية و الخلفية	81
47	معمارية ملفات الربط بين واجهة الامامية و الخلفية	82
48	الكود البرمجي الخاص بعملية الربط بين end-Front و end-Back	83
49	الكود البرمجي الخاص بعملية الربط بين end-Front و end-Back	83
50	الواجهة البرمجية لعملية انشاء حساب في الموقع	84
51	الواجهة البرمجية لعملية تسجيل حساب في الموقع	84
52	الواجهة البرمجية لعملية استرجاع كلمة المرور	85
53	الواجهة البرمجية لعملية اعادة تعيين كلمة المرور	85
54	الواجهة البرمجية للوحة التحكم Dashboard	86
55	الواجهة البرمجية لتعديل و اضافة سيارة	86
56	الواجهة البرمجية لعرض السيارات المضافة	87
57	الواجهة البرمجية لعرض المستخدمين	87
58	الواجهة البرمجية للوحة التحكم خاصة بالزبون	88
59	الواجهة البرمجية للوحة التحكم خاصة بالزبون	88
60	الواجهة البرمجية لبوابة الدفع صفحة رئيسية	89
61	الواجهة البرمجية لبوابة الدفع المدفوعات	89
62	الواجهة البرمجية للوحة التحكم ال خاصة بـ Manger Sales	90
63	الواجهة البرمجية لقائمة المبيعات الخاصة بـ Manger Sales	90
64	الواجهة البرمجية لعرض السيارة من قائمة المبيعات	91

فهرس المحتويات

65	الواجهة البرمجية الستعراض السيارات في الصفحة الرئيسية للموقع	91
66	الواجهة البرمجية 'من نحن' الظهار فريق العمل في موقع Drivona	92
67	الواجهة البرمجية للتواصل مع ال طاقم العمل في موقع Drivona	92
68	الواجهة البرمجية جميع تفاصيل السيارة قبل عملية الشراء	93

المقدمة العامة

- قطاع بيع السيارات يُعد حجر الزاوية في الاقتصاد العالمي، وقد شهد تطوراً ملحوظاً من الطرق التقليدية التي تعتمد على المعارض الفيزيائية والتفاعل المباشر. في السابق، كان شراء سيارة يتطلب زيارة صالات العرض والتفاوض وجهاً لوجه. لكن مع ظهور الإنترنت والتقدم في الذكاء الاصطناعي، بدأ هذا القطاع يتجه نحو الأتمتة والرقمنة. هذا التحول لم يغير فقط كيفية تفاعل العملاء مع السيارات، بل أتاح فرصاً غير مسبقة للوصول إلى قاعدة عملاء أوسع وتقديم خدمات بيع وشراء أكثر كفاءة ومرونة، مما يدفع الصناعة نحو آفاق جديدة.
- رغم هذه التطورات، يواجه سوق بيع السيارات تحديات كبيرة، خصوصاً في ظل المنافسة المتزايدة وتوقعات المستهلكين المتغيرة. فالطرق التقليدية غالباً ما تكون مكلفة في التشغيل ومحدودة الانتشار، بينما تفتقر الحلول الرقمية أحياناً إلى اللمسة الشخصية التي يفضلها بعض المشترين. يمثل إنشاء موقع إلكتروني فعال لبيع السيارات تحدياً مزدوجاً: كيف يمكن لهذا الموقع أن يوفر تجربة سلسة وجذابة للعملاء عبر الإنترنت، وفي نفس الوقت يحافظ على ثقة العميل ويقدم له كل المعلومات التي يحتاجها لاتخاذ قرار الشراء، متغلباً على العقبات اللوجستية والفنية؟
- من هنا، تهدف هذه المذكرة إلى استكشاف سبل تطوير وبناء موقع إلكتروني متكامل لبيع السيارات، مع التركيز على دمج أحدث التقنيات لتحسين تجربة المستخدم وزيادة فعالية المبيعات. لقد قمنا بتحليل شامل لاحتياجات السوق الحالية، وتحديد الميزات الأساسية التي يجب أن يتضمنها الموقع ليكون جذاباً وتنافسياً. تم إنجاز تصميم هيكلي للموقع، ووضع خطة عمل مفصلة لمراحله الإنشائية، بالإضافة إلى تحديد التقنيات اللازمة لضمان سهولة الاستخدام والأمان. هذا العمل يهدف إلى تقديم حل رقمي مبتكر يلبي توقعات البائعين والمشترين على حد سواء.
- تتألف هذه المذكرة من ثلاثة فصول أساسية تشرح الجوانب المختلفة لعملية إنشاء الموقع الإلكتروني لبيع السيارات. يبدأ الفصل الأول مبني على دراسة تحليلية جيدة الخاص بالأسواق واكتشاف نقاط القوة والضعف لدى المنافسين ومحاولة تطبيقها وتطويرها، ثم يتناول الجزء الثاني يتعلق بالتصميم ومحاكاة الواقع بالمخططات المنجزة. أما الجزء الثالث فهو عملية التنفيذ يهدف الى تطبيق الفصل الاول والثاني لبناء الموقع متكامل ويوفق بين تلبية حاجة السوق وكسب ثقة المستخدمين والزبائن.

الفصل الأول: التحليل و الدراسة التحليلية

1.2 تحليل السوق والمنافسين

1.1.2 نظرة عامة على السوق الحالي :

يشهد سوق تجارة السيارات تحولاً رقمياً سريعاً، مع تزايد الاهتمام بالمنصات الإلكترونية لشراء وبيع المركبات. يفضل المستخدمون بشكل متزايد الطرق الرقمية نظراً للراحة والشفافية وكفاءة الوقت التي تقدمها. بينما تتوسع المنصات العالمية، لا تزال العديد من الأسواق المحلية تعتمد بشكل كبير على الطرق التقليدية، مما يوفر فرصة قوية للتحويل الرقمي.

2.1.2 تحليل المنافسين المحليين والعالميين :

• Cars.com

• Autotrader

• Carvana

تشمل المنافسين المحليين منصات مثل:

• Haraj

• OpenSooq

تختلف هذه المنصات في تصميم واجهة المستخدم، والخدمات المقدمة، ومنطقة التغطية، والثقة التي تولدها بين المستخدمين.

3.1.2 نقاط القوة والضعف لدى المنافسين :

- مخزون كبير وانتشار واسع.
- تطبيقات جوال سهلة الاستخدام.
- ميزات بحث وتصفية متقدمة.

نقاط الضعف:

- تحقق محدود من البائعين.
- تجربة مستخدم غير متسقة
- نقص في خدمات فحص المركبات أو حماية المشتري.

2.2 الجمهور المستهدف واحتياجات المستخدمين :

1.2.2 تحديد المستخدمين المستهدفين (البائعين والمشتريين) :

الأفراد الذين يبيعون مركباتهم الشخصية، وكلاء السيارات، وشركات التأجير.

1. الأفراد الخاصون - مالكون يبيعون مركباتهم المستعملة الشخصية.

2. وكلاء السيارات - وكلاء معتمدون يقدمون سيارات جديدة أو معتمدة مسبقاً.

3. شركات التأجير - أساطيل تباع مركبات الإيجار المستعملة.

4. شركات التأجير التمويلي - مركبات منتهية الإيجار يتم إعادة بيعها.

5. بيوت المزادات - شركات تسعى لتصفية المخزون.

المشترون: العملاء الأفراد الباحثين عن سيارات جديدة أو مستعملة، والشركات التي تشتري مركبات للأسطول، وبائعو السيارات.

1. المستهلكون الأفراد - الأشخاص الباحثون عن مركبات للاستخدام الشخصي.

2. الشركات ومديرو الأساطيل - الشركات التي تشتري مركبات لأساطيل الشركات.

3. بائعو السيارات والمصدرون - الوكلاء أو الوسطاء الذين يشترون المخزون لإعادة البيع.

4. الهواة والمجمعون - المشترون الباحثون عن مركبات نادرة أو خاصة.

5. المؤسسات المالية - البنوك أو المقرضون الذين يستعيدون/يبيعون المركبات.

2.2.2 تحليل سلوك المستخدم وتوقعاته :

- بحث سريع ودقيق - توقع نتائج ذات صلة مع فلاتر متقدمة (الموديل، السنة، نطاق السعر، الموقع).
- قوائم سيارات مفصلة - صور عالية الجودة، مواصفات كاملة، سجل الصيانة، وتقارير حالة المركبة.
- تسعير شفاف - تسعير نهائي واضح بدون رسوم خفية، مع أدوات مقارنة
- اتصال موثوق - رسائل آمنة داخل المنصة، دعم التفاوض، وجدولة الفحص.
- تحديثات في الوقت الحقيقي - إشعارات فورية للقوائم الجديدة أو تغييرات الأسعار.

3.2.2 الاحتياجات الأساسية والمشاكل التي يعاني منها المستخدم :

. الاحتياجات الأساسية:

1. واجهة سهلة الاستخدام وبديهية: تنقل بسيط للمشتري والبائع.
2. معلومات موثوقة عن البائعين والمركبات: ملفات تعريف موثقة، تقييمات، وتاريخ شفاف للبائع.
3. أدوات مقارنة سيارات متقدمة: مقارنة جانبية للمواصفات والأسعار.
4. بحث وفلاتر مخصصة: تصفية دقيقة (محرك، السعر، الموقع، إلخ).
5. عملية دفع آمنة: بوابات دفع مؤمنة وحماية من الاحتيال.

. المشاكل :

1. إعلانات مزيفة أو مضللة –عمليات احتيال، أو أوصاف غير دقيقة، أو صور معدلة.
2. عدم وجود تحقق من المركبات –لا توجد تقارير فنية أو تاريخية (حوادث، إصلاحات).
3. سوء التواصل وعدم استجابة البائعين –تأخير في الردود أو تعامل غير احترافي.
4. طرق دفع غير آمنة –خطر الاحتيال مع التحويلات المباشرة أو عدم وجود حماية عبر وسيط.
5. إعلانات بأسعار مبالغ فيها أو غير قابلة للتفاوض –تسعير غير واقعي دون مرونة.

الحلول المقترحة:

- ✓ تحقق صارم من البائعين –التحقق من الهوية وتسجيل النشاط التجاري.
- ✓ فحص للمركبات من طرف ثالث –فحص من قبل ميكانيكي معتمد وتقارير تاريخية.
- ✓ نظام محادثة ومكالمات داخل التطبيق –تواصل مشفر مع تتبع لزمان الاستجابة.
- ✓ حماية الدفع عبر وسيط –يتم الإفراج عن الأموال فقط بعد تأكيد المشتري

4.2.2 المشكلات الشائعة في المنصات الحالية :

1 مشاكل الجودة والمصادقية:

- غياب المعايير : تفاوت في جودة الإعلانات ومعلومات غير دقيقة عن المركبات.
- عدم وجود معايير تحقق واضحة : لا توجد إرشادات واضحة لوصف حالة السيارة.
- إعلانات مضللة: اختلافات بين المواصفات المعلنة والحالة الفعلية للمركبة

2 التحديات التقنية:

- واجهة مستخدم سيئة: تنقل معقد وواجهات غير بديهية.
- أنظمة بحث قديمة: نتائج غير متعلقة بطلبات المستخدم.
- مشاكل في الأداء: بطء في التحميل وانقطاعات متكررة في الخدمة.

3 مخاوف أمنية:

- مخاطر الاحتيال: انتشار عمليات الاحتيال والممارسات المضللة.
- ثغرات في حماية البيانات: ضعف في حماية المعلومات الشخصية للمستخدمين.
- غياب الرقابة: عدم مراجعة كافية للإعلانات المشبوهة.

4 مشاكل تجربة المستخدم:

- دعم عملاء ضعيف: بطء في الرد على الاستفسارات والشكاوى.
- خدمات غير مترابطة: لا يوجد تكامل بين مراحل ما قبل البيع وما بعده.
- ميزات محدودة: غياب أدوات ضرورية مثل حاسبات التمويل.

5 مشكلات إدارة المحتوى:

- إعلانات مكررة: ظهور نفس السيارة في عدة إعلانات.
- تنظيم ضعيف: فلاتر وخيارات فرز غير فعالة.
- إعلانات منتهية: بقاء الإعلانات القديمة بعد بيع المركبة.

6 نقص الثقة:

- أنظمة تقييم متحيزة: تقييمات ومراجعات يمكن التلاعب بها.
- غياب حماية للمشتري: لا توجد ضمانات أو سياسات تأمين.
- شروط غير واضحة: سياسات المنصة ومسؤولياتها غير مفهومة.

. الحلول المقترحة:

- ✓ تطبيق متطلبات موحدة للإعلانات
- ✓ تطوير نظام تقييم موضوعي للبائعين
- ✓ فرض فحوصات معتمدة للمركبات
- ✓ إنشاء مراكز دعم عملاء تعمل على مدار الساعة.

3.2 تحليل عملية الشراء والبيع الحالية :

1.3.2 وصف طرق تجارة المركبات التقليدية :

1. المعاملات عبر الوكالات :

- يزور المشترون الوكالات فعليًا لفحص المركبات، التفاوض على الأسعار، واستكمال الأوراق (مثل العقود، التمويل).

- **المشاكل :** ضغوط مندوبي المبيعات، رؤية محدودة للمخزون، ومفاوضات تستغرق وقتًا طويلاً.

2. المبيعات الخاصة:

- يستخدم البائعون الإعلانات المبوبة (الصحف، لوحات الإعلانات) أو التواصل الشفهي للعثور على المشترين.

- **المشاكل :** نطاق وصول محدود، غياب معايير التسعير، ومخاوف أمنية أثناء اللقاءات.

3. دور المزادات:

- يقوم المشترون بالمزايدة شخصيًا، غالبًا دون معلومات كافية عن المركبة (مثل سجل الصيانة، حالة السيارة).

- **المشاكل :** خطر دفع مبلغ كبير لمركبات تالفة، ولا توجد اختبارات قيادة.

4. مبيعات بمساعدة وسطاء:

- يسهل الوسطاء الخارجيون الصفقات مقابل رسوم، مما يزيد التكلفة والتعقيد.

- **المشاكل :** عمولات عالية، وغياب الشفافية في الرسوم.

2.3.2 تحديات العمليات اليدوية أو غير المتصلة بالإنترنت :

1. القيود الجغرافية:
 - يقتصر المشترون/البائعون على الأسواق المحلية، مما يؤدي إلى تفويت صفقات أفضل في أماكن أخرى.
2. استهلاك الوقت
 - تتطلب العملية عدة لقاءات شخصية (اختبارات القيادة، التفاوض، الأوراق)، مما يؤخر إتمام الصفقات.
3. عدم تكافؤ المعلومات:
 - قد يخفي البائعون العيوب، ولا يستطيع المشترون الوصول إلى تاريخ المركبة (مثل الحوادث، التلاعب بعدد الكيلومترات).
4. مخاطر الاحتيال:
 - الدفع نقدًا غير آمن، وتحويل الملكية قد يكون قابلاً للتزوير.
5. تسعير غير فعال:
 - لا توجد بيانات سوقية فورية، مما يؤدي إلى تسعير مبالغ فيه أو أقل من السوق.

3.3.2 مزايا الانتقال إلى نظام رقمي :

1. سوق يعمل على مدار الساعة:
 - يمكن للمشتريين والبائعين إجراء المعاملات في أي وقت، مما يعزز الراحة والسيولة في السوق.
2. شفافية السوق:
 - توفر بيانات التسعير الفورية مثل التكامل مع صفقات عادلة.
3. سير عمل سلس:
 - العقود الرقمية مثل الدفع الإلكتروني يقلل من الأعمال الورقية.
4. تعزيز الثقة:
 - تقييمات موثوقة و معلومات مؤكدة على سيارة المختارة.

الفصل الثاني: التصميم

1.3 مقدمة :

في الفصل السابق ، قمنا بدراسة الوضع الحالي و تحليل المشاكل كما اقترحنا الحلول المناسبة لتتقل من الاساليب التقليدية الى الاساليب الالكترونية.

اما في هذا الفصل فان الهدف الأساسي هو نمذجة مشروع لمحاكاة الواقع من خلال انشاء تصميم.

و لضمان ذلك ، من الضروري اتباع منهجية تصميم واضحة. ومن بين الاساليب التي تتيح ذلك ، قمنا باختيار التصميم الموجه نحو الكائنات (Object-Oriented Design) نظرا لقدرته العالية على تمثيل الواقع بشكل منطقي و ثابت ، و ذلك من خلال لغة النمذجة UML.

2.3 منهجية التصميم (2TUP (Two-Track Unified Process :

1.2.3 تعريف منهجية 2 TUP

تعد 2TUP منهجية حديثة لتطوير البرمجيات مقتبسة من المنهجية المشهورة RUP(s Unified Rational) لكنها تضيف اليها بعدا عمليا جديدا في عملية التطوير.

2.2.3 مبدأ عمل المنهجية :

ترتكز منهجية TUP على تطوير تكراري و ترايدي من خلال مسارين متوازيين :

أ. المسار الوظيفي (Functional Path) :

يهتم بجمع وتحليل المتطلبات، كما يركز على المستخدم النهائي واحتياجاته ويستخدم أدوات مثل: حالات الاستخدام (Use Cases) ، نماذج سلوك النظام وواجهات الاستخدام (XU/II). والهدف هو تحديد ما يجب أن يفعله النظام.

ب. المسار التقني (Technical Path) :

يهتم بالتصميم الداخلي والمعماري للنظام. ويتعامل مع الجوانب التالية: اختيار التكنولوجيا، الأمن المعلوماتي، الأداء والبنية التحتية (مصاربة النظام). والهدف هو تحديد كيف سيتم تنفيذ ما طلب في المسار الوظيفي.

3.2.3 المراحل الاساسية TUP 2 :

تشبه إلى حد كبير مراحل RUP لكنها تُطبق على كل المسارين بشكل متزامن:

1. **البداية (Inception) :** تحديد نطاق المشروع، الأهداف الأساسية، والمتطلبات الأولية. ويتم فيها استخدام Use Classes Diagram, Activity Diagram, Use Case Diagram
2. **التفصيل (Elaboration) :** تحليل متعمق للمتطلبات، تحديد المخاطر، وضع التصميم المعماري الأولي. ويتم فيها استخدام Class Diagram , Activity Diagram .
3. **البناء (Construction) :** تطوير البرمجية تدريجياً حسب الأولويات.
4. **الانتقال (Transition) :** اختبار، توثيق، نشر وتسليم النظام للمستخدم النهائي

3.3 لغة النمذجة UML :

1.3.3 تعريف UML :

لغة النمذجة الموحدة هي لغة نمذجة بصرية معيارية تُستخدم لتصميم وتوثيق الأنظمة البرمجية عبر مخططات هيكلية وسلوكية مثل مخطط الفئات والتسلسل. تساعد في توثيق المتطلبات، تحسين التواصل بين الفرق، وتخطيط هيكل النظام قبل التنفيذ.

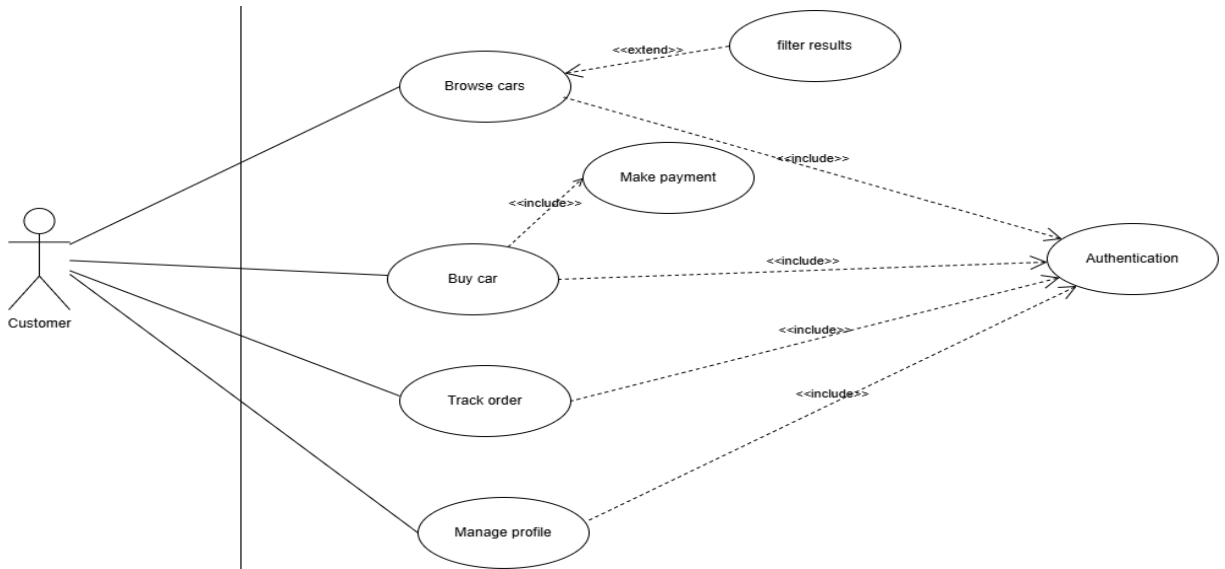
2.3.3 اقسام المخططات :

● مخطط حالات الاستخدام Use Case :

◦ ما هو مخطط Use Case ؟

هو وصف منهجي يحدد تفاعلاً محدداً بين "المستخدم" (Actor) والنظام لتحقيق هدف وظيفي، ويشمل الخطوات الأساسية والاستثنائية التي توضح سلوك النظام من منظور المستخدم.

1. مخطط حالات استخدام العميل (Customer) :



الشكل 1 : مخطط حالات استخدام العميل (Customer)

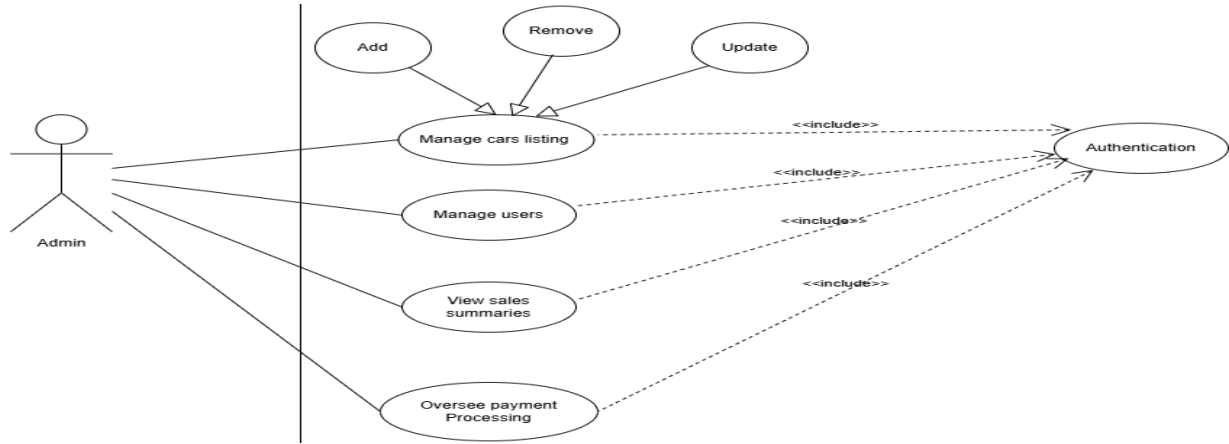
العقد المتواجدة في المخطط :

- العنوان: مخطط حالات استخدام العميل (Customer)
- الهدف: يُظهر الإجراءات التي يمكن للعميل تنفيذها ضمن نظام بيع السيارات.
- المعني : العميل (Customer)

وصف المخطط :

1. تصفح السيارات: (Browse cars)
 - البحث عن السيارات المتاحة وعرضها.
2. إجراء الدفع: (Make payment)
 - إتمام عملية الدفع لشراء سيارة.
3. شراء سيارة: (Buy car)
 - تنفيذ عملية الشراء الكاملة للسيارة (الاختيار، الدفع، التأكد).
4. تتبع الطلب: (Track order)
 - التحقق من حالة الطلب (مثل: قيد المعالجة، قيد التوصيل).
5. إدارة الملف الشخصي: (Manage profile)
 - تحديث البيانات الشخصية (العنوان، طرق الدفع، وغيرها).

2. مخطط حالات استخدام المسؤول (Admin Use Case Diagram) :



الشكل 2 : مخطط حالات استخدام المسؤول (Admin Use Case Diagram)

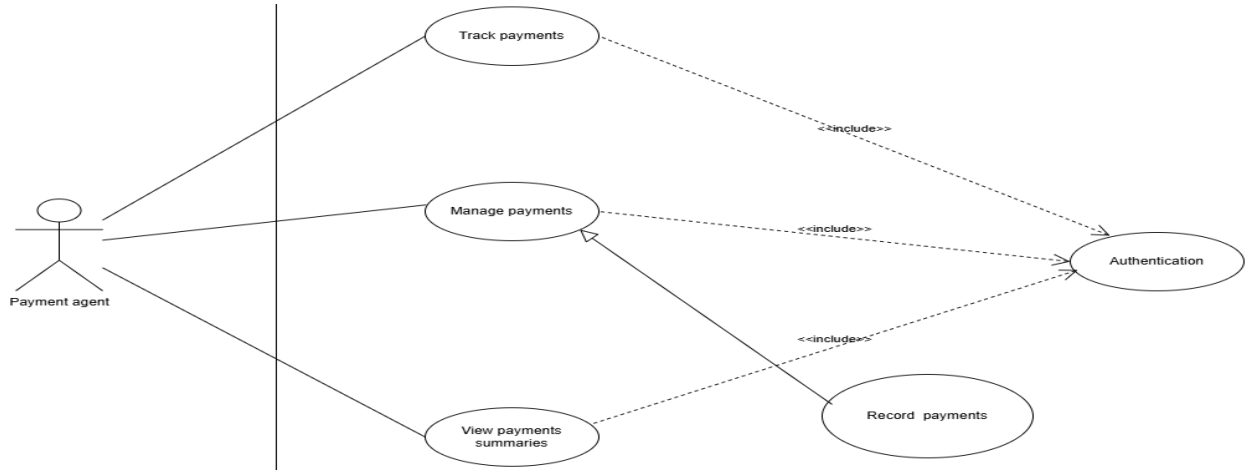
ملخص التعريف :

- العنوان: مخطط حالات استخدام المسؤول (Admin)
- الوصف: يُظهر الإجراءات التي يمكن للمسؤول تنفيذها ضمن نظام بيع السيارات.
- المعني: المسؤول (Admin)

وصف المخطط :

- إدارة إعلانات السيارات:
 - إضافة (Add): إدراج سيارات جديدة في الإعلانات.
 - حذف (Remove): إزالة سيارات من قائمة الإعلانات.
 - تحديث (Update): تعديل تفاصيل السيارات الموجودة.
- إدارة المستخدمين: إضافة، حذف، أو تعديل حسابات المستخدمين والأدوار الخاصة بهم.
- عرض ملخصات المبيعات: التحقق من جميع بيانات المبيعات على المنصة.

3 . مخطط حالات استخدام وكيل الدفع (Payment Use Case Diagram) :



الشكل 3 : مخطط حالات استخدام وكيل الدفع (Payment Use Case Diagram)

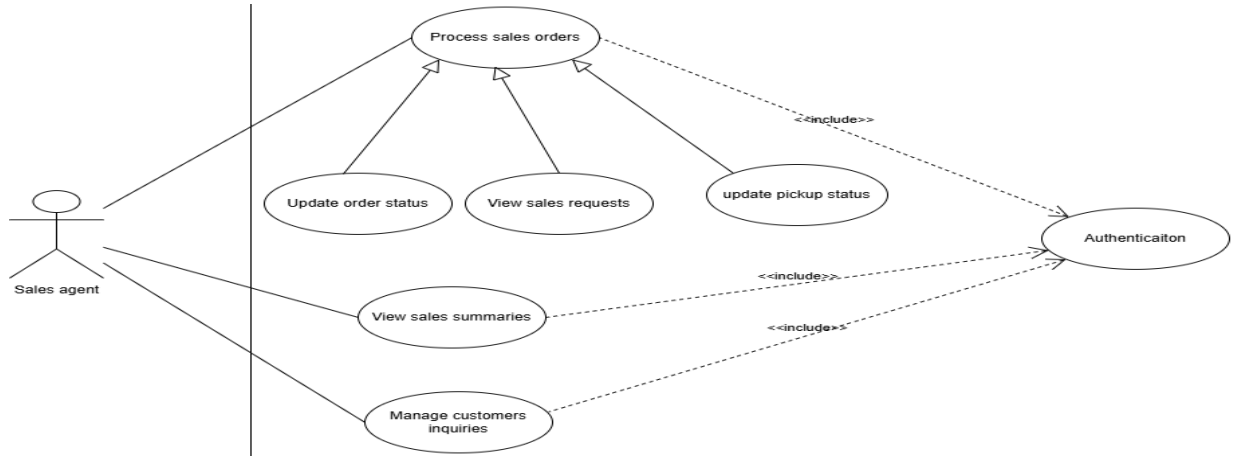
العقد المتواجدة في المخطط :

- العنوان: مخطط حالات استخدام وكيل الدفع (Payment Use Case)
- الوصف: يُظهر الإجراءات اللازمة لدفع من طرف العميل.
- المعني: عميل الدفع (Payment Agent)

وصف المخطط :

- تتبع المدفوعات :مراقبة تقدم المدفوعات أو تاريخها.
- إدارة المدفوعات :التعامل مع مهام مختلفة مثل تعديل أو التحقق من المدفوعات.
- تسجيل المدفوعات :إدخال تفاصيل الدفع في النظام.
- عرض ملخصات الدفع :رؤية تقارير أو ملخص لجميع المعاملات المالية.
- الوظائف العامة (مشاركة بين جميع الحالات): المصادقة (Authentication) التأكد من أن الوكيل مسجل الدخول ومخول قبل تنفيذ أي إجراء.

4. مخطط حالات استخدام وكيل المبيعات (Sales Agent Use Case Diagram) :



الشكل 4 : مخطط حالات استخدام وكيل المبيعات (Sales Agent Use Cas Diagram):

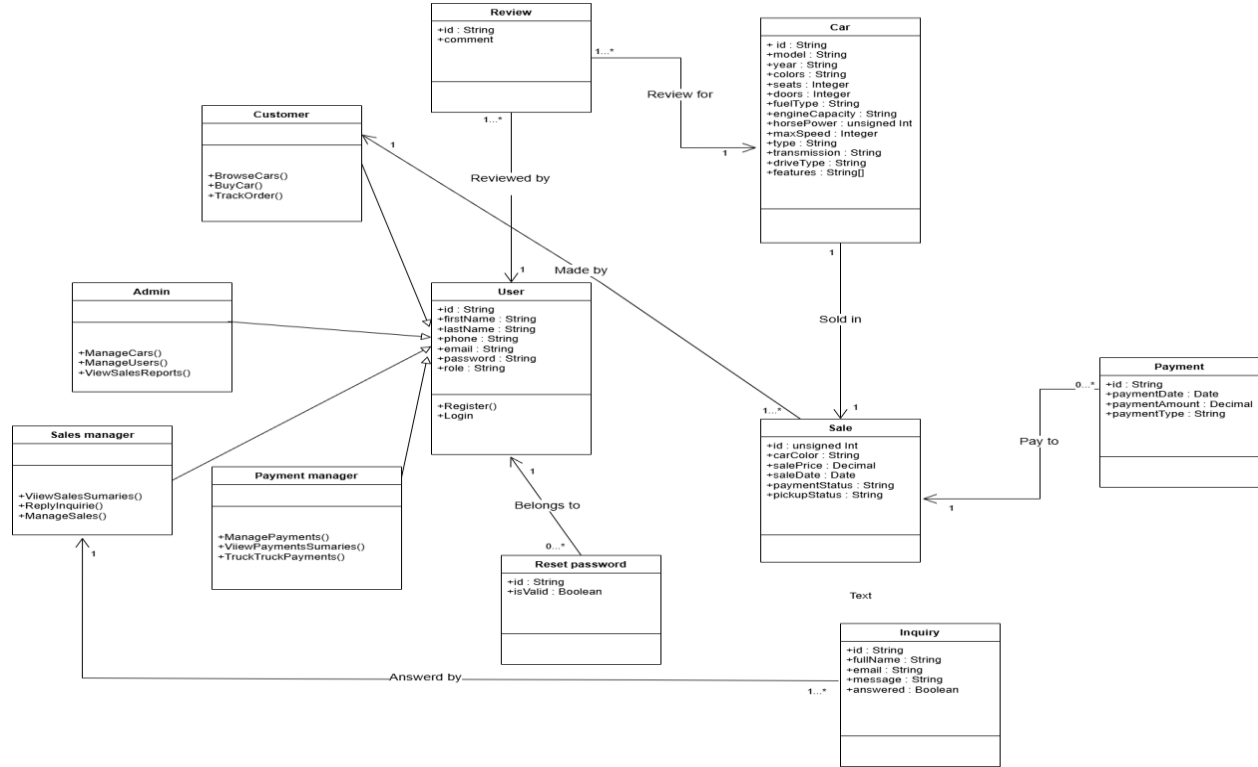
العقد المتواجدة في المخطط :

- العنوان: مخطط حالات استخدام وكيل المبيعات (Sales Agent Use Case Diagram)
- الوصف: يُظهر الإجراءات التي بها وكيل المبيعات .
- المعني : وكيل المبيعات (Sales Agent).

وصف المخطط :

- معالجة طلبات البيع:
 - تحديث حالة الطلب :تحديد ما إذا تم شحن أو توصيل الطلب.
 - عرض طلبات البيع :التحقق من طلبات العملاء.
 - تحديث حالة الاستلام :تسجيل بيانات الاستلام أو التسليم.
 - عرض ملخصات المبيعات :رؤية إجمالي المبيعات وإحصائيات الأداء.
 - إدارة استفسارات العملاء :الرد على أسئلة العملاء أو متابعتها.
- الوظائف العامة:
- المصادقة : (Authentication) مطلوبة قبل الوصول إلى أي وظيفة في النظام.

• مخطط الفئات : Class Diagram



الشكل 5 : مخطط الفئات الخاص Class Diagram

الفئات المتواجدة في المخطط :

- User : مستخدم
- Car : سيارة
- Sales : مبيعات
- Payment : مدفوعات
- Inquiry : استفسارات
- Review : تقييمات
- Reset Password : استعادة كلمة مرور

وصف للمخطط :

● User : المستخدم

- Id معرف اساسي
- FirstName الاسم الاول
- last Name الاسم الاخير
- Phone رقم الهاتف
- Email البريد الالكتروني
- Password كلمة المرور
- Role دور المستخدم في النظام

● Car : سيارة

- Id معرف اساسي
- Model اسم موديل السيارة
- Year سنة الصنع
- Colors الالوان المتوفرة
- Seats عدد المقاعد
- Doors عدد الابواب
- maxSped السرعة القصوى
- Type نوع السيارة
- Transmission نوع ناقل الحركة
- drive Type نوع الدفع (امامي, خلفي, رباعي)
- features المزايا المتوفرة للسيارة
- Price سعر السيارة
- Discount التخفيض
- Availability حالة التوفر
- fuelType نوع وقود المحرك
- engineCapacity سعة المحرك
- Horsepower القوة الحصانية

● Sales مبيعات

- Id معرف اساسي
- Car Color لون السيارة المختارة من طرف الزبون
- Sale Price السعر الاجمالي
- Sale date تاريخ الطلب
- Payment Status حالة الدفع
- Pickup Status حالة الاستلام

• Payment : مدفوعات

- Id معرف اساسي
- Payment date تاريخ عملية الدفع
- PaymentAmount المبلغ المدفوع
- PaymentType نوع الدفعة

• Inquiry : استفسارات

- Id معرف اساسي
- fullName الاسم الكامل للمرسل
- Email البريد الالكتروني
- Message محتوى الاستفسار
- Answered حالة الاجابة

• Review : تقييمات

- Id معرف اساسي
- Comment تعليق العميل
-

• Reset Password : اعادة كلمة المرور

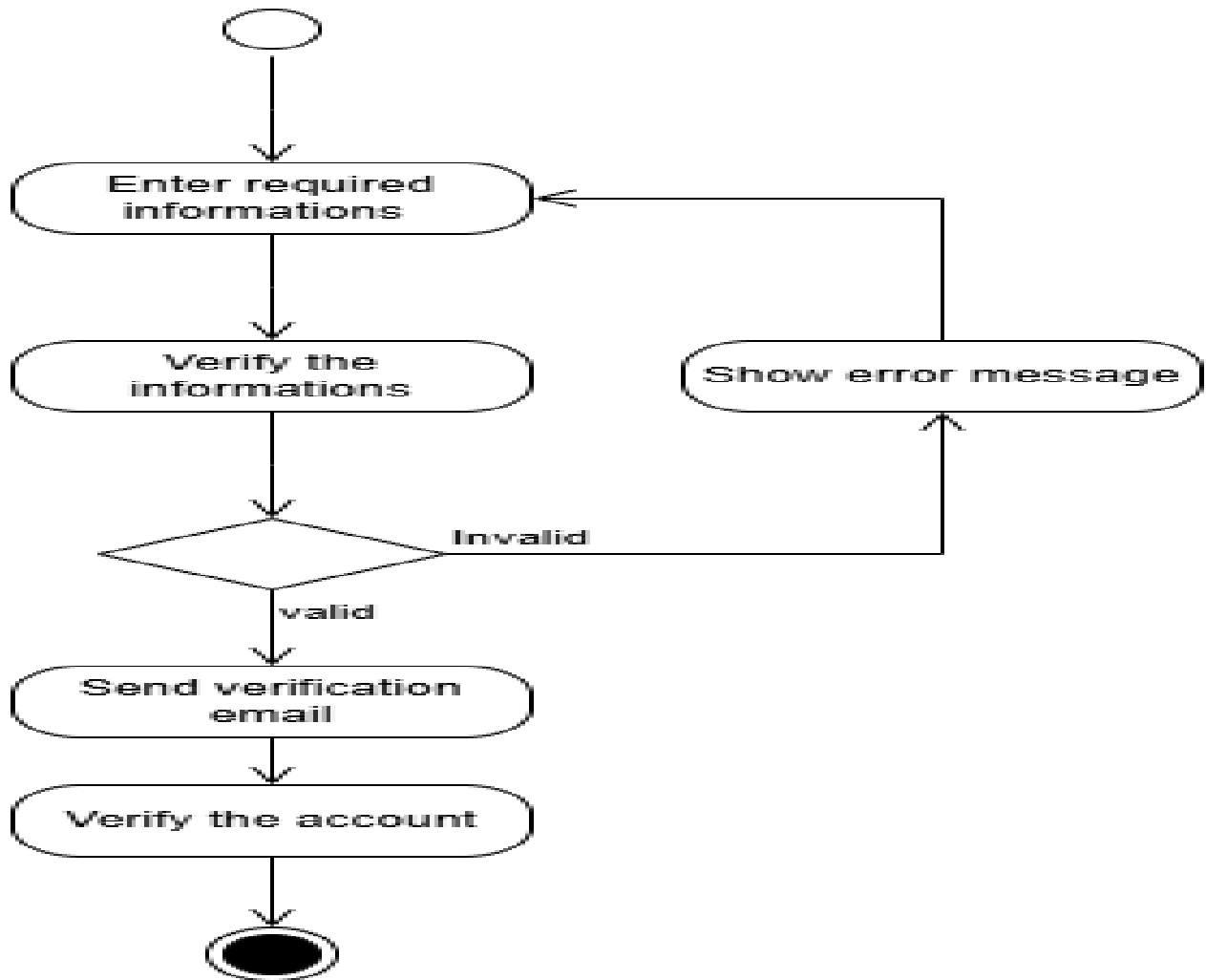
- Id معرف اساسي
- is Valid حالة عملية اعادة تعيين كلمة المرور

• مخطط النشاط Activity :

◦ ما هو مخطط النشاط (Activity Diagram) ؟

هو نوع من مخططات UML يُستخدم لتمثيل تدفق العمل أو سير العمليات داخل النظام، ويُظهر تسلسل الأنشطة، التفرعات، التكرارات، والقرارات من البداية حتى النهاية بطريقة مرئية.

1. تسجيل المستخدمين (Users Register) :



الشكل 6 : مخطط النشاط تسجيل المستخدمين (Users Register)

العقد المتواجدة في المخطط :

- ادخال المعلومات المطلوبة (Enter required information's)
- التحقق من المعلومات (Verify the information's)
- عرض رسالة خطأ (في حالة المعلومات غير صحيحة). (Show error message)
- إرسال بريد التحقق (في حالة المعلومات صحيحة). (Send verification email)
- التحقق من الحساب. (Verify the account)

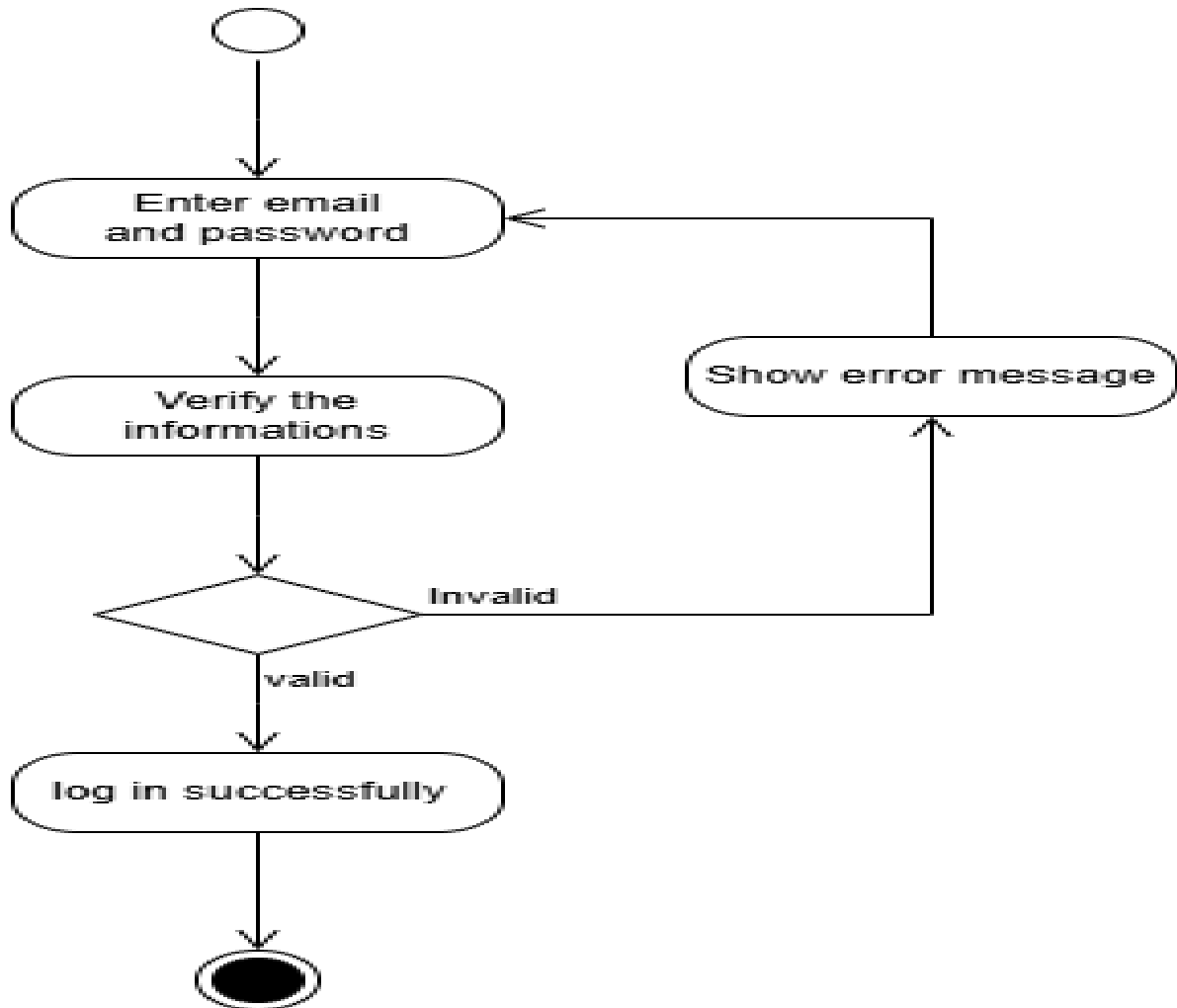
وصف المخطط :

1. "إدخال المعلومات المطلوبة" (نموذج يحتوي على عدة حقول).
2. يقوم النظام بـ:
 - التحقق من تنسيق الحقول (من جهة العميل).
 - التحقق من التكرار (من جهة الخادم).
3. نقطة اتخاذ القرار:
 - إذا كانت البيانات غير صحيحة:
 - عرض أخطاء لكل حقل.
 - إعادة المستخدم لصفحة الإدخال.
 - إذا كانت البيانات صحيحة:
 - إنشاء سجل مستخدم بانتظار التحقق.
 - إرسال بريد إلكتروني للتأكيد.
 - يتطلب تأكيد البريد الإلكتروني ("تأكيد الحساب").

اعتبارات الأمان:

- تخزين كلمات المرور.
- رمز التحقق بصلاحية لمدة 24 ساعة.
- محتوى البريد الإلكتروني يجب أن يتجنب الكلمات المحفزة للرسائل المزعجة

2. تسجيل الدخول (Users Login) :



الشكل 7 : مخطط النشاط تسجيل دخول المستخدمين (Users Login)

العقد المتواجدة في المخطط :

- (Enter email and password) أدخل البريد الإلكتروني وكلمة المرور
- (Verify the information's) تحقق من المعلومات
- (Show error message) عرض رسالة خطأ
- (Log in successfully) تسجيل الدخول بنجاح

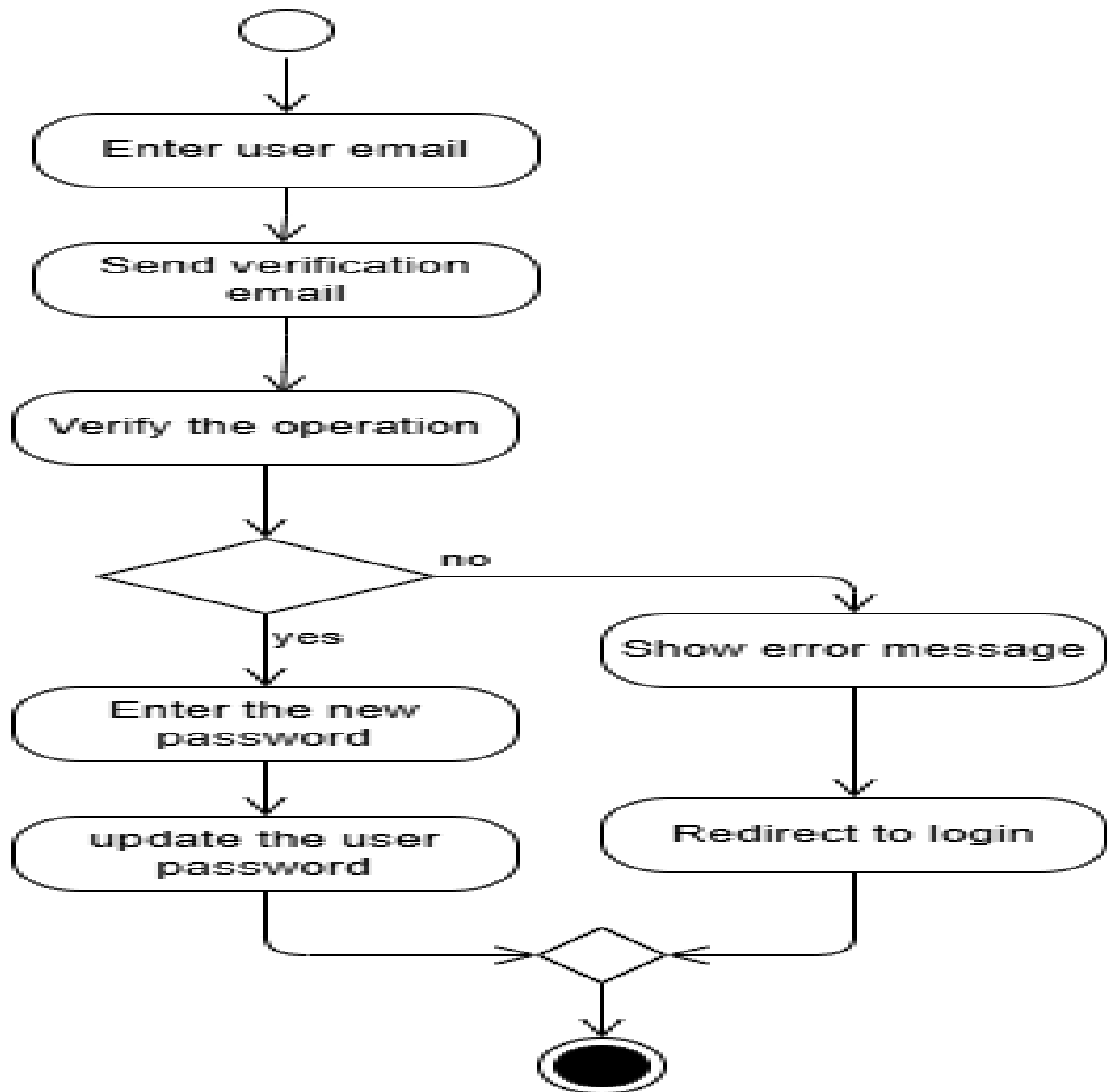
وصف المخطط :

1. "إدخال البريد الإلكتروني وكلمة المرور."
2. "التحقق من المعلومات" (نظام).
3. عقدة قرار:
 - المسار الخاطئ (غير صالح):
 - "عرض رسالة خطأ" → نهاية التدفق.
 - المسار الصحيح (صالح):
 - "تسجيل الدخول بنجاح" → نهاية النشاط.

قواعد العمل

- التحقق من صحة تنسيق البريد وكلمة المرور.
 - رسائل الخطأ يجب أن تكون عامة ("بيانات الاعتماد غير صحيحة").
 - يجب إنشاء رمز جلسة عند تسجيل الدخول الناجح.
- معايير التحقق:
- نمط البريد الإلكتروني: $^{\wedge}[\wedge s@]^+@[\wedge s@]^+.[\wedge s@]^+ \$$
- شروط كلمة المرور: 8 أحرف على الأقل، حرف كبير واحد، ورقم واحد

3. إعادة تعيين كلمة المرور (Reset Password) :



الشكل 8 : مخطط النشاط إعادة تعيين كلمة مرور (Reset Password)

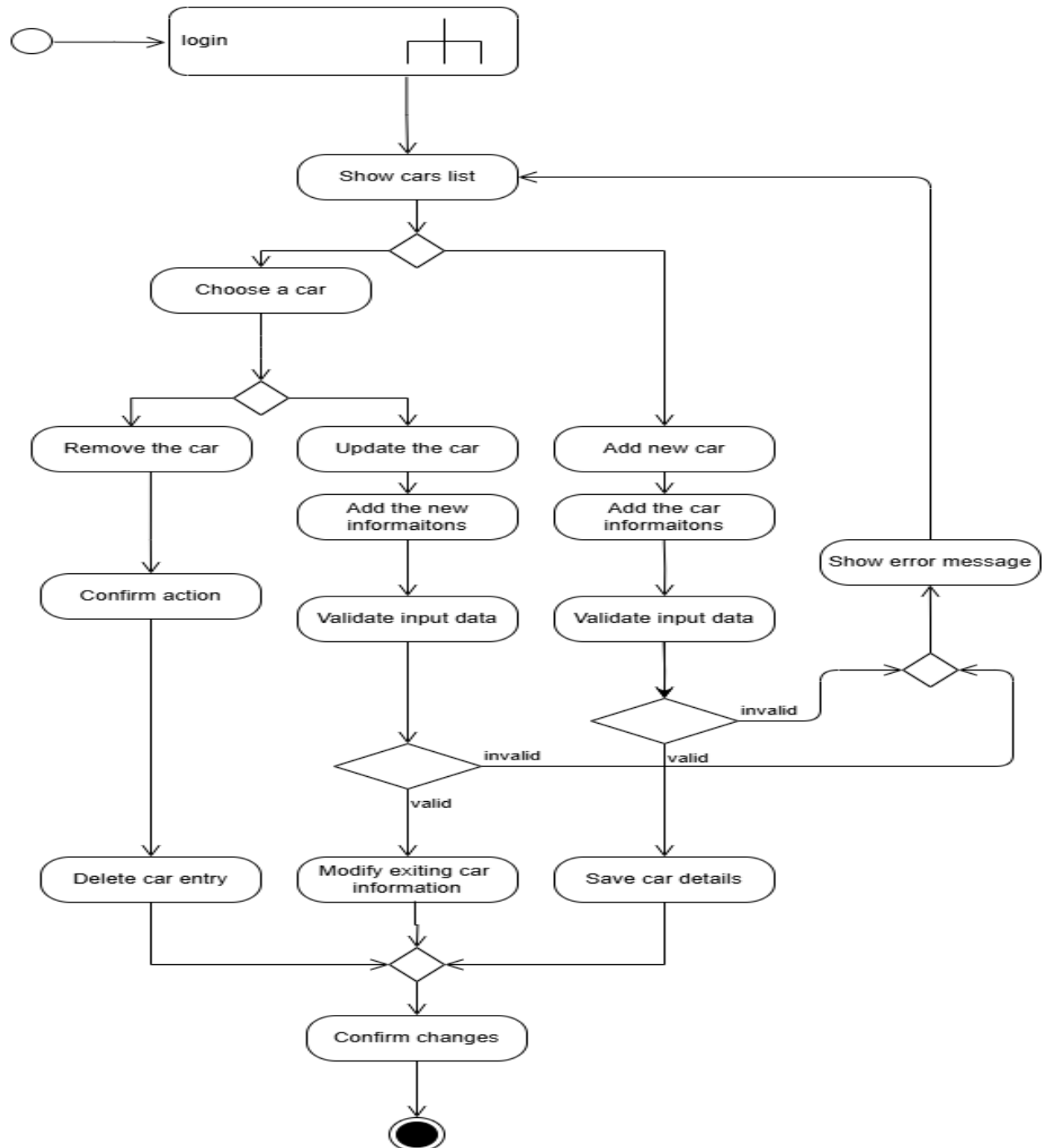
العقد المتواجدة

- Enter user email : أدخل البريد الإلكتروني للمستخدم
- Send verification email : إرسال بريد التحقق
- Verify the operation : التحقق من العملية
- Show error message : عرض رسالة خطأ
- Enter the new password : أدخل كلمة المرور الجديدة
- Update the user password : تحديث كلمة مرور المستخدم
- Redirect to login : إعادة التوجيه إلى تسجيل الدخول

وصف المخطط :

1. إدخال البريد الإلكتروني → النظام يتحقق من صحته.
 2. إرسال رابط إعادة التعيين (صلاحية مؤقتة، مثلاً 15 دقيقة).
 3. التحقق من الرمز:
 - إذا كان صالحاً → الانتقال لإعادة التعيين.
 - إذا كان غير صالح/منتهي → عرض رسالة خطأ.
 4. إدخال كلمة مرور جديدة (تراعي قواعد التعقيد).
 5. التحديث والأمان:
 - النظام يُشَقِّر كلمة المرور، يسجل التغيير، ويُنتهي الجلسات النشطة.
 6. إعادة التوجيه إلى تسجيل الدخول.
- مميزات رئيسية:
- أمان الرموز.
 - قواعد كلمة المرور.
 - إنهاء الجلسات.

4. إدارة السيارات (Manage Car Listings) :



الشكل 9 : مخطط النشاط ادارة السيارات Manage Car Listings

العقد المتواجدة في المخطط :

- **Login** : تسجيل الدخول
- **Show cars list** : عرض قائمة السيارات
- **Choose a car** : اختيار السيارة
- **Remove the car** : ازالة السيارة
- **Update the car** : تحديث السيارة
- **Add new car** : اضافة سيارة جديدة
- **Add the new information's** : اضافة المعلومات الجديدة
- **Add the car information's** : اضافة معلومات السيارة
- **Confirm action** : تأكيد الاجراء
- **Validate input data** : التحقق من صحة البيانات المدخلة
- **Show error message** : عرض رسالة الخطأ
- **Delete car entry** : حذف سجل السيارة
- **Modify existing car information** : تعديل معلومات السيارة الحالية
- **Save car details** : حفظ تفاصيل السيارة
- **Confirm changes** : تأكيد التغييرات

وصف المخطط

أ. حذف سيارة:

1. "اختيار السيارة": عرض تمهيدي للحذف.
2. "حذف السيارة": التحقق من الطلبات النشطة.
3. "تأكيد الحذف": تسجيل في سجل التدقيق.
4. "تعيين السيارة كغير نشطة".

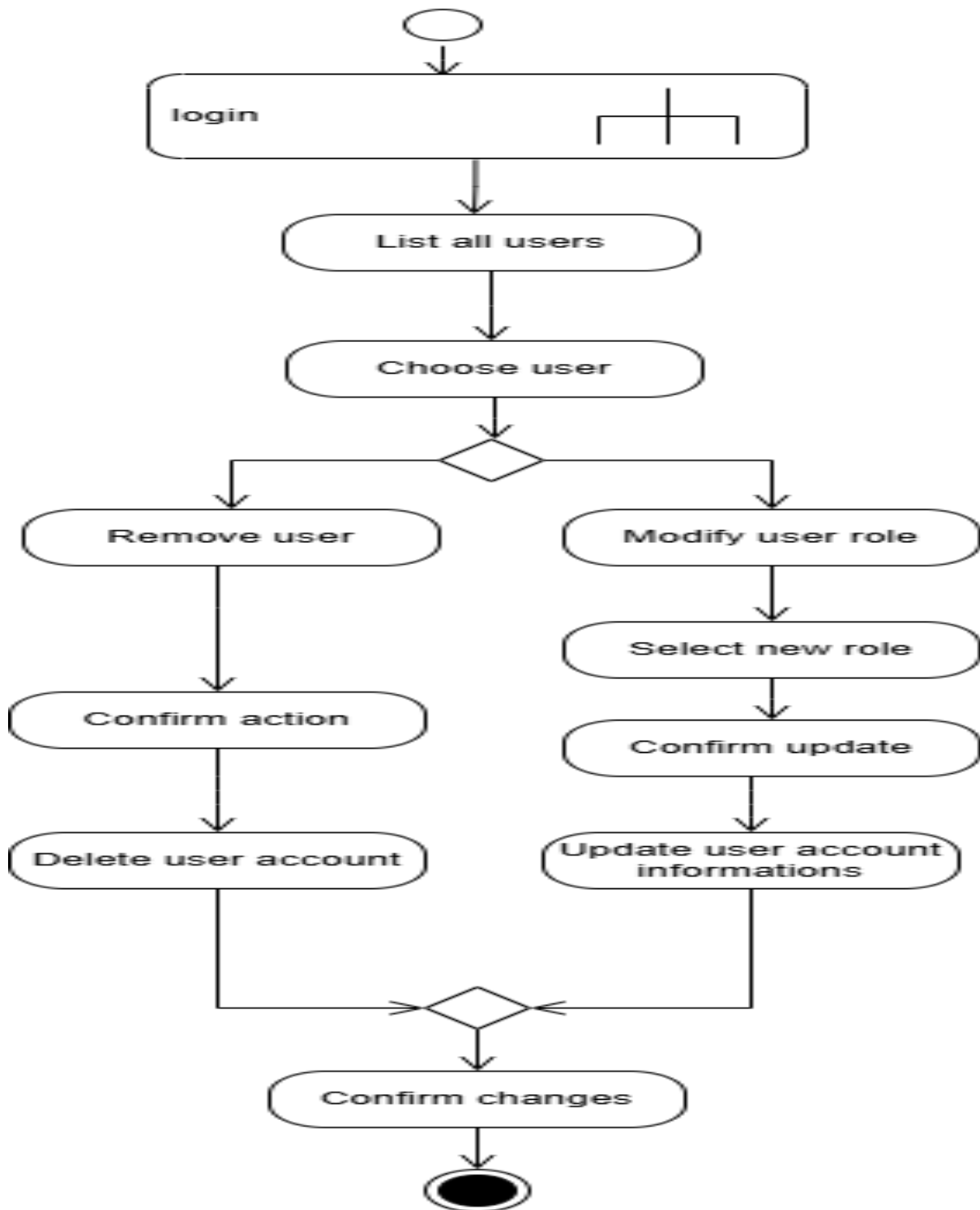
ب. تعديل سيارة:

1. "تحديث السيارة": تحميل البيانات الحالية.
2. "إضافة المعلومات الجديدة": التحقق من النموذج.
3. "التحقق من البيانات" (السعر، المواصفات...).
4. "حفظ التعديلات": حفظ نسخة في السجل.

ج. إضافة سيارة:

5. "إضافة سيارة جديدة": نموذج فارغ.
6. "إدخال بيانات السيارة": ملء الحقول المطلوبة.
7. دورة التحقق (كما في التحديث).
8. "تأكيد التعديلات".

5. إدارة المستخدمين (Manage Users) :



الشكل 10 : مخطط النشاط ادارة المستخدمين (Manage User)

العقد المتواجدة في المخطط :

- **login** : تسجيل الدخول
- **List all users** : عرض جميع المستخدمين
- **Choose user** : اختيار مستخدم
- **Remove user** : إزالة مستخدم
- **Confirm action** : تأكيد الإجراء
- **Delete user account** : حذف حساب المستخدم
- **Modify user role** : تعديل دور المستخدم
- **Select new role** : اختيار دور جديد
- **Confirm update** : تأكيد التحديث
- **Update user account information's** : تحديث معلومات حساب المستخدم
- **Confirm changes** : تأكيد التغييرات

وصف المخطط :

إجراءات إدارة الصلاحيات:

1. "عرض جميع المستخدمين" مع:

- ترقيم الصفحات.
- تصفية حسب الدور.
- البحث.

2. "اختيار مستخدم" → عرض الصلاحيات الحالية.

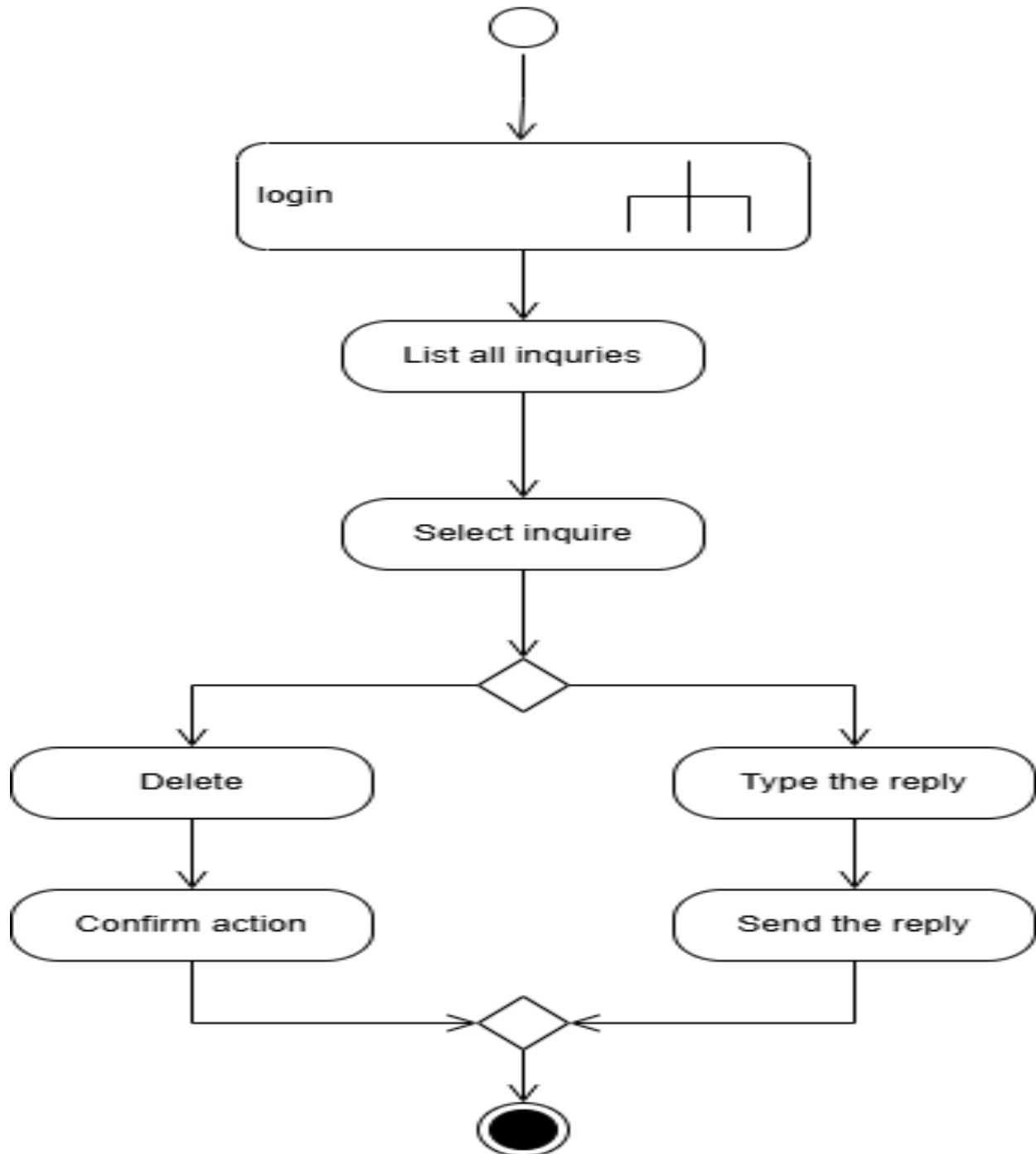
3. تعديل الدور:

- "اختيار دور جديد" مع عرض الصلاحيات.
- "تأكيد التغيير" → التحقق من الترفيع غير المصرح به.
- "تحديث معلومات المستخدم" → سجل تدقيق.

بروتوكولات الأمان:

- تطبيق تسلسل الأدوار (لا يمكن الترفيع لدور أعلى).
- إنهاء الجلسات عند تخفيض الدور.
- موافقة شخصين عند تغيير دور إداري.

6. الرد على استفسارات العملاء (Replay to Customer Inquiries) :



الشكل 11 : الرد على استفسارات العملاء Replay to Customer Inquiries

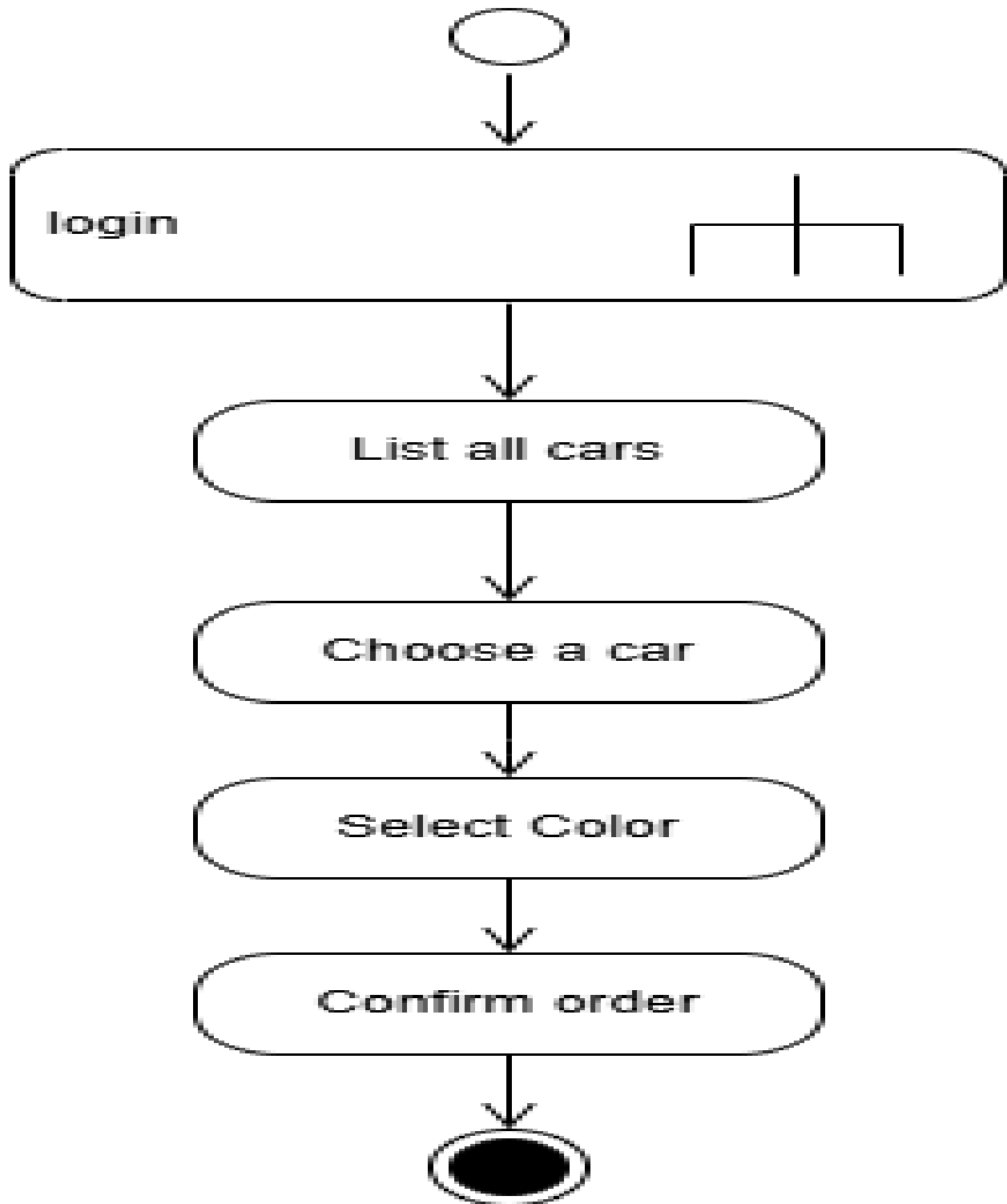
العقد المتواجدة في المخطط :

- Login : تسجيل الدخول
- List all inquiries : سرد جميع الاستفسارات
- Select inquire : اختيار الاستفسار
- Delete : حذف
- Type the reply : كتابة الرد
- Confirm action : تأكيد الإجراء
- Send the reply : إرسال الرد

وصف المخطط :

1. "عرض جميع الاستفسارات" حسب:
 - الأولوية (المدفوعة أولاً).
 - الأقدمية.
 - الفئة (مبيعات، دعم...).
2. "اختيار استفسار" → تحميل سجل المحادثة.
3. مسارات المعالجة:
 - الحذف:
 - تصفية البريد المزعج.
 - الامتثال للـ GDPR.
 - الرد:
 - "كتابة الرد" مع:
 - دعم القوالب.
 - دعم المرفقات.
 - "إرسال الرد".
 - إشعار بالبريد.
 - تحديث حالة التذكرة.

5. إنشاء الطلبات (Create Orders) :



الشكل 12 : مخطط النشاط انشاء الطلبات (Create Orders)

العقد المتواجدة في المخطط :

- Login : تسجيل الدخول
- List all inquiries : سرد جميع الاستفسارات
- Select inquire : اختيار الاستفسار
- Delete : حذف
- Type the reply : كتابة الرد
- Confirm action : تأكيد الإجراء
- Send the reply : إرسال الرد

وصف المخطط

1. التحقق من الدخول (تسجيل الدخول).

2. جلب المخزون:

- "عرض جميع السيارات" مع التوفر اللحظي.
- دعم الترقيم.

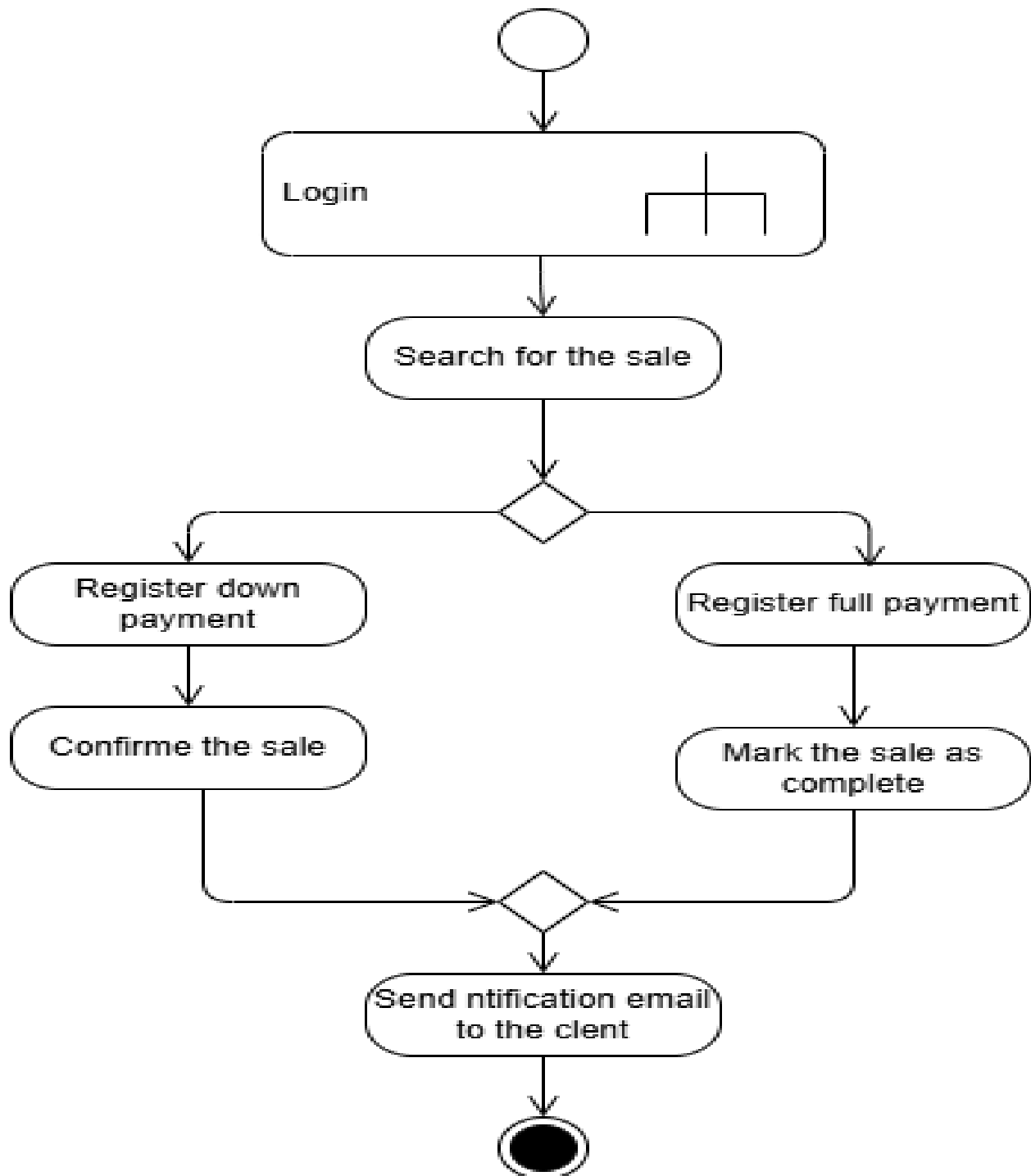
3. الاختيار:

- "اختيار سيارة" → التحقق من الكمية.
- "اختيار اللون" → تحقق من الألوان المتاحة للطراز.

4. إنهاء الطلب:

- "تأكيد الطلب" → إنشاء معاملة معقدة.

6. تسجيل الدفع (Register Payment) :



الشكل 13 : مخطط النشاط تسجيل الدفع (Register Payment)

العقد المتواجدة في المخطط :

- **Login** : تسجيل الدخول.
- **Search for the sale** : البحث عن عملية البيع.
- **Register down payment** : تسجيل الدفعة المقدمة.
- **Register full payment** : تسجيل الدفعة الكاملة.
- **Confirm the sale** : تأكيد عملية البيع.
- **Mark the sale as complete** : الاعلام بالبيع المكتمل.
- **Send notification email to the client** : إرسال بريد إلكتروني إشعاري للعميل.

وصف المخطط :

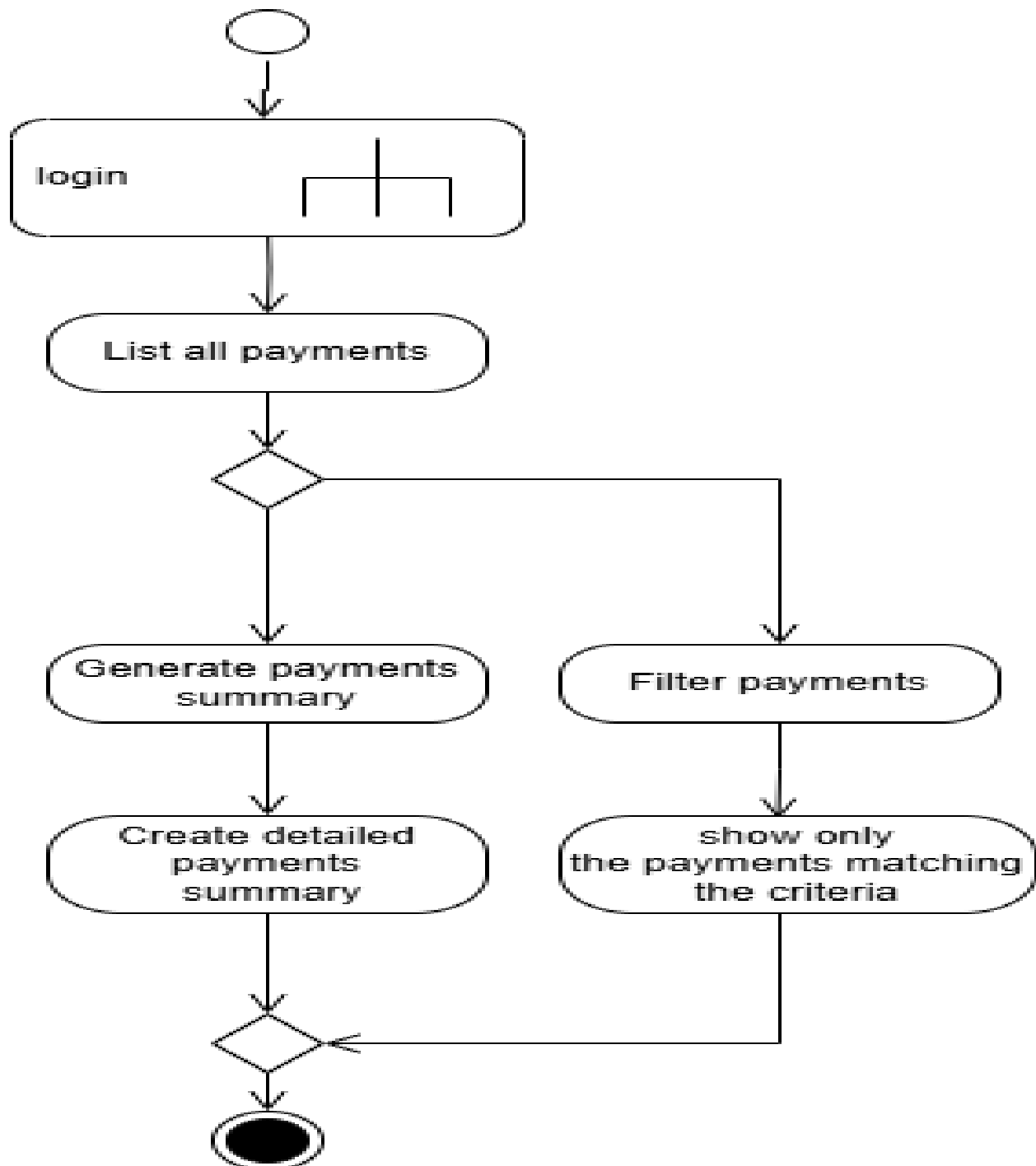
1. "البحث عن عملية البيع" عبر:

- رقم الطلب.
- بيانات العميل.
- نطاق التاريخ.

2. تسجيل الدفع:

- دفعة أولى:
- الحد الأدنى 20% من السعر.
- حجز السيارة لـ 7 أيام.
- دفع كامل:
- إنهاء المعاملة.
- بدء الشحن.

7. تتبع المدفوعات (Track Payment) :



الشكل 14 : مخطط النشاط تتبع المدفوعات (Track Payment)

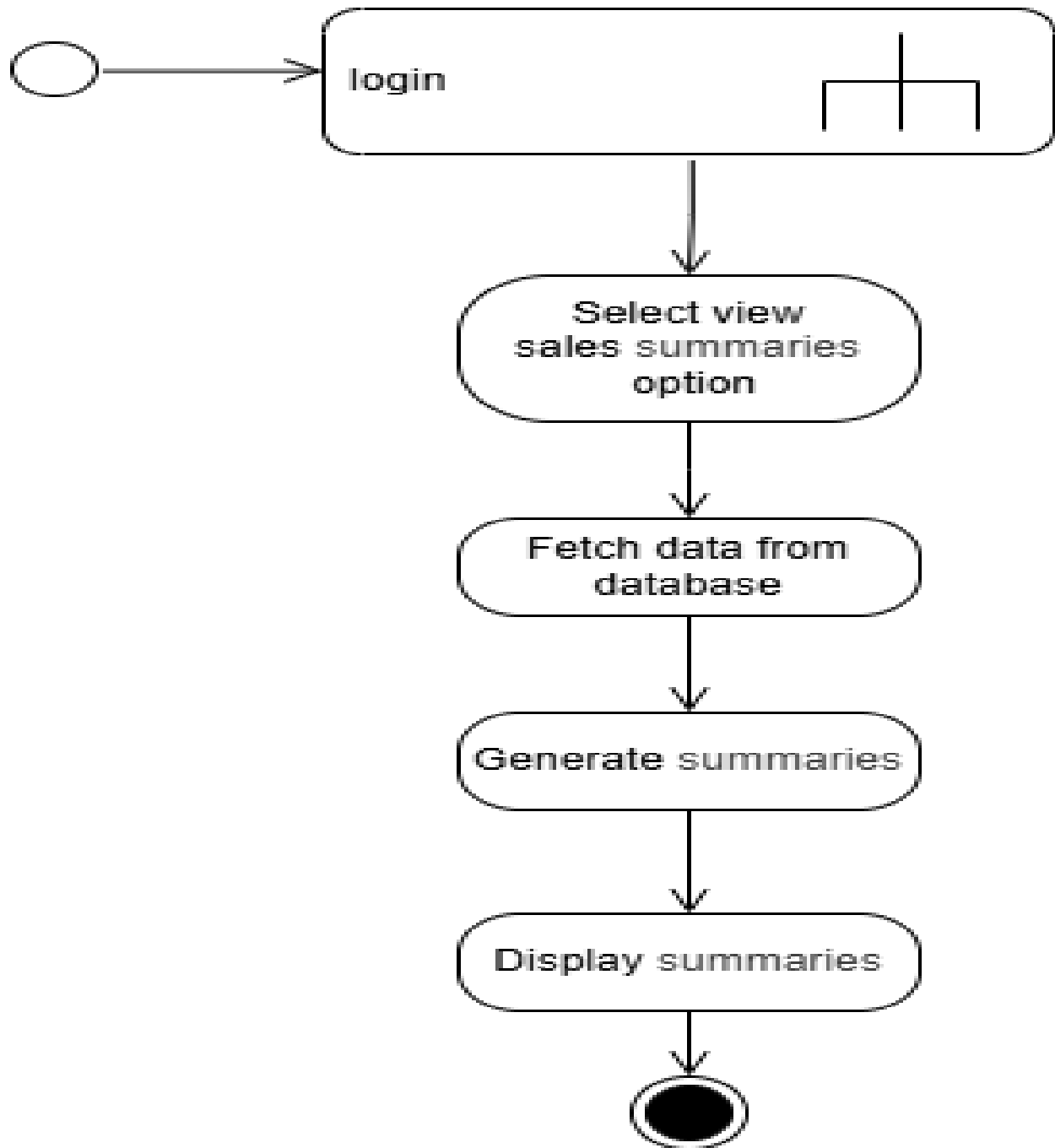
العقد المتواجدة في المخطط :

- Login: تسجيل الدخول.
- List all payments : عرض جميع المدفوعات.
- Generate payments summary : إنشاء ملخص المدفوعات.
- Filter payments : تصفية المدفوعات.
- Create detailed payments summary : إنشاء ملخص مفصل للمدفوعات.
- Show only the payments matching the criteria : عرض فقط المدفوعات التي تطابق المعايير.

وصف المخطط

1. "عرض جميع المدفوعات:"
 - آخر 30 يوم.
 - العمليات الناجحة.
 - حسب طريقة الدفع.
 2. "توليد ملخص المدفوعات" → للمديرين:
 - الإيرادات حسب الفترة.
 - توزيع طرق الدفع.
 - إحصائيات الاسترداد.
 3. "إنشاء ملخص مفصل" → للمحاسبين:
 - تفاصيل المعاملة.
 - حسابات الضرائب.
 - حالة المطابقة المحاسبية.
 4. "تصفية المدفوعات:"
 - حسب التواريخ.
 - حسب القيمة.
 - حسب الحالة (معلقة، مكتملة، فاشلة).
- هندسة البيانات:
- قاعدة بيانات عمودية للتحليلات.
 - خط معالجة ETL للتحديث الليلي.
 - صلاحيات وصول حسب الدور.

10. عرض ملخص المبيعات (View Sales Summary) :



الشكل 15 : مخطط النشاط عرض ملخص المبيعات (View Sales Summary)

العقد المتواجدة في المخطط :

- Login : تسجيل الدخول
- Select view sales summaries option : اختيار خيار عرض ملخصات المبيعات
- Fetch data from database : جلب البيانات من قاعدة البيانات
- Generate summaries : إنشاء الملخصات
- Display summaries : عرض الملخصات

وصف المخطط

1. التحقق من الهوية والصلاحيات.
2. "اختيار عرض ملخصات المبيعات" يشمل:
 - تحديد الفترة.
 - التصفية حسب المنطقة.
 - تقسيم حسب الطراز.
3. معالجة البيانات:
 - "جلب البيانات من قاعدة البيانات".
 - "توليد الملخصات" (تجميع، حسابات).
4. العرض:
 - بيانات جدولية.
 - رسوم بيانية (أعمدة، خطوط، دائرية).

الأداء:

- إستراتيجية التخزين المؤقت للاستعلامات المتكررة.
- التوليد غير المتزامن للبيانات الكبيرة.
- فهرسة قواعد البيانات على حقول التاريخ والمبيعات.

4.3 البرنامج المستعمل Draw.IO :

هو أداة مجانية لإنشاء المخططات والمخططات الانسيابية، يدعم إنشاء جميع أنواع مخططات UML بطريقة بصرية وسهلة، مع إمكانية العمل عبر المتصفح أو كتطبيق مستقل.

• مميزاته في إنشاء مخططات UML :

- يدعم كافة أنواع مخططات UML (Class, Activity, Use Case, إلخ).
- واجهة سهلة بالسحب والإفلات مع مكتبة أشكال جاهزة.
- يعمل عبر المتصفح أو كتطبيق سطح مكتب (Windows/macOS/Linux).
- يتيح التعاون في الوقت الفعلي ومشاركة الملفات.
- يدعم التصدير بعدة صيغ (PNG, PDF, SVG).
- يوفر قوالب مخصصة لتسريع إنشاء المخططات.
- يعمل دون اتصال عند استخدامه كتطبيق.
- مناسب للفرق وللأفراد مع إمكانية التحكم بالإصدارات.
- أداة مثالية لتوثيق المتطلبات وتصميم الأنظمة.

الفصل الثالث : التنفيذ

1.4 مقدمة :

يمثل جزء التنفيذ مرحلة تحويل الأفكار والتصاميم النظرية إلى واقع ملموس في مشروع الموقع الإلكتروني لبيع السيارات . هنا سنبدأ كتابة الأكواد البرمجية للواجهة الأمامية والخلفية، مختارين لغات البرمجة المناسبة ومطابقين أفضل الممارسات لضمان الكفاءة والأمان. الهدف هو بناء منصة رقمية قوية ومتكاملة توفر تجربة بيع وشراء سيارات سلسلة وفعالة عبر الإنترنت.

2.4 الادوات و لغات البرمجية :

1.2.4 لغات الواجهة الأمامية (Front-end) :

- **HTML** : هي لغة توصيف تُستخدم لبناء هيكل صفحات الويب وتحديد عناصرها مثل العناوين والفقرات والروابط والصور والنماذج.

- **CSS** : هي لغة تنسيق تُستخدم لتحديد الشكل والمظهر الخاص بصفحات الويب، حيث تتحكم في الألوان، الخطوط، التخطيطات، والتصميم البصري لعناصر HTML.

- **JavaScript** : هي لغة برمجة ديناميكية تُستخدم لإضافة التفاعلية والسلوك الديناميكي إلى صفحات الويب وتشغيل الأكواد على المتصفح.

- **TypeScript** : هي لغة برمجة تعتمد على JavaScript وتضيف لها نظام أنواع (Types) ثابت لتقليل الأخطاء وتحسين تطوير التطبيقات الكبيرة.



2.2.4 لغات الواجهة الخلفية (Back-end) :

- **JavaScript** : هي لغة برمجة ديناميكية تُستخدم لإضافة التفاعلية والسلوك الديناميكي إلى صفحات الويب وتشغيل الأكواد على المتصفح.
- **NodeJS** : هو بيئة تشغيل مفتوحة المصدر تُمكن من تشغيل JavaScript على الخادم (server) لبناء تطبيقات الويب الخلفية والتعامل مع الملفات وقواعد البيانات.
- **Express.js** : هو إطار عمل خفيف وسريع لـ Node.js يُستخدم لبناء تطبيقات الويب وواجهات برمجة التطبيقات (APIs) بسهولة وكفاءة.

ex



3.2.4 الادوات المستعملة :

- **Visual Studio Code** : هو محرر شيفرة مصدريه مجاني وخفيف من تطوير Microsoft ، يدعم العديد من لغات البرمجة ويوفر ميزات ذكية مثل الإكمال التلقائي، التصحيح، والإضافات.



- **PostMan** : هو أداة مجانية تُستخدم لاختبار واجهات برمجة التطبيقات (APIs) من خلال إرسال الطلبات (Requests) واستعراض الاستجابات (Responses) بطريقة سهلة وتفاعلية.

4.2.4 التحكم في الاصدارات باستخدام Git/Github :

- **Git** : هو نظام تحكم في الإصدارات (Version Control System) يُستخدم لمتابعة التعديلات على الشيفرة المصدريه، مما يسمح لك بحفظ نسخ متعددة من المشروع، الرجوع إلى نسخ سابقة، والعمل على فروع مختلفة دون التأثير على الشيفرة الرئيسية.



- **GitHub** : هو منصة مستضافة على الإنترنت تُستخدم لحفظ مشاريع Git ومشاركتها، وتوفر أدوات للتعاون مثل إدارة القضايا (Issues) ، طلبات السحب (Pull Requests) ، والمراجعة الجماعية للكود.

5.2.4 الاستضافات و الخدمات السحابية :

- **الاستضافة والنشر (Hosting & Deployment) :**
تم استخدام **Netlify** و **Railway** هي منصة سحابية تساعد المطورين على نشر وإدارة تطبيقاتهم وقواعد بياناتهم بسهولة دون الحاجة لإعداد البنية التحتية يدويًا.
- **خدمات البريد الالكتروني :**
تم استخدام خدمات مثل **Resend** لإرسال رسائل التحقق من الحساب المنجز مسبقًا.
- **خدمات تخزين الصور :**
Supabase Storage هي خدمة مدمجة ضمن منصة **Supabase** تُستخدم لتخزين وإدارة الملفات مثل الصور والفيديوهات، مع دعم التحكم في الوصول، والربط السهل مع قاعدة البيانات **PostgreSQL**.
- **قاعدة البيانات (Database) :**
يتم استخدام **PostgreSQL** لتخزين وإدارة جميع البيانات (حسابات المستخدمين، إعلانات السيارات، المعاملات، وغيرها)، ويتم الوصول إليها من خلال **Prisma ORM**.

3.4 تهيئة بيئة التطوير و التقنيات المستخدمة :

1.3.4 بيئة العمل الواجهة الامامية (Front-end) :

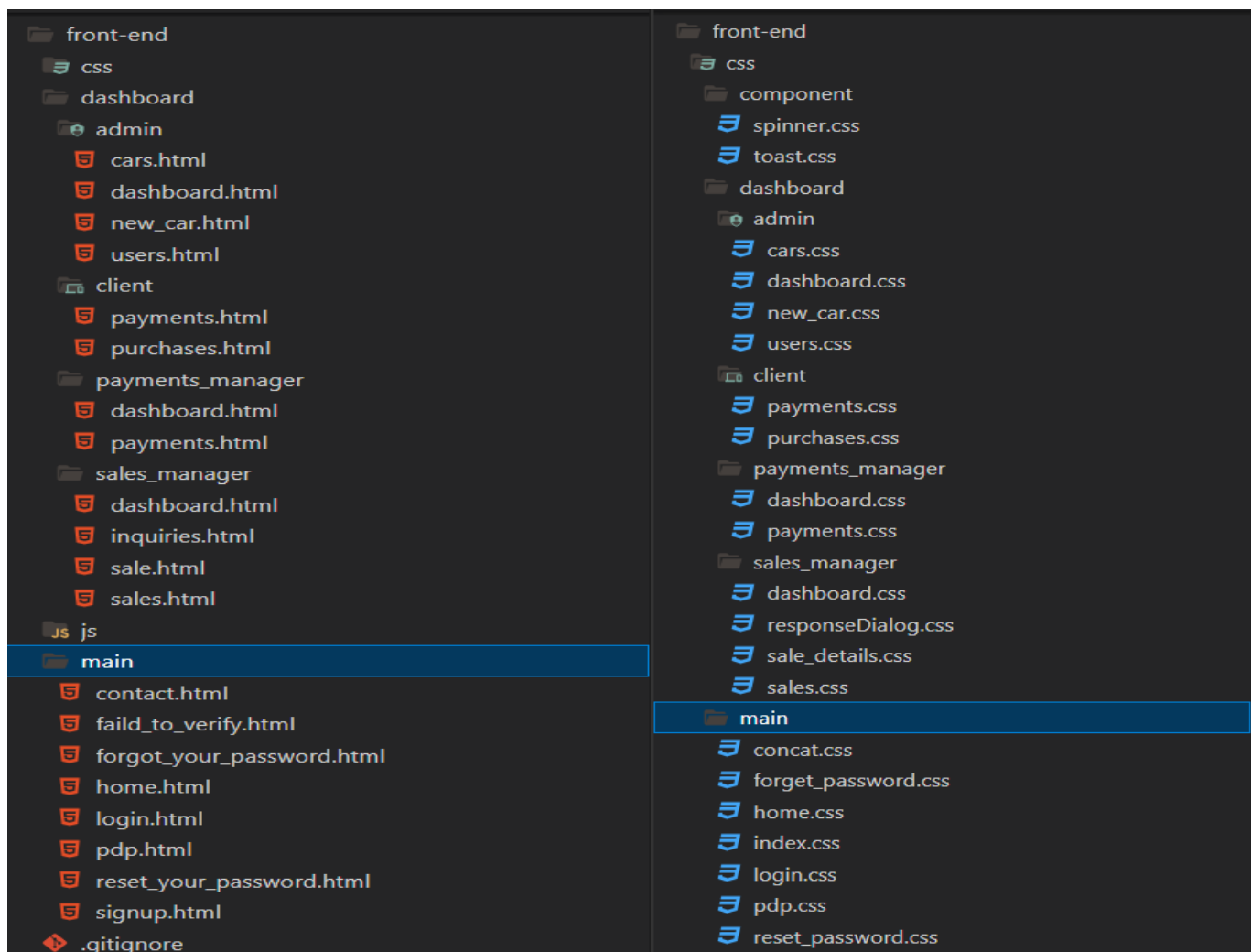
إعداد بيئة التطوير بشكل صحيح هو خطوة أساسية لضمان التنظيم، الأمان، وسهولة التطوير والاختبار.

• الطريقة التقليدية (HTML + CSS + JS) :

1. المميزات :

- بسيطة وسهلة الفهم.
- لا تحتاج أدوات بناء.(Build tools)
- مناسبة للمشاريع الصغيرة أو التعلم.

2 . هيكل المشروع:



الشكل 16 : هيكل الملفات الخاص بالواجهة الأمامية Front-End

2.3.4 بيئة العمل الواجهة الخلفية (Back-end)

اعداد بيئة تطوير مناسبة للواجهة الخلفية و هيكلته بطريقة منظمة تجعل العمل عليها سهلاً، لذا فقد تم اختيار المعمارية المناسبة لبناء الواجهة الخلفية للموقع و هي معمارية "REST API"

- **REST API** (Representational State Transfer Application Programming Interface) :

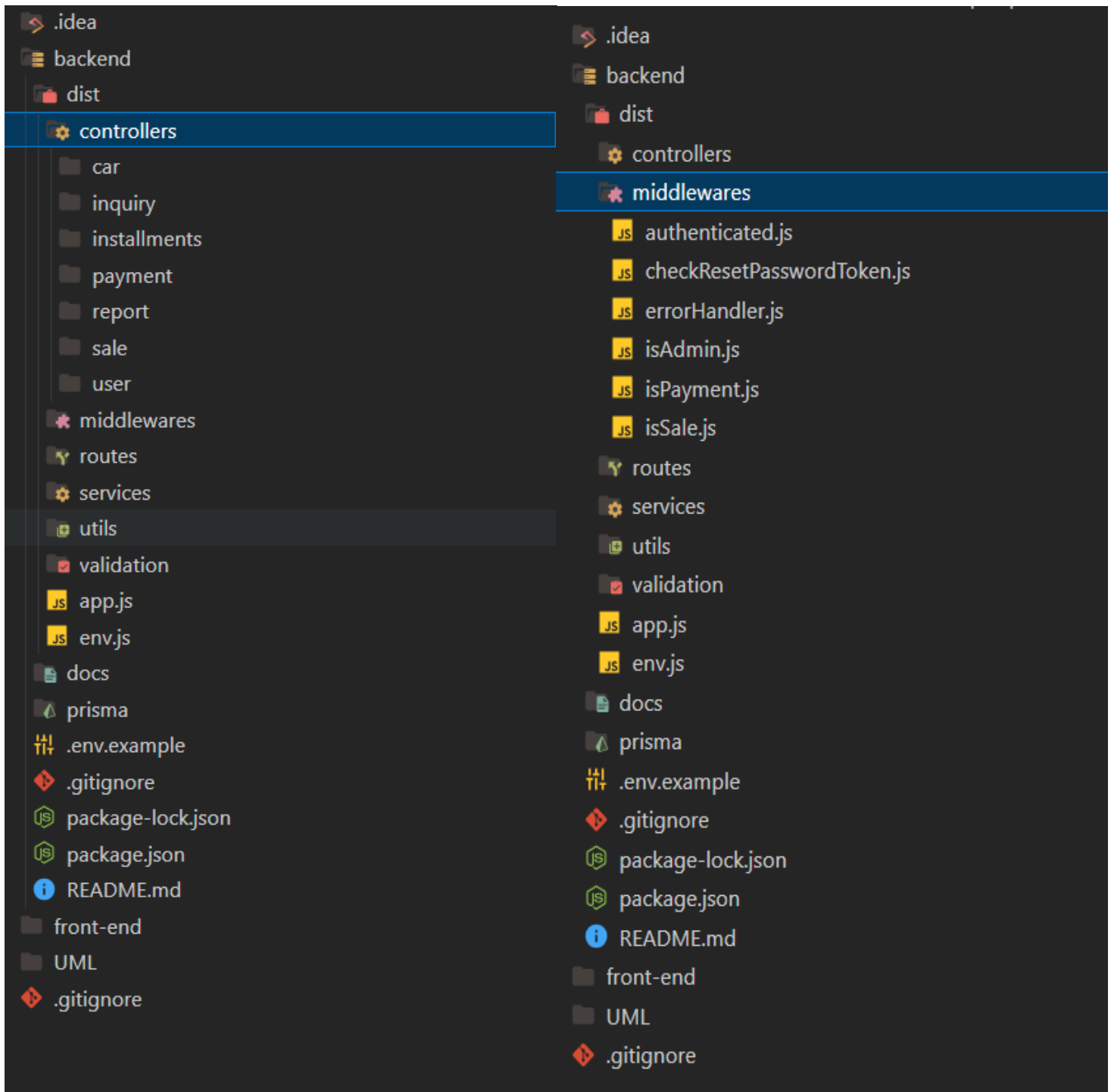
هو نمط معماري لتصميم واجهات برمجة التطبيقات (APIs) يعتمد على مبادئ بروتوكول HTTP ويُستخدم لتبادل البيانات بين الأنظمة بطريقة موحدة وسهلة الفهم.

- ابرز مميزاته :

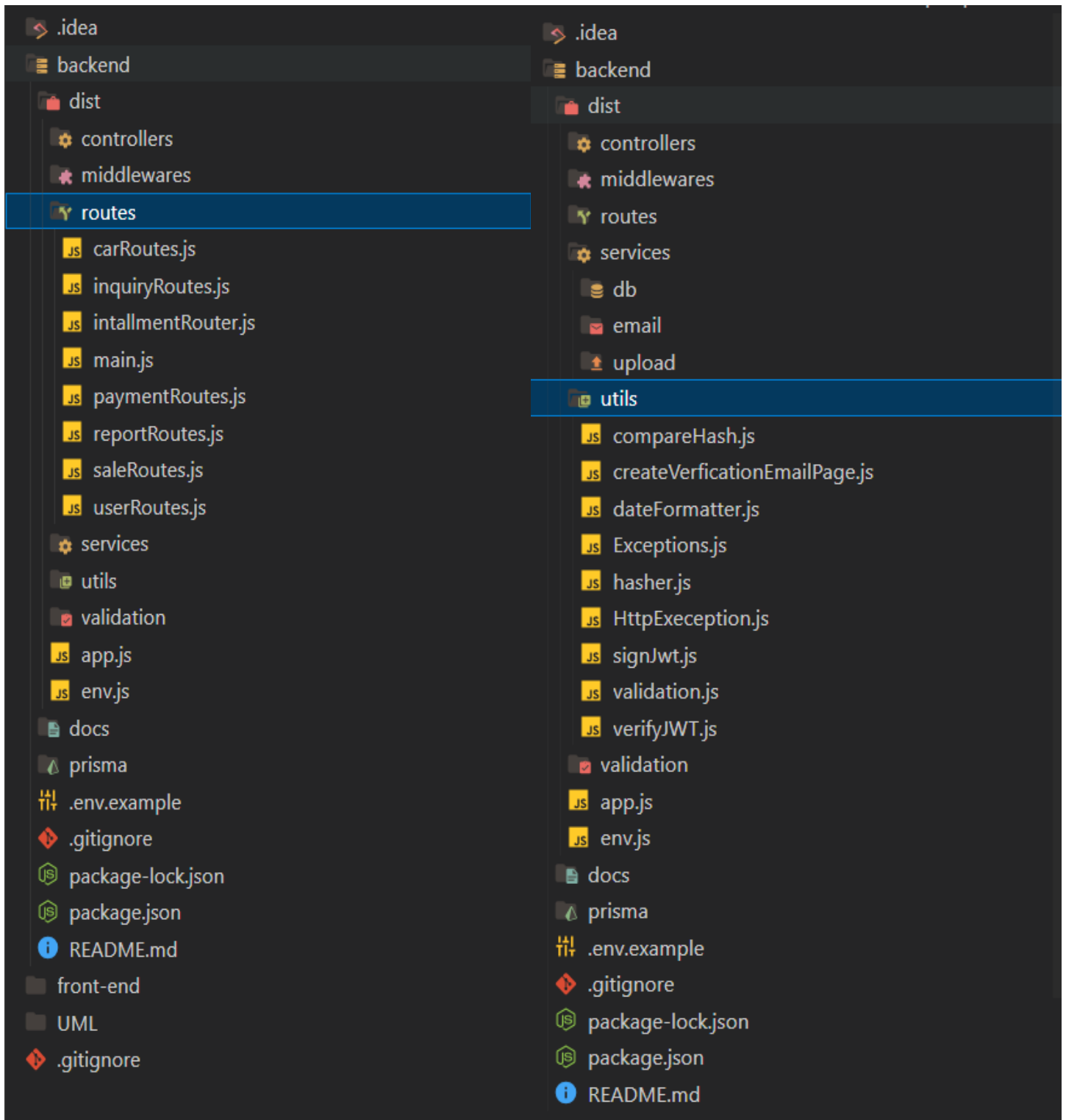
- كل طلب يجب ان يحتوي على كل المعلومات اللازمة لمعالجته دون الاعتماد على الطلب السابق .
- يعتمد على الموارد عبر روابط URL'S و يتم وصول اليها مثل (/users) (/cars)
- استخدام HTTP Methods :

جلب البيانات	GET
إنشاء بيانات جديدة	POST
تحديث البيانات	PUT/PATCH
حذف البيانات	DELETE

• هيكل الواجهة الخلفية (Back-end) :



الشكل 17 : هيكل الملفات الخاص بالواجهة الخلفية Back-End



الشكل 18 : هيكل الملفات الخاص بالواجهة الخلفية Back-End

4.4 تطوير الواجهة الخلفية :

1.4.4 اعداد الخادم و قاعدة البيانات :

إعداد الخادم وقاعدة البيانات هو خطوة أساسية لتشغيل أي تطبيق أو موقع ويب. يتضمن ذلك تهيئة الخادم (Server) وربطه بقاعدة البيانات (Database) لضمان تخزين البيانات ومعالجتها بكفاءة.

- اعداد الخادم (Server) :

Express.js هو إطار عمل Node.js لإنشاء خوادم ويب سريعة ومرنة. يتيح التعامل مع الطلبات، توجيه المسارات، وربط قواعد البيانات بسهولة.

- تثبيت Express.js :

بواسطة سطر امر المكتوب يتم تنزيل Express.js بهدف استخدامه لبناء الخادم .

```
bash Copy Download
npm install express
```

- انشاء الخادم :

الهدف من انشاء خادم هو توفير البيئة المناسبة و اساسية لارسال الطلبات

(Requests) و استقبال الردود (Responses)

```
app.js
backend > dist > .js app.js > ...
1 import express from "express";
2 import { PORT } from "./env.js";
3 import router from "./routes/main.js";
4 import cors from "cors";
5 import { ErrorResponse } from "./middlewares/errorHandler.js";
6 const app = express();
7 app.use(express.json());
8 app.use(express.urlencoded({ extended: true }));
9 app.use(cors({
10   origin: "*",
11 }));
12 app.get('/test', (req, res) => {
13   res.status(200).json({
14     ok: true,
15     msg: "app is work well"
16   });
17 });
18 app.use('/api', router);
19 app.use(ErrorResponse);
20 app.listen(PORT, (err) => {
21   err ? console.log("something went wrong") : console.log(`application run on port : ${PORT}`);
22 });
23 export default app;
```

الشكل 19 : تهيئة و اعداد الخادم Server بواسطة Express.js

2.4.4 تنفيذ عمليات CRUD :

CRUD هي العمليات الأساسية للتعامل مع البيانات: إنشاء (Create)، قراءة (Read)، تحديث (Update)، وحذف (Delete).

تُمثل هذه العمليات حجر الأساس في معظم التطبيقات، مثل إدارة المستخدمين أو المحتوى.

في Express.js، ننفذها عبر RESTful APIs باستخدام طلبات HTTP مثل `POST, GET, PUT, DELETE`.

و تشمل الموقع كاملا عمليات CRU :

● المستخدمين :

- اضافة او انشاء طلب على السيارة المختارة .
- عملية الدفع تتضمن ادخال معلومات بطاقة الائتمان.
- حذف الطلب في حالة تغيير رأي المستخدم.
- تحديث الطلب على السيارة او اعادة ادخال الطلب من جديد.

● المسؤولين :

- اضافة السيارات بهدف عرضها للبيع.
- اضافة العروض على السيارات.
- ترقية المستخدمين الى ادوار فعالة في الموقع.
- جدولة و تنظيم الطلبات المستقبلية من طرف الزبائن.

● المركبات :

- اضافة سيارة بكافة المعلومات الخاصة بها دون نقصان.
- اضافة العروض على السيارات.
- تعديل على بيانات السيارة بهدف مصدقية المعلومات خاصة بالسيارة.
- حذف السيارة في حالة بيعها او نفاذها

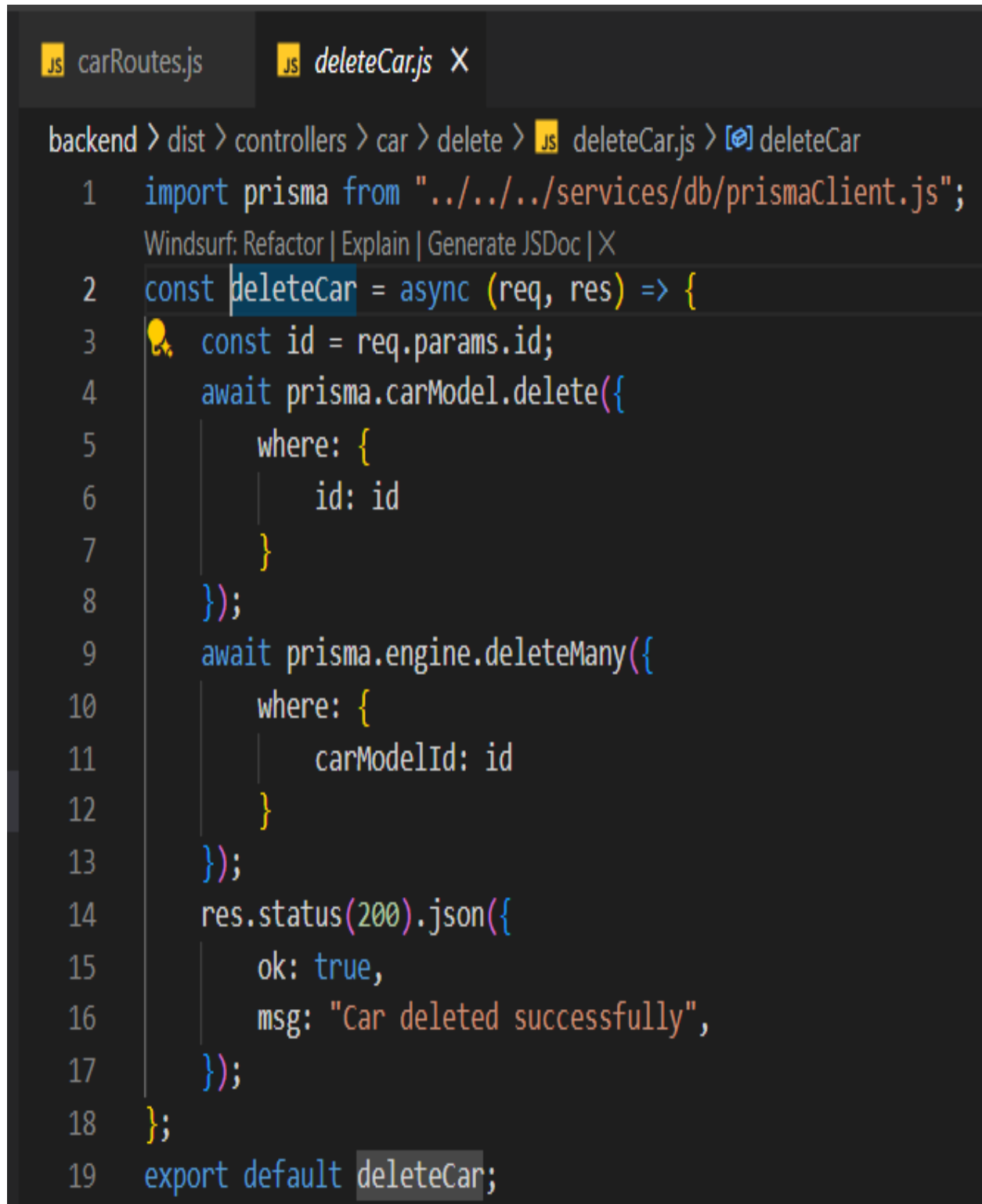
• الصور التوضيحية لعملية CRUD :

```
carRoutes.js  getCarsList.js X  getCar.js  getIncompletedInfoCars.js

backend > dist > controllers > car > browse > getCarsList.js > getCarsList

1 import prisma from "../../services/db/prismaClient.js";
  Windsurf: Refactor | Explain | Generate JSDoc | X
2 const getCarsList = async (req, res) => {
3   const cars = await prisma.carModel.findMany({
4     where: {
5       availability: true,
6     },
7     select: {
8       id: true,
9       model: true,
10      price: true,
11      discount: true,
12      mainImage: true,
13      transmission: true,
14      seats: true,
15      year: true,
16      availability: true,
17      engine: {
18        select: {
19          type: true,
20        }
21      }
22    },
23  });
24  res.status(200).json({
25    ok: true,
26    data: cars
27  });
28 };
29 export default getCarsList;
```

الشكل 20 : اظهار كافة السيارات في صفحة عرض السيارات



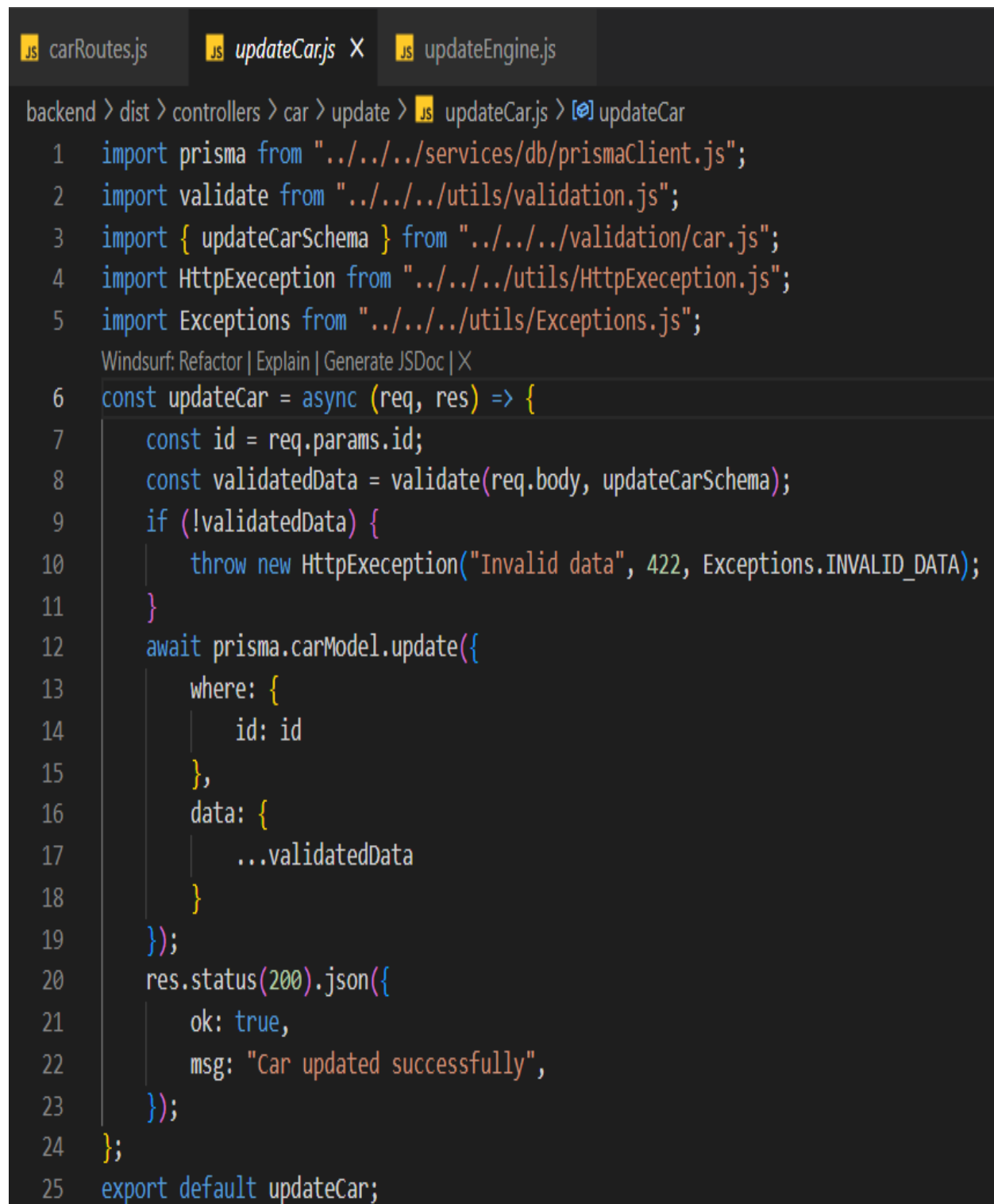
```
JS carRoutes.js JS deleteCar.js X
backend > dist > controllers > car > delete > JS deleteCar.js > [🔗] deleteCar
1 import prisma from "../../services/db/prismaClient.js";
  Windsurf: Refactor | Explain | Generate JSDoc | X
2 const deleteCar = async (req, res) => {
3   ⚠ const id = req.params.id;
4   await prisma.carModel.delete({
5     where: {
6       id: id
7     }
8   });
9   await prisma.engine.deleteMany({
10    where: {
11      carModelId: id
12    }
13  });
14  res.status(200).json({
15    ok: true,
16    msg: "Car deleted successfully",
17  });
18 };
19 export default deleteCar;
```

الشكل 21 : اظهار عملية حذف السيارة من قاعدة البيان



```
backend > dist > controllers > car > browse > JS getCar.js > ...
1 import prisma from "../../../services/db/prismaClient.js";
2 import HttpException from "../../../utils/HttpException.js";
3 import Exceptions from "../../../utils/Exceptions.js";
Windsurf: Refactor | Explain | Generate JSDoc | X
4 const getCar = async (req, res) => {
5   const carId = req.params.id;
6   const car = await prisma.carModel.findUnique({
7     where: {
8       id: carId,
9     },
10    include: {
11      engine: {
12        select: {
13          type: true,
14          capacity: true,
15          horsepower: true,
16        }
17      }
18    }
19  });
20  if (!car) {
21    throw new HttpException("Car not found", 404, Exceptions.NOT_FOUND);
22  }
23  res.status(200).json({
24    ok: true,
25    data: car
26  });
27 };
28 export default getCar;
```

الشكل 22 : اظهار السيارة المختارة



```
JS carRoutes.js JS updateCar.js X JS updateEngine.js
backend > dist > controllers > car > update > JS updateCar.js > [🔍] updateCar
1 import prisma from "../../services/db/prismaClient.js";
2 import validate from "../../utils/validation.js";
3 import { updateCarSchema } from "../../validation/car.js";
4 import HttpException from "../../utils/HttpException.js";
5 import Exceptions from "../../utils/Exceptions.js";
Windsurf: Refactor | Explain | Generate JSDoc | X
6 const updateCar = async (req, res) => {
7   const id = req.params.id;
8   const validatedData = validate(req.body, updateCarSchema);
9   if (!validatedData) {
10     throw new HttpException("Invalid data", 422, Exceptions.INVALID_DATA);
11   }
12   await prisma.carModel.update({
13     where: {
14       id: id
15     },
16     data: {
17       ...validatedData
18     }
19   });
20   res.status(200).json({
21     ok: true,
22     msg: "Car updated successfully",
23   });
24 };
25 export default updateCar;
```

الشكل 23 : عملية تحديث بيانات السيارة

3.4.4 تطوير التوجيهات Routes :

تُحدد كيفية استجابة الخادم للطلبات (Requests) بناءً على مسارات (URLs) وطرق HTTP (GET, POST, etc.).

تقسيم الـ Routes يسهل تنظيم الكود ويجعل التطبيق أكثر قابلية للصيانة والتوسع. في إطار عمل مثل Express.js ، تُعرّف الـ Routes باستخدام Router() لإنشاء مسارات متعددة وربطها بوظائف (Controllers) معالجة الطلبات.

• مميزات استخدام Routes :

1. تنظيم الكود : تجزئة المسارات (Routes) يُحسن بنية المشروع ويجعل الكود أكثر تنظيماً وسهولة في الصيانة.
2. توجيه الطلبات: تحديد استجابة الخادم لكل مسار (URL) وطريقة HTTP (GET, POST, PUT, DELETE) بشكل دقيق.
3. تحسين الأمان: فصل الصلاحيات عبر مسارات محددة (مثل `admin/` للمدراء فقط).
4. سهولة التوسعة: إضافة مميزات جديدة عبر إنشاء مسارات دون التأثير على الأكواد الموجودة.
5. تحسين الأداء: تقليل التحميل على الخادم بتوجيه الطلبات مباشرة إلى الوظائف المختصة (Controllers).

```
main.js x paymentRoutes.js reportRoutes.js saleRoutes.js userRoutes.js
backend > dist > routes > main.js > ...
1 |import { Router } from "express";
2 |import usersRouter from "../userRoutes.js";
3 |import carsRouter from "../carRoutes.js";
4 |import saleRouter from "../saleRoutes.js";
5 |import paymentRouter from "../paymentRoutes.js";
6 |import installmentRouter from "../installmentRouter.js";
7 |import inquiryRouter from "../inquiryRoutes.js";
8 |import reportRouter from "../reportRoutes.js";
9 |const router = Router();
10 |router.use('/user', usersRouter);
11 |router.use('/car', carsRouter);
12 |router.use('/sale', saleRouter);
13 |router.use('/payment', paymentRouter);
14 |router.use('/installment', installmentRouter);
15 |router.use('/inquiry', inquiryRouter);
16 |router.use('/report', reportRouter);
17 |export default router;
```

الشكل 24 : واجهة التوجيه الرئيسية في الواجهة الخلفية

```
carRoutes.js inquiryRoutes.js x main.js paymentRoutes.js reportRoutes.js saleRoutes.js
backend > dist > routes > inquiryRoutes.js > ...
1 |import { Router } from "express";
2 |import { newInquiry } from "../controllers/inquiry/newInquiry.js";
3 |import { errorHandler } from "../middlewares/errorHandler.js";
4 |import { getInquiries } from "../controllers/inquiry/getInquiries.js";
5 |import isSale from "../middlewares/isSale.js";
6 |import { getInquiry } from "../controllers/inquiry/getInquiry.js";
7 |import { responseInquiry } from "../controllers/inquiry/responseInquiry.js";
8 |import { deleteInquiry } from "../controllers/inquiry/deleteInquiry.js";
9 |const inquiryRouter = Router();
10 |inquiryRouter.post('/', errorHandler(newInquiry));
11 |inquiryRouter.get('/', errorHandler(isSale), errorHandler(getInquiries));
12 |inquiryRouter.get('/:id', errorHandler(isSale), errorHandler(getInquiry));
13 |inquiryRouter.post('/response/:id', errorHandler(isSale), errorHandler(responseInquiry));
14 |inquiryRouter.delete('/:id', errorHandler(isSale), errorHandler(deleteInquiry));
15 |export default inquiryRouter;
```

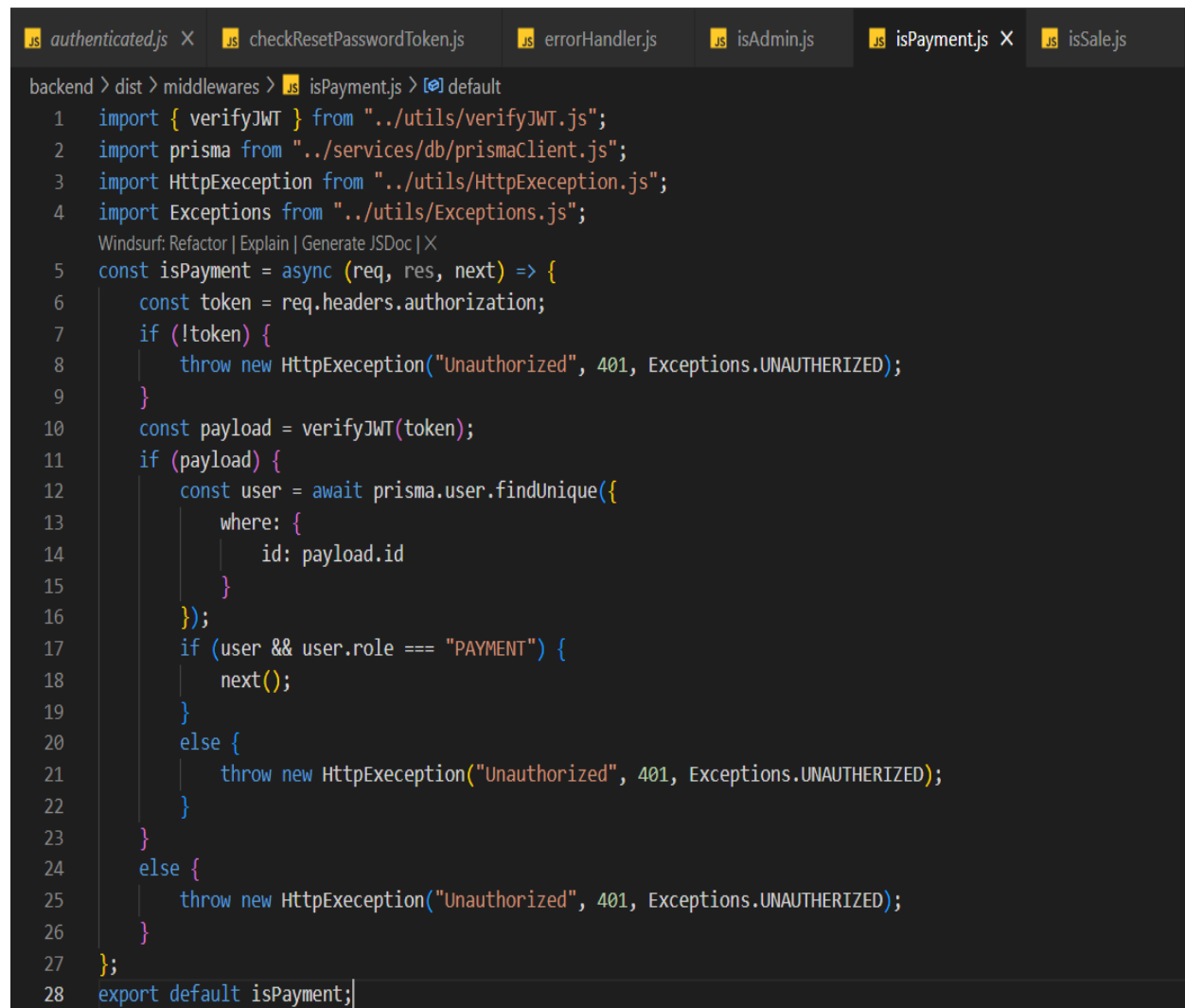
الشكل 25 : واجهة التوجيه للاستفسارات في الواجهة الخلفية

4.4.4 ادارة حسابات المستخدمين و الصلاحيات :

1. إدارة المستخدمين: نظام يتحكم في إنشاء الحسابات، تعديلها، وحذفها مع ضمان أمان البيانات.

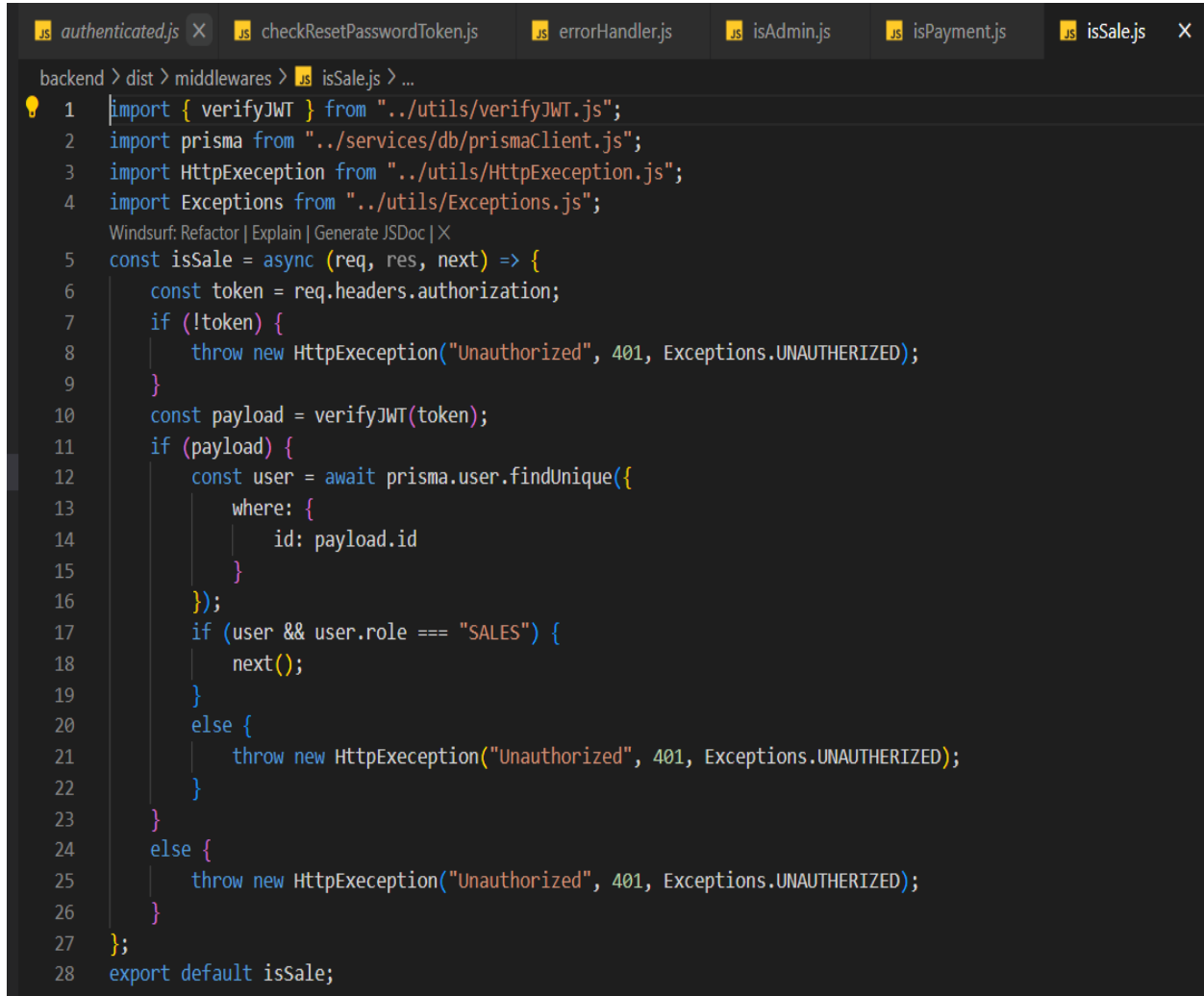
2. الصلاحيات: تحديد مستوى وصول كل مستخدم (مثل: مدير، مستخدم عادي) لمنع التجاوزات.

الشكل 26 : واجهة توضيح صلاحية Admin



```
backend > dist > middlewares > isPayment.js > default
1 import { verifyJWT } from "../utils/verifyJWT.js";
2 import prisma from "../services/db/prismaClient.js";
3 import HttpException from "../utils/HttpException.js";
4 import Exceptions from "../utils/Exceptions.js";
5 const isPayment = async (req, res, next) => {
6   const token = req.headers.authorization;
7   if (!token) {
8     throw new HttpException("Unauthorized", 401, Exceptions.UNAUTHORIZED);
9   }
10  const payload = verifyJWT(token);
11  if (payload) {
12    const user = await prisma.user.findUnique({
13      where: {
14        id: payload.id
15      }
16    });
17    if (user && user.role === "PAYMENT") {
18      next();
19    }
20    else {
21      throw new HttpException("Unauthorized", 401, Exceptions.UNAUTHORIZED);
22    }
23  }
24  else {
25    throw new HttpException("Unauthorized", 401, Exceptions.UNAUTHORIZED);
26  }
27 };
28 export default isPayment;
```

الشكل 27 : واجهة توضيح صلاحية Payment



```
backend > dist > middlewares > isSale.js > ...
1 import { verifyJWT } from "../utils/verifyJWT.js";
2 import prisma from "../services/db/prismaClient.js";
3 import HttpException from "../utils/HttpException.js";
4 import Exceptions from "../utils/Exceptions.js";
5 const isSale = async (req, res, next) => {
6   const token = req.headers.authorization;
7   if (!token) {
8     throw new HttpException("Unauthorized", 401, Exceptions.UNAUTHORIZED);
9   }
10  const payload = verifyJWT(token);
11  if (payload) {
12    const user = await prisma.user.findUnique({
13      where: {
14        id: payload.id
15      }
16    });
17    if (user && user.role === "SALES") {
18      next();
19    }
20    else {
21      throw new HttpException("Unauthorized", 401, Exceptions.UNAUTHORIZED);
22    }
23  }
24  else {
25    throw new HttpException("Unauthorized", 401, Exceptions.UNAUTHORIZED);
26  }
27 };
28 export default isSale;
```

الشكل 28 : واجهة توضح صلاحية Sales

5.4.4 ربط الواجهة الخلفية بقاعدة البيانات :

Prisma هو (ORM (Object-Relational Mapping) تحديث يعمل مع PostgreSQL (وغيره من قواعد البيانات) لتبسيط التعامل مع البيانات. يوفر واجهة سهلة وآمنة لنمذجة البيانات وتنفيذ الاستعلامات.

- تثبيت Prisma :

```
bash
```

[Copy](#) [Download](#)

```
npm install prisma @prisma/client
```

- تهيئة Prisma :

```
bash
```

[Copy](#) [Download](#)

```
npx prisma init
```

- تهيئة الاتصال بين PostgreSQL في 'schema.prisma'

```
prisma
```

[Copy](#) [Download](#)

```
datasource db {
  provider = "postgresql"
  url      = env("DATABASE_URL") // يُعرّف في ملف .env
}

generator client {
  provider = "prisma-client-js"
}
```

- اعداد متغير البيئة .env.

```
env
```

[Copy](#) [Download](#)

```
DATABASE_URL="postgresql://USER:PASSWORD@HOST:PORT/DATABASE_NAME?schema=public"
```

• تعريف الـ (Models) للجداول :

```
schema.prisma X
backend > prisma > schema.prisma > ...
100
101 //User model
102 model User {
103     id          String      @id @default(cuid())
104     firstName   String
105     lastName    String
106     email       String      @unique
107     phone       String
108     password    String
109     role        Role        @default(CLIENT)
110     isVerified  Boolean      @default(false)
111     sales       Sale[]
112     createdAt   DateTime     @default(now())
113     updatedAt   DateTime     @updatedAt
114
115     @@index([email, id])
116 }
117
118 //Car Model model
119 model CarModel {
120     //initialize
121     id          String      @id @default(cuid())
122     model       String
123     year        Int
124     type        CarModelType
125     //second step
126     engine      Engine[]
127     //third step
128     transmission Transmission
129     driveType   DriveType
130     colors      String[]    @default([])
131     features    String[]    @default([])
132     seats       Int?
133     doors       Int?
134     maxSpeed    Int?
135     price       Int
136     discount    Int
```

الشكل 29 : مخطط الـ Models في ملف schema.prisma

5.4 تطوير الواجهة الامامية :

1.5.4 الاجزاء الرئيسية للصفحات :

تتكون الصفحات من ثلاثة اقسام او بما يعرف بـ Components :

- **Navbar** : (شريط التنقل) هو عنصر واجهة مستخدم موجود أعلى الصفحة، يحتوي على روابط رئيسية لتسهيل التنقل بين أقسام الموقع أو التطبيق. عادةً ما يتضمن شعار الموقع، قوائم، وأزرار بحث أو تسجيل دخول. يُعد جزءًا أساسيًا في التصميم لتحسين تجربة المستخدم وإمكانية الوصول.
- **Hero** : (القطعة البطولية) هو قسم رئيسي في الصفحة الرئيسية للموقع، يظهر مباشرةً تحت ال-Navbar. يحتوي عادةً على عنوان جذاب، صورة أو خلفية مميزة، ووصف مختصر مع دعوة لإجراء (CTA). يهدف إلى جذب انتباه الزوار وإبراز أهم محتوى أو عرض في الموقع.
- **Footer** : هو الجزء الثابت في أسفل الصفحة، ويحتوي عادةً على روابط مهمة مثل سياسة الخصوصية، معلومات الاتصال، وسائل التواصل الاجتماعي، وخريطة الموقع. يُعد عنصرًا أساسيًا لتحسين تجربة المستخدم وتسهيل الوصول إلى المحتوى الإضافي. غالبًا ما يتضمن حقوق النشر والعلامة التجارية.

هذه هي الاجزاء الاساسية التي يجب ان تتوفر في كل الصفحات.

2.5.4 انشاء الصفحات الرئيسية :

- **Home page:** هي الواجهة الأساسية لأي موقع ويب أو تطبيق، وتُعد نقطة البداية التي يصل إليها المستخدمون. تحتوي عادةً على نظرة عامة عن المحتوى، وأهم الأقسام، وروابط التنقل الرئيسية لتسهيل الوصول إلى الخدمات المقدمة.

```

home.html X contact.html signup.html login.html pdp.html
front-end > main > home.html > ...
1
2 <!DOCTYPE html>
3 <html lang="ar" dir="rtl">
4 <head>
5   <meta charset="UTF-8" />
6   <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
7   <title>الصفحة الرئيسية</title>
8   <link href="https://fonts.googleapis.com/css2?family=Cairo:wght@400;700&display=swap" rel="stylesheet"/>
9   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.5.0/css/all.min.css"/>
10  <link rel="stylesheet" type="text/css" href="css/main/home.css">
11 </head>
12 <body>
13
14 <nav>
15   <div class="logo">
16     
17     <h1>بيع السيارات</h1>
18   </div>
19
20   <!-- Burger Icon -->
21   <button class="burger" id="burgerToggle" aria-label="فتح القائمة">
22     <i class="fas fa-bars"></i>
23   </button>
24
25   <!-- Links -->
26   <ul class="nav-links" id="nav-links">
27     <li><a href="#">الرئيسية</a></li>
28     <li><a href="/contact.html">اتصل بنا</a></li>
29     <li><a href="/index.html">من نحن</a></li>
30   </ul>
31 </nav>
32 <div class="toggle-mode">
33   <button id="darkModeToggle">
34     <i class="fas fa-moon"></i>
35     <span>الوضع الليلي</span>
36   </button>
37 </div>
38 <div class="page">

```

الشكل 30 : كود البرمجي للواجهة الرئيسية Home Page

- **About Us** : هي قسم في الموقع يُعرّف بالشركة أو الفريق، ويوضح رؤيته، رسالته، وقيمه الأساسية. تهدف إلى بناء الثقة مع الزوار من خلال مشاركة تاريخ المؤسسة، أهدافها، وأبرز إنجازاتها.

```

home.html contact.html signup.html login.html pdp.html index.html X
front-end > index.html > ...
  2  <html lang="ar" dir="rtl">
 11  <body>
32
33  <div class="toggle-mode">
34    <button id="darkModeToggle" >
35      <i class="fas fa-moon"></i>
36      <span>الوضع الليلي</span>
37    </button>
38  </div>
39
40  <div class="container">
41    <h1>من نحن</h1>
42    <p class="about-text">
43      نرعى لتوفير تجربة سلسة وموثوقة لعملائنا. هدفنا هو تقديم سيارات حديثة، آمنة، وبأسعار تنافسية تناسب جميع الاحتياجات
44    </p>
45
46    <div class="about-section">
47      
48      <div>
49        <h2>رؤيتنا</h2>
50        <p>أن نكون الخيار الأول في بيع السيارات من خلال الابتكار والراحة والثقة</p>
51        <h2>رسالتنا</h2>
52        <p>توفير خدمة بيع سهلة، آمنة، وميسرة تناسب جميع فئات العملاء</p>
53      </div>
54    </div>
55
56    <h2>فريق العمل</h2>
57    <div class="team">
58      <div class="team-card">
59        
60        <h3>أحمد الزمراني</h3>
61        <p>المدير التنفيذي</p>
62      </div>
63      <div class="team-card">
64        
65        <h3>سارة العتيبي</h3>
66        <p>مديرة التسويق</p>

```

الشكل 31 : كود البرمجي للواجهة About Us

- **Login/Signup**: هي واجهة تسمح للمستخدمين بالدخول إلى حساباتهم الشخصية أو إنشاء حساب جديد. تحتوي عادةً على حقول لإدخال البيانات مثل البريد الإلكتروني وكلمة المرور، مع خيارات مثل استعادة كلمة المرور أو الربط بحسابات وسائل التواصل الاجتماعي.

• Signup :

```

home.html contact.html signup.html X login.html pdp.html
front-end > main > signup.html > ...
1 <!DOCTYPE html>
2 <html lang="ar" dir="rtl">
3
4 <head>
5   <meta charset="UTF-8" />
6   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7   <title>إنشاء حساب</title>
8   <link href="https://fonts.googleapis.com/css2?family=Cairo:wght@400;700&display=swap" rel="stylesheet" />
9   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.5.0/css/all.min.css" />
10  <link rel="stylesheet" type="text/css" href="../css/main/signup.css">
11  <link rel="stylesheet" type="text/css" href="../css/component/toast.css">
12  <link rel="stylesheet" type="text/css" href="../css/component/spinner.css">
13 </head>
14
15 <body>
16
17   <div class="toggle-mode">
18     <button id="darkModeToggle">
19       <i class="fas fa-moon"></i>
20       <span>الوضع الليلي</span>
21     </button>
22   </div>
23   <!-- Toast Container -->
24   <div id="toastContainer" class="toast-container"></div>
25   <div class="signup-container">
26     <div class="logo">
27       
28     </div>
29     <h2>إنشاء حساب</h2>
30     <form id="signupForm">
31       <div class="form-group">
32         <i class="fas fa-user"></i>
33         <input type="text" id="first-name" placeholder="الاسم" required>
34       </div>
35       <div class="form-group">
36         <i class="fas fa-user-tag"></i>
37         <input type="text" id="last-name" placeholder="اللقب" required>
38       </div>
39     </form>
40   </div>
41 </body>
42 </html>

```

الشكل 32 : كود البرمجي للواجهة انشاء حساب

• Login :

```

home.html contact.html signup.html login.html X pdp.html
front-end > main > login.html > ...
1 |<!DOCTYPE html>
2 |<html lang="ar" dir="rtl">
3 |<head>
4 |   <meta charset="UTF-8" />
5 |   <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
6 |   <title>تسجيل الدخول</title>
7 |   <link href="https://fonts.googleapis.com/css2?family=Cairo:wght@400;700&display=swap" rel="stylesheet"/>
8 |   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.5.0/css/all.min.css"/>
9 |   <link rel="stylesheet" type="text/css" href="../../css/main/login.css">
10 |  <link rel="stylesheet" type="text/css" href="../../css/component/toast.css">
11 |  <link rel="stylesheet" type="text/css" href="../../css/component/spinner.css">
12 |</head>
13 |<body>
14 |
15 |   <div class="toggle-mode">
16 |     <button id="darkModeToggle" >
17 |       <i class="fas fa-moon"></i>
18 |       <span>الوضع الليلي</span>
19 |     </button>
20 |   </div>
21 |   <!-- Toast Container -->
22 |   <div id="toastContainer" class="toast-container"></div>
23 |
24 |   <div class="login-container" id="loginBox">
25 |     <div class="logo">
26 |       
27 |     </div>
28 |     <h2>تسجيل الدخول</h2>
29 |     <div id="loginForm">
30 |       <div class="form-group">
31 |         <i class="fas fa-envelope"></i>
32 |         <input type="email" id="email" placeholder="البريد الإلكتروني" required>
33 |       </div>
34 |       <div class="form-group">
35 |         <i class="fas fa-lock"></i>
36 |         <i class="fas fa-eye toggle-pass" id="togglePassword"></i>
37 |         <input type="password" id="password" placeholder="كلمة المرور" required>
38 |       </div>

```

الشكل 33 : كود البرمجي للواجهة Login

- **(Contacts) :** هي قسم مخصص لتوفير معلومات التواصل مع الشركة أو الفريق، مثل: البريد الإلكتروني، رقم الهاتف، العنوان، أو نموذج تواصل مباشر. تهدف إلى تمكين الزوار من التواصل بسهولة مع الدعم أو خدمة العملاء عند الحاجة.

```
home.html contact.html X signup.html login.html pdp.html
front-end > main > contact.html > ...
  2  <html lang="ar" dir="rtl">
 14  <body>
 15  <nav>
 21  <!-- Burger Icon -->
 22  <button class="burger" id="burgerToggle" aria-label="فتح القائمة">
 23    <i class="fas fa-bars"></i>
 24  </button>
 25
 26  <!-- Links -->
 27  <ul class="nav-links" id="nav-links">
 28    <li><a href="/home.html">الرئيسية</a></li>
 29    <li><a href="#">اتصل بنا</a></li>
 30    <li><a href="/index.html">من نحن</a></li>
 31  </ul>
 32  </nav>
 33
 34  <div class="toggle-mode">
 35    <button id="darkModeToggle">
 36      <i class="fas fa-moon"></i>
 37      <span>الوضع الليلي</span>
 38    </button>
 39  </div>
 40
 41  <div class="container">
 42    <h1>اتصل بنا</h1>
 43    <p class="contact-description">
 44      يسعدنا تواصلك معنا! سواء كان لديك استفسار، اقتراح، أو ترغب في معرفة المزيد عن خدماتنا، لا تتردد في إرسال رسالة أو
 45      الاتصال بنا مباشرة.
 46    </p>
 47
 48    <div class="contact-layout">
 49      <form class="contact-form">
 50        <input type="text" id="full-name" placeholder="الاسم الكامل" required />
 51        <input type="email" id="email" placeholder="البريد الإلكتروني" required />
 52        <textarea rows="5" id="message" placeholder="اكتب رسالتك هنا..." required></textarea>
 53        <button type="button" id="submit">إرسال الرسالة</button>
 54      </form>
```

الشكل 34 : كود البرمجي للواجهة Contact

3.5.4 انشاء بوابة الدفع : بوابة الدفع هي نظام آمن يسمح للمستخدمين بإتمام المعاملات المالية عبر الموقع أو التطبيق، مثل شراء الخدمات أو الاشتراكات. تُعد عنصرًا حيويًا لضمان تجربة دفع سلسلة وآمنة، مع دعم وسائل دفع متعددة (بطاقات ائتمان، محافظ إلكترونية، تحويل بنكي).

```

payments.html X
front-end > dashboard > payments_manager > payments.html > ...
1 <!DOCTYPE html>
2 <html lang="ar" dir="rtl">
3
4 <head>
5   <meta charset="UTF-8" />
6   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7   <title>البوابات المدفوعة</title>
8   <link href="https://fonts.googleapis.com/css2?family=Cairo:wght@400;700&display=swap" rel="stylesheet">
9   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.5.0/css/all.min.css">
10  <link rel="stylesheet" href="../../css/dashboard/payments_manager/payments.css" />
11  <link rel="stylesheet" href="../../css/component/toast.css" />
12  <link rel="stylesheet" href="../../css/component/spinner.css" />
13 </head>
14
15 <body>
16   <div class="sidebar">
17     <div>
18       <h2><i class="fas fa-credit-card"></i> بوابة الدفع</h2>
19       <ul>
20         <li><a href="./dashboard.html"><i class="fas fa-home"></i> الرئيسية</a></li>
21         <li class="active"><a href="#"><i class="fas fa-credit-card"></i> المدفوعات</a></li>
22       </ul>
23     </div>
24     <button class="logout" id="logout-btn"><i class="fas fa-sign-out-alt"></i> تسجيل الخروج</button>
25   </div>
26
27   <div class="main">
28     <div class="dashboard-header">عمليات الدفع</div>
29     <div class="actions-container">
30       <div class="search-container">
31         <input type="text" id="email-search" placeholder="أدخل معرف عملية البيع" />
32         <button id="search-btn" class="btn-search"><i class="fas fa-search"></i></button>
33       </div>
34     </div>
35     <br/>
36     <div class="table-container">
37       <table>
38

```

الشكل 35 : كود البرمجي بوابة الدفع

4.5.4 انشاء Dashboard :

لوحة التحكم هي واجهة مركزية تتيح للمستخدمين أو المسؤولين عرض البيانات المهمة وإدارتها بطريقة سهلة ومرئية. تحتوي عادةً على إحصائيات، رسوم بيانية، وإشعارات لتوفير نظرة عامة سريعة على الأداء والأنشطة.

- **Admin Dashboard :** لوحة تحكم المشرف (Admin Dashboard) هي واجهة متكاملة تمنح المسؤولين تحكمًا كاملاً في إدارة النظام. تحتوي على أدوات متقدمة لمراقبة المستخدمين، تحليل البيانات، وإعدادات النظام مع واجهة سهلة للتحكم في جميع جوانب المنصة.

```
dashboard.html X
front-end > dashboard > admin > dashboard.html > ...
1 <!DOCTYPE html>
2 <html lang="ar" dir="rtl">
3
4 <head>
5   <meta charset="UTF-8" />
6   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7   <title>لوحة تحكم المشرف</title>
8   <link href="https://fonts.googleapis.com/css2?family=Cairo:wght@400;700&display=swap" rel="stylesheet">
9   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.5.0/css/all.min.css">
10  <link rel="stylesheet" href="../../css/dashboard/admin/dashboard.css" />
11  <link rel="stylesheet" href="../../css/component/toast.css" />
12  <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
13 </head>
14
15 <body>
16
17   <div class="sidebar">
18     <div>
19       <h2><i class="fas fa-car"></i> لوحة المشرف</h2>
20       <ul>
21         <li class="active"><a href="#"><i class="fas fa-chart-line"></i> لوحة التحكم</a></li>
22         <li><a href="./cars.html"><i class="fas fa-car-side"></i> السيارات</a></li>
23         <li><a href="./users.html"><i class="fas fa-users"></i> المستخدمين</a></li>
24       </ul>
25     </div>
26     <button class="logout" id="logout-btn"><i class="fas fa-sign-out-alt"></i> تسجيل الخروج</button>
27   </div>
28
29   <div class="main">
30     <div class="dashboard-header">لوحة تحكم المشرف</div>
31     <div class="cards">
32       <div class="card">
33         <div class="info">
34           <h3>إجمالي السيارات</h3>
35           <div class="value" id="total-cars"></div>
36         </div>
37         <div class="icon"><i class="fas fa-car"></i></div>
38       </div>
39     </div>
40   </div>
41 </body>
42 </html>
```

الشكل 36 : كود البرمجي للواجهة Dashboard

```
cars.html X
front-end > dashboard > admin > cars.html > ...
1 <!DOCTYPE html>
2 <html lang="ar" dir="rtl">
3
4 <head>
5   <meta charset="UTF-8" />
6   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7   <title>إدارة السيارات</title>
8   <link href="https://fonts.googleapis.com/css2?family=Cairo:wght@400;700&display=swap" rel="stylesheet">
9   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.5.0/css/all.min.css">
10  <link rel="stylesheet" href="../../css/dashboard/admin/cars.css" />
11  <link rel="stylesheet" href="../../css/component/toast.css" />
12 </head>
13 <body>
14   <div class="sidebar">
15     <div>
16       <h2><i class="fas fa-car"></i> لوحة المشرف</h2>
17       <ul>
18         <li><a href="/dashboard.html"><i class="fas fa-chart-line"></i> لوحة التحكم</a></li>
19         <li class="active"><a href="#"><i class="fas fa-car-side"></i> السيارات</a></li>
20         <li><a href="/users.html"><i class="fas fa-users"></i> المستخدمين</a></li>
21       </ul>
22     </div>
23     <button class="logout" id="logout-btn"><i class="fas fa-sign-out-alt"></i> تسجيل الخروج</button>
24   </div>
25
26   <div class="main">
27     <div class="dashboard-header">إدارة السيارات</div>
28     <button class="btn-add"><a href="/new_car.html"><i class="fas fa-plus"></i> إضافة سيارة جديدة</a></button>
29     <div class="table-container">
30       <table>
31         <thead>
32           <tr>
33             <th>المعرف</th>
34             <th>الموديل</th>
35             <th>السنة</th>
36             <th>السعر</th>
37             <th>الحالة</th>
38           </tr>
39         </thead>
40       </table>
41     </div>
42   </div>
43 </body>
44 </html>
```

الشكل 37 : كود البرمجي للواجهة لإدارة السيارات

- **Client Dashboard**: لوحة تحكم العميل (Client Dashboard) هي الواجهة الرئيسية التي تتيح للمستخدمين العاديين إدارة حساباتهم وخدماتهم بشكل مركزي. تحتوي على أدوات مبسطة لمتابعة المعاملات، تحديث البيانات الشخصية، والوصول السريع للخدمات الأساسية.

المميزات الأساسية:

- نظرة عامة على الحساب والنشاطات الحديثة
- إدارة الطلبات والاشتراكات
- واجهة تفاعلية سهلة الاستخدام مع إمكانية التخصيص

```
payments.html x
front-end > dashboard > client > payments.html > ...
2  <html lang="ar" dir="rtl">
15 <body>
16
17 <div class="sidebar">
18   <div>
19     <h2><i class="fas fa-user"></i> بوابة الزبون </h2>
20     <ul>
21       <li><a href="/purchases.html"><i class="fas fa-shopping-cart"></i> مشترياتي </a></li>
22       <li class="active"><a href="#"><i class="fas fa-money-bill-wave"></i> مدفوعاتي </a></li>
23     </ul>
24   </div>
25   <button class="logout" id="logout-btn"><i class="fas fa-sign-out-alt"></i> تسجيل الخروج </button>
26 </div>
27
28 <div class="main">
29   <div class="dashboard-header">مدفوعاتي</div>
30   <table>
31     <thead>
32       <tr>
33         <th>رقم الدفع</th>
34         <th>التاريخ</th>
35         <th>المبلغ</th>
36         <th>النوع</th>
37         <!--<th>الحالة</th>-->
38       </tr>
39     </thead>
40     <tbody id="payments-table">
41     </tbody>
42   </table>
43
44 </div>
45 <script type="module" src="../../js/utls/config.js"></script>
46 <script type="module" src="../../js/auth/authorized.js"></script>
47 <script type="module" src="../../js/auth/isClient.js"></script>
48 <script type="module" src="../../js/auth/logout.js"></script>
49 <script type="module" src="../../js/utls/showToast.js"></script>
50 <script type="module" src="../../js/dashboard/client/myPayments.js"></script>
51 </body>
```

الشكل 38 : كود البرمجي للواجهة Dashboard Client

```
purchases.html X
front-end > dashboard > client > purchases.html > ...
2 <html lang="ar" dir="rtl">
13 <body>
24 </div>
25
26 <div class="main">
27 <div class="dashboard-header">مشترياتي</div>
28 <div class="cards">
29 <div class="card">
30 <div class="info">
31 <h3>إجمالي المشتريات</h3>
32 <div class="value" id="total-sales"></div>
33 </div>
34 <div class="icon"><i class="fas fa-box-open"></i></div>
35 </div>
36 <div class="card">
37 <div class="info">
38 <h3>المشتريات المكتملة</h3>
39 <div class="value" id="completed-sales"></div>
40 </div>
41 <div class="icon"><i class="fa-solid fa-check"></i></div>
42 </div>
43 <div class="card">
44 <div class="info">
45 <h3>المشتريات المعلقة</h3>
46 <div class="value" id="pending-sales"></div>
47 </div>
48 <div class="icon"><i class="fa-solid fa-pause"></i></div>
49 </div>
50 </div>
51
52 <table class="purchase-table">
53 <thead>
54 <tr>
55 <th>المعرف</th>
56 <th>السيارة</th>
57 <th>السعر</th>
58 <th>تاريخ الشراء</th>
59 <th>حالة الشراء</th>
```

الشكل 39 : كود البرمجي للواجهة Purchase Client

- **Payment Manger** : لوحة مدير المدفوعات (Payment Manager) هي وحدة متخصصة تتيح للمسؤولين إدارة وتتبع جميع المعاملات المالية في النظام. توفر أدوات متقدمة لمراقبة عمليات الدفع، الفواتير، والعمليات المالية مع ضمان الأمان والشفافية.

المميزات الأساسية:

- تتبع المعاملات: عرض سجل المدفوعات (الناجحة/الفاشلة/المعلقة)
- إدارة الفواتير: إنشاء الفواتير وتعديلها وتصديرها
- تقارير مالية: تحليلات وإحصائيات حول الإيرادات والمدفوعات
- تكامل مع بوابات الدفع: دعم متعدد للطرق (بطاقات، محافظ إلكترونية، تحويلات بنكية)
- إشعارات: تنبيهات فورية لأي عمليات مشبوهة أو مدفوعات معلقة

الفصل الثالث : التنفيذ

```

1 <!DOCTYPE html>
2 <html lang="ar" dir="rtl">
3 <head>
4   <meta charset="UTF-8" />
5   <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
6   <title>بوابة الدفع</title>
7   <link href="https://fonts.googleapis.com/css2?family=Cairo:wght@400;700&display=swap" rel="stylesheet">
8   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.5.0/css/all.min.css">
9   <link rel="stylesheet" href="../../css/dashboard/payments_manager/dashboard.css"/>
10  <link rel="stylesheet" href="../../css/component/toast.css"/>
11 </head>
12 <body>
13
14   <div class="sidebar">
15     <div>
16       <h2><i class="fas fa-credit-card"></i> بوابة الدفع</h2>
17       <ul>
18         <li class="active"><a href="#"><i class="fas fa-home"></i> الرئيسية</a></li>
19         <li><a href="./payments.html"><i class="fas fa-credit-card"></i> المدفوعات</a></li>
20       </ul>
21     </div>
22     <button class="logout" id="logout-btn"><i class="fas fa-sign-out-alt"></i> تسجيل الخروج</button>
23   </div>
24
25   <div class="main">
26     <div class="dashboard-header">بوابة الدفع</div>
27
28     <div class="cards">
29       <div class="card">
30         <h3>إجمالي المدفوعات</h3>
31         <div class="value" id="total-revenues"></div>
32       </div>
33       <div class="card">
34         <h3>عمليات الدفع المسبقة</h3>
35         <div class="value" id="down-payments"></div>
36       </div>
37       <div class="card">

```

الشكل 40 : كود البرمجي للواجهة Dashboard Clinet

```

payments.html ×
front-end > dashboard > payments_manager > payments.html > ...
 2 <html lang="ar" dir="rtl">
15 <body>
16 <div class="sidebar">
17 <div>
19 <ul>
22 </ul>
23 </div>
24 <button class="logout" id="logout-btn"><i class="fas fa-sign-out-alt"></i> تسجيل الخروج</button>
25 </div>
26
27 <div class="main">
28 <div class="dashboard-header">عمليات البيع</div>
29 <div class="actions-container">
30 <div class="search-container">
31 <input type="text" id="email-search" placeholder="أدخل معرف عملية البيع" />
32 <button id="search-btn" class="btn-search"><i class="fas fa-search"></i></button>
33 </div>
34 </div>
35 <br/>
36 <div class="table-container">
37 <table>
38 <thead>
39 <tr>
40 <th>الرقم</th>
41 <th>العميل</th>
42 <th>السيارة</th>
43 <th>المبلغ</th>
44 <th>التاريخ</th>
45 <th>الحالة</th>
46 <th>الإجراء</th>
47 </tr>
48 </thead>
49 <tbody id="sale-table-body">
50 </tbody>
51 </table>
52 </div>
53 <!--Update payment status dialog-->

```

الشكل 41 : كود البرمجي للواجهة Dashboard Payments

- **Sales Manger** : هي وحدة متكاملة تمكن فرق المبيعات والإدارة من تتبع وتحليل أداء المبيعات بشكل فعال. توفر أدوات متقدمة لإدارة العملاء، متابعة الفرص التجارية، وتحليل البيانات لتحسين الاستراتيجيات.

- **المميزات الأساسية:**

- تتبع المبيعات: عرض سجل الطلبات وحالة كل صفقة
- إدارة العملاء: قاعدة بيانات العملاء مع سجل المشتريات والتفضيلات
- تقارير أداء: تحليلات مبيعات تفصيلية (إيرادات، أهداف، مقارنات زمنية)
- إدارة العروض: إنشاء وتعديل العروض الترويجية وحسابات الخصومات
- تكامل مع أنظمة أخرى: ربط مع بوابات الدفع ولوحة المخزون

```
dashboard.html x inquiries.html sale.html sales.html
front-end > dashboard > sales_manager > dashboard.html > ...
1 <!DOCTYPE html>
2 <html lang="ar" dir="rtl">
3 <head>
4   <meta charset="UTF-8" />
5   <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
6   <title>لوحة مبيعات</title>
7   <link href="https://fonts.googleapis.com/css2?family=Cairo:wght@400;700&display=swap" rel="stylesheet">
8   <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/chart.js"></script>
9   <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/chart.js"></script>
10  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/chart.js"></script>
11  <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
12 </head>
13 <body>
14   <div class="sidebar">
15     <div>
16       <h2><i class="fas fa-car"></i> بوابة المبيعات</h2>
17       <ul>
18         <li class="active"><a href="#"><i class="fas fa-tachometer-alt"></i> لوحة التحكم</a></li>
19         <li><a href="/sales.html"><i class="fas fa-list"></i> قائمة المبيعات</a></li>
20         <li><a href="/inquiries.html"><i class="fas fa-envelope"></i> الاستفسارات</a></li>
21       </ul>
22     </div>
23     <button class="logout" id="logout-btn"><i class="fas fa-sign-out-alt"></i> تسجيل الخروج</button>
24   </div>
25
26   <div class="main">
27     <div class="dashboard-header">لوحة المبيعات</div>
28
29     <div class="cards">
30       <div class="card">
31         <div class="info">
32           <h3>إجمالي المبيعات</h3>
33           <div class="value" id="total-revenue"></div>
34         </div>
35         <div class="icon"><i class="fas fa-dollar-sign"></i></div>
36       </div>
37       <div class="card">
38         <div class="info">
39           <h3>إجمالي المبيعات</h3>
40           <div class="value" id="total-revenue"></div>
41         </div>
42         <div class="icon"><i class="fas fa-dollar-sign"></i></div>
43       </div>
44     </div>
45   </div>
46 </body>
47 </html>
```

الشكل 42 : كود البرمجي للواجهة Dashboard Sales

الفصل الثالث : التنفيذ

```
dashboard.html inquiries.html x sale.html sales.html
front-end > dashboard > sales_manager > inquiries.html > ...
2 <html lang="ar" dir="rtl">
15 <body>
16
17 <div class="sidebar">
18 <div>
19 <h2><i class="fas fa-car"></i>بوابة المبيعات</h2>
20 <ul>
21 <li><a href="./dashboard.html"><i class="fas fa-tachometer-alt"></i>لوحة التحكم</a></li>
22 <li><a href="./sales.html"><i class="fas fa-list"></i>قائمة المبيعات</a></li>
23 <li class="active"><i class="fa-solid fa-message"></i>الاستفسارات</a></li>
24 </ul>
25 </div>
26 <button class="logout" id="logout-btn"><i class="fas fa-sign-out-alt"></i>تسجيل الخروج</button>
27 </div>
28
29 <div class="main">
30 <div class="dashboard-header">الاستفسارات</div>
31 <div class="table-section">
32 <table>
33 <thead>
34 <tr>
35 <th>المعرف</th>
36 <th>الاسم الكامل</th>
37 <th>البريد الالكتروني</th>
38 <th>الإجراء</th>
39 </tr>
40 </thead>
41 <tbody id="inquiry-table-body">
42 </tbody>
43 </table>
44 </div>
45 </div>
46 <!-- Message Response Modal -->
47 <div id="messageModal" class="message-modal">
48 <div class="message-modal-content">
49 <div class="message-header">
50 <h3>الرد على الاستفسار</h3>
51 <button class="close-modal" id="close-btn"><time></button>
```

الشكل 43 : كود البرمجي للواجهة Dashboard inquiries

```
dashboard.html inquiries.html sale.html sales.html x
front-end > dashboard > sales_manager > sales.html > ...
1 <!DOCTYPE html>
2 <html lang="ar" dir="rtl">
3
4 <head>
5 <meta charset="UTF-8" />
6 <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7 <title>قائمة المبيعات</title>
8 <link href="https://fonts.googleapis.com/css2?family=Cairo:wght@400;700&display=swap" rel="stylesheet">
9 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/font-awesome@6.5.0/css/all.min.css">
10 <link rel="stylesheet" href="../../css/dashboard/sales_manager/sales.css" />
11 <link rel="stylesheet" href="../../css/component/toast.css" />
12 </head>
13 <body>
14
15 <div class="sidebar">
16 <div>
17 <h2><i class="fas fa-car"></i>بوابة المبيعات</h2>
18 <ul>
19 <li><a href="./dashboard.html"><i class="fas fa-tachometer-alt"></i>لوحة التحكم</a></li>
20 <li class="active"><a href="#"><i class="fas fa-list"></i>قائمة المبيعات</a></li>
21 <li><i class="fa-solid fa-message"></i><a href="./inquiries.html"></a></li>
22 </ul>
23 </div>
24 <button class="logout" id="logout-btn"><i class="fas fa-sign-out-alt"></i>تسجيل الخروج</button>
25 </div>
26
27 <div class="main">
28 <div class="dashboard-header">قائمة المبيعات</div>
29 <div class="actions-container">
30 <div class="search-container">
31 <input type="text" id="email-search" placeholder="أدخل معرف عملية البيع" />
32 <button id="search-btn" class="btn-search"><i class="fas fa-search"></i></button>
33 </div>
34 </div>
35
36 <div class="table-section">
37 <table>
```

الشكل 44: كود البرمجي للواجهة Dashboard Sale

الفصل الثالث : التنفيذ

```
dashboard.html inquiries.html sale.html x sales.html
front-end > dashboard > sales_manager > sale.html > ...
1 |<!DOCTYPE html>
2 |<html lang="ar" dir="rtl">
3 |
4 |<head>
5 |   <meta charset="UTF-8" />
6 |   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7 |   <title>تفاصيل المبيعات</title>
8 |   <link href="https://fonts.googleapis.com/css2?family=Cairo:wght@400;700&display=swap" rel="stylesheet">
9 |   <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/font-awesome@6.5.0/css/all.min.css">
10 |   <link rel="stylesheet" href="../../css/dashboard/sales_manager/sale_details.css" />
11 |   <link rel="stylesheet" href="../../css/component/toast.css" />
12 |   <link rel="stylesheet" href="../../css/component/spinner.css" />
13 |</head>
14 |
15 |<body>
16 |
17 |   <div class="sidebar">
18 |     <div>
19 |       <h2><i class="fas fa-car"></i>بوابة المبيعات</h2>
20 |       <ul>
21 |         <li><a href="../../dashboard.html"><i class="fas fa-tachometer-alt"></i>لوحة التحكم</a></li>
22 |         <li class="active"><a href="../../sales.html"><i class="fas fa-list"></i>قائمة المبيعات</a></li>
23 |         <li><a href="../../inquiries.html"><i class="fa-solid fa-message"></i>الاستفسارات</a></li>
24 |       </ul>
25 |     </div>
26 |     <button class="logout" id="logout-btn"><i class="fas fa-sign-out-alt"></i>تسجيل الخروج</button>
27 |   </div>
28 |
29 |   <div class="main">
30 |     <div class="dashboard-header">تفاصيل عملية البيع</div>
31 |     <div class="actions-container">
32 |       <a href="../../sales.html" class="back-btn"><i class="fas fa-arrow-right"></i>العودة إلى قائمة المبيعات</a>
33 |     </div>
34 |
35 |     <div class="sale-details-container">
36 |       <div class="sale-header">
37 |         <div class="sale-id">
```

الشكل 45 : كود البرمجي للواجهة Dashboard Sales

5.5.4 الربط بين الواجهة الامامية و الخلفية :

يُعد الربط بين الواجهة الأمامية (Frontend) والخلفية (Backend) حجر الأساس في تطوير التطبيقات الحديثة، حيث يضمن تبادل البيانات بسلاسة بين واجهة المستخدم وأنظمة الخادم. يتم ذلك عادةً عبر واجهات برمجة التطبيقات (APIs) ، والتي تنقل الطلبات وتستقبل الاستجابات بشكل آمن وفعال.

- تجربة مستخدم سلسة : تضمن سرعة استجابة الواجهة ودقة عرض البيانات.
- أمان النظام : تحمي نقاط الاتصال من الاختراقات عبر طبقات مصادقة وتشفير.
- كفاءة التطوير: تسمح لفريق Frontend و Backend بالعمل بشكل متوازٍ مع تحديد أدوار واضحة.

بدون هذا التكامل، يصبح التطبيق غير وظيفي، مما يُظهر أهمية تصميم واجهات اتصال موثوقة وقابلة للتطوير.

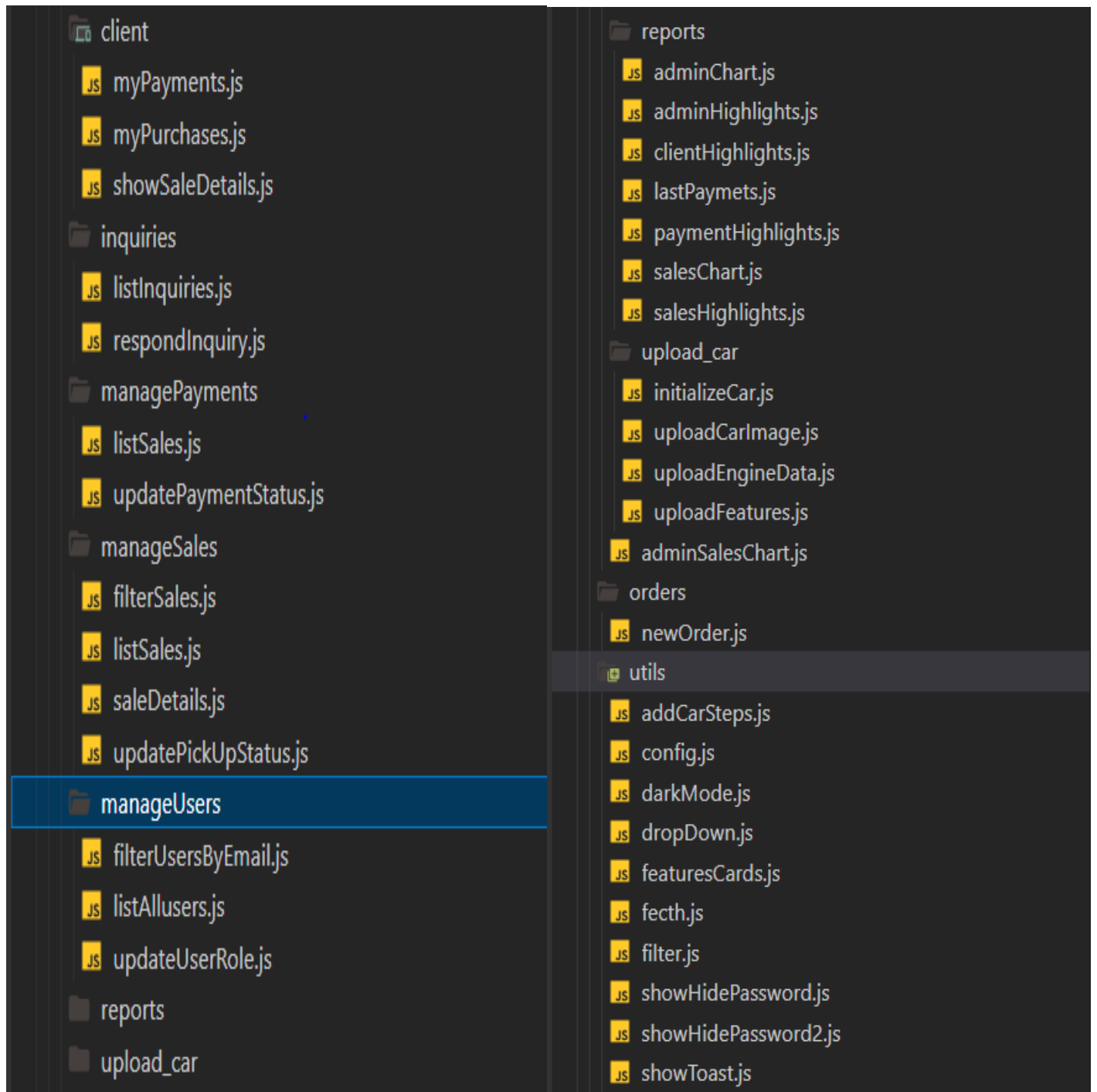
في تطوير الويب، يُعتبر **Fetch API** أداة أساسية مدمجة في متصفحات الويب الحديثة ولغة JavaScript ، تسمح للمطورين بإرسال واستقبال البيانات من الخوادم (Backend) بسهولة دون الحاجة إلى مكتبات خارجية.

شرح بسيط:

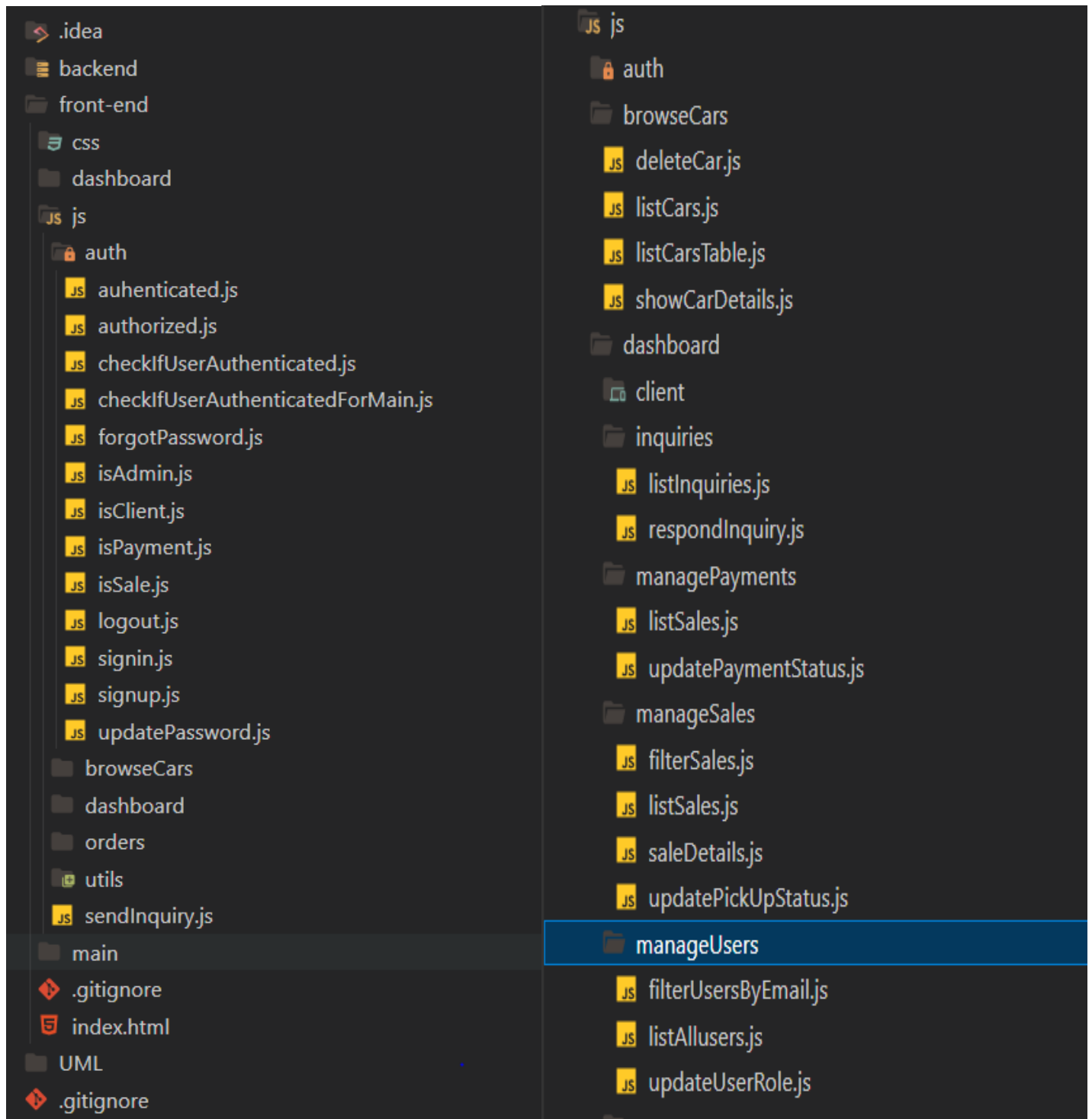
• ما هو **Fetch** ؟

دالة جاهزة في JavaScript تُستخدم لطلب البيانات من الخادم (مثل JSON أو نصوص) عبر شبكة الإنترنت باستخدام **HTTP Requests** (مثل GET أو POST).

• معمارية ملفات الربط بين (Front-end) و (Back-end) :



الشكل 46: معمارية ملفات الربط بين واجهة الامامية و الخلفية



الشكل 47 : معمارية ملفات الربط بين واجهة الامامية و الخلفية

. الشكل المعتمد في الربط بين (Front-end) و (Back-end) :

```
carRoutes.js uploadCarImage.js X
front-end > js > dashboard > upload_car > .js uploadCarImage.js > ...
1 import { config } from "../../utils/config.js";
2 import { showToast } from "../../utils/showToast.js";
Windsurf: Refactor | Explain | Generate JSDoc | X
3 export const uploadCarImage = async (carModelId) => {
4   const imageInput = document.getElementById('image');
5   const formData = new FormData();
6   if(!imageInput.files[0]){
7     showToast("يرجى اختيار صورة للسيارة",error);
8     return null;
9   }
10  formData.append('image',imageInput.files[0]);
11  const token = localStorage.getItem('token');
12  try{
13    const response = await fetch(`${config.API_URL}/api/car/images/${carModelId}`, {
14      method : 'PUT',
15      headers : {
16        'Authorization' : `${token}`
17      },
18      body : formData,
19    });
20    const data = await response.json();
21    if(data.ok){
22      showToast("تم إضافة الصورة بنجاح","success");
23      return data.data.id
24    }else {
25      console.log(data);
26      showToast("حدث خطأ أثناء إضافة الصورة","error");
27      return null;
28    }
29  }catch(error){
30    console.log(error);
31    showToast("حدث خطأ غير متوقع يرجى إعادة المحاولة لاحقاً","error");
32    return null;
33  }
34 }
35 }
```

الشكل 48 : الكود البرمجي الخاص بعملية الربط بين Back-end و Front-end

```
carRoutes.js X fetch.js X
front-end > js > utils > .js fetch.js > [0] handelFetch
1 import {config} from "../config.js";
Windsurf: Refactor | Explain | Generate JSDoc | X
2 export const handelFetch = async (uri,info) => {
3   try{
4     const response = await fetch(`${config.API_URL + uri}`,info);
5     const data = await response.json();
6     return data;
7   }catch(err){
8     return;
9   }
10 }
```

الشكل 49 : الكود البرمجي الخاص بعملية الربط بين Back-end و Front-end

6.4 واجهات البرمجية :

1.6.4 واجهات تسجيل الدخول/تسجيل :

تُمثل واجهات تسجيل الدخول (Login) والتسجيل (Signup) نقطة الدخول الرئيسية للمستخدمين، حيث تتيح إنشاء حسابات جديدة أو الوصول إلى الحسابات الموجودة.

• Signup :

• الشكل 50 : الواجهة البرمجية لعملية انشاء حساب في الموقع

• Login :

• الشكل 51 : الواجهة البرمجية لعملية تسجيل حساب في الموقع

• Forget Password :

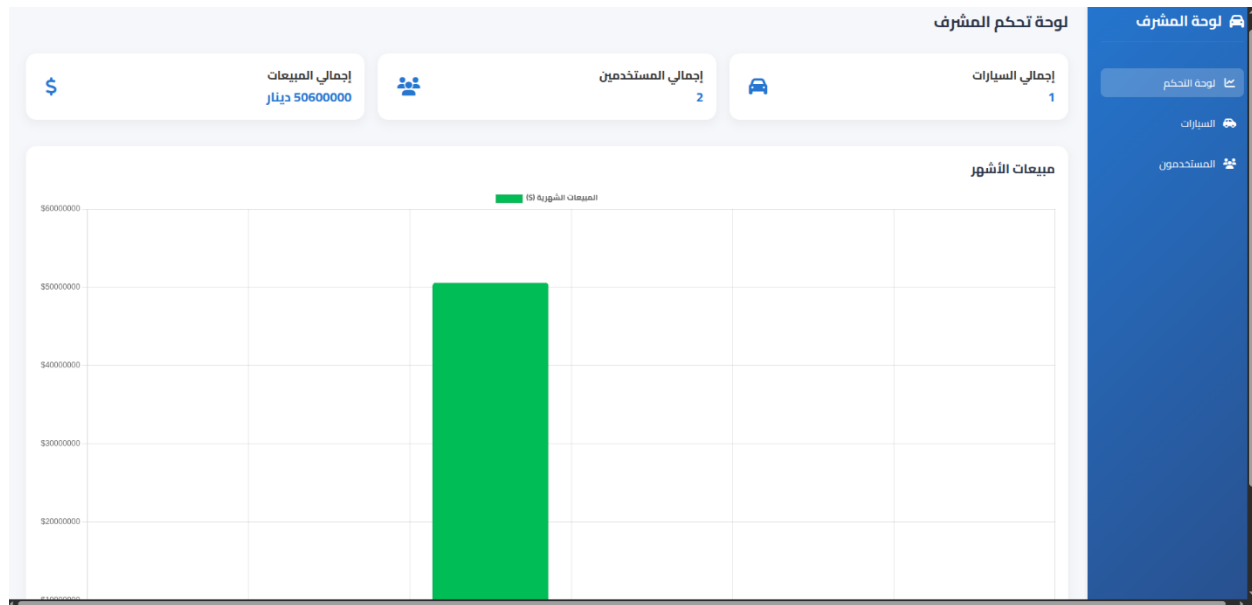
• الشكل 52 : الواجهة البرمجية لعملية استرجاع كلمة المرور

• Reset Password :

• الشكل 53 : الواجهة البرمجية لعملية إعادة تعيين كلمة المرور

2.6.4 واجهات : Dashboard

- **Admin :** واجهة شاملة تمنح المسؤولين تحكماً كاملاً في إدارة النظام وعرض إحصائياته الحيوية.



الشكل 54 : الواجهة البرمجية للوحة التحكم Dashboard

لوحة المشرف

لوحة التحكم

السيارات

المستخدمون

تعديل بيانات السيارة

معلومات أساسية

بالنقر الحثي

حفظ المعلومات الأساسية

التمويل

سنة الصنع

نوع السيارة

سيدان

نوع ناقل الحركة

أوتوماتيك

نوع الدفع

دفع أمامي

السعر

التخفيض (إختياري)

0

الشكل 55 : الواجهة البرمجية لتعديل و اضافة سيارة

الفصل الثالث : التنفيذ

إدارة السيارات

+ إضافة سيارة جديدة

المعرف	الموديل	السنة	السعر	الحالة	العمليات
cma2tu95h0003kv635c8tg5gn	EON23	2023	23000000 دينار	متوفرة	عرض حذف

لوحة المشرف

لوحة التحكم

السيارات

المستخدمون

تسجيل الخروج

. الشكل 56 : الواجهة البرمجية لعرض السيارات المضافة

إدارة المستخدمين

البحث بالبريد الإلكتروني

المعرف	الاسم الأول	اللقب	البريد الإلكتروني	الدور	الإجراءات
cmBq5s2yg0000kvgaW3kpevm2	Oussama	Khelef	khelefoussama2005@gmail.com	مستخدم	تعديل الدور
cma30gta40000kvongvpqbwvt	Oussama	Khelef	oussamakhelef2005@gmail.com	مدير محتويات	تعديل الدور

لوحة المشرف

لوحة التحكم

السيارات

المستخدمون

تسجيل الخروج

. الشكل 57 : الواجهة البرمجية لعرض المستخدمين

الفصل الثالث : التنفيذ

- **Client :** واجهة شخصية تتيح للمستخدمين متابعة تعاملاتهم وطلباتهم وتحديث بياناتهم بسهولة.

The screenshot displays the 'Client Dashboard' (مشتراتي) interface. It features a top navigation bar with three tabs: 'المشتريات المعلقة' (1), 'المشتريات المكتملة' (1), and 'إجمالي المشتريات' (3). Below the tabs is a table with columns: 'المعرف' (ID), 'السيارة' (Car), 'السعر' (Price), 'تاريخ الشراء' (Purchase Date), 'الحالة' (Status), 'المتبقي' (Remaining), and 'العمليات' (Operations). The table lists three transactions. The right sidebar contains a 'بوابة الزبون' (Customer Portal) section with links for 'مشترياتي' (My Purchases) and 'مدفوعاتي' (My Payments), and a 'تسجيل الخروج' (Logout) button at the bottom.

المعرف	السيارة	السعر	تاريخ الشراء	الحالة	المتبقي	العمليات
cma54akg0001kylf8u2kqbm	EON23	دينار 23000000	2025-05-01	غير المكتمل	18400000 دينار	عرض
cma817wz0001kvpq1by0xjam	EON23	دينار 23000000	2025-05-03	مكتمل	0 دينار	عرض
cma488e7k0001kvmmeyt1f9a6	EON23	دينار 23000000	2025-04-30	في انتظار	23000000 دينار	عرض

- الشكل 58 : الواجهة البرمجية للوحة التحكم خاصة بالزبون (مشترياتي)

The screenshot displays the 'Client Dashboard' (مدفوعاتي) interface. It features a top navigation bar with three tabs: 'المشتريات المعلقة' (1), 'المشتريات المكتملة' (1), and 'إجمالي المشتريات' (3). Below the tabs is a table with columns: 'رقم الدفع' (Payment ID), 'التاريخ' (Date), 'المبلغ' (Amount), and 'النوع' (Type). The table lists five transactions. The right sidebar contains a 'بوابة الزبون' (Customer Portal) section with links for 'مشترياتي' (My Purchases) and 'مدفوعاتي' (My Payments), and a 'تسجيل الخروج' (Logout) button at the bottom.

رقم الدفع	التاريخ	المبلغ	النوع
cmaa5vcl0001kvwddvzyrmq	2025-05-04	دينار 46000000	مسيقه
cmaaawgs60001kvn84zqv5yuc	2025-05-05	دينار 184000000	تخمينيه
cmaaawgpai0003kvn8oz3waaou	2025-05-05	دينار 46000000	مسيقه
cmaaawgv9z0005kvn8tsuzy6cc	2025-05-05	دينار 46000000	مسيقه
cmaaawgxdb0007kvn8l3u2slgl	2025-05-05	دينار 184000000	تخمينيه

- الشكل 59 : الواجهة البرمجية للوحة التحكم خاصة بالزبون (مدفوعاتي)

الفصل الثالث : التنفيذ

- **Payment Manger** : نافذة مركزية لمتابعة جميع المعاملات المالية وتتبع حالات الدفع والشحن.

The screenshot displays the 'بوابة الدفع' (Payment Gateway) interface. At the top, there are three summary cards: 'إجمالي المدفوعات' (Total Payments) showing 50600000 Dinar, 'عمليات الدفع المسبقه' (Prepaid Payments) showing 3, and 'عمليات الدفع التكميلية' (Supplementary Payments) showing 2. Below these is a table titled 'عمليات الدفع الأخيره' (Latest Payments) with columns for 'المعرف' (ID), 'العميل' (Customer), 'المبلغ' (Amount), 'نوع الدفعه' (Payment Type), and 'التاريخ' (Date). The table lists five transactions from Oussama Khelef, including prepaid and supplementary payments.

المعرف	العميل	المبلغ	نوع الدفعه	التاريخ
cmaawgjs60001kvn84zvq5yuc	Oussama Khelef	دينار 18400000	تكميلية	2025-05-05
cmaawgpai0003kvn8oz3waaou	Oussama Khelef	دينار 4600000	مسبقه	2025-05-05
cmaawgv9z0005kvn8tsuzzyfcc	Oussama Khelef	دينار 4600000	مسبقه	2025-05-05
cmaawgxdb0007kvn83uzsigr	Oussama Khelef	دينار 18400000	تكميلية	2025-05-05
cmaas5vcl0001kvn8ddvzyrmq	Oussama Khelef	دينار 4600000	مسبقه	2025-05-04

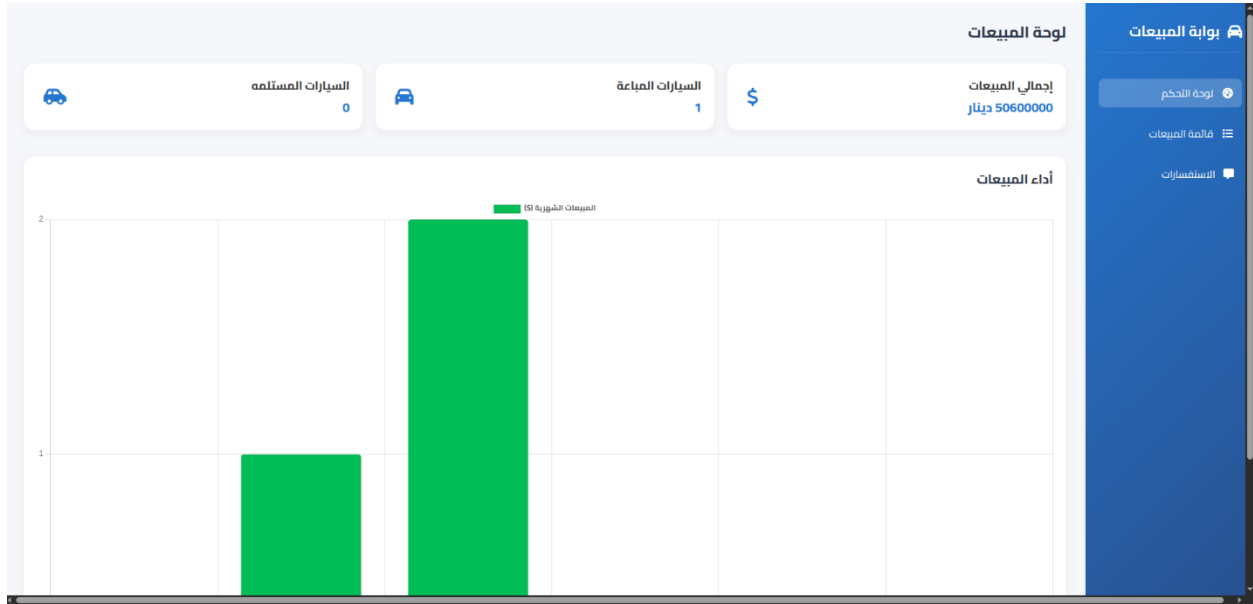
- **الشكل 60** : الواجهة البرمجية لبوابة الدفع (صفحة رئيسية)

This screenshot shows a detailed view of payment transactions. A search bar at the top allows filtering by 'أدخل معرف عملية البيع' (Enter sale transaction ID). The table below lists transactions with columns for 'الرقم' (ID), 'العميل' (Customer), 'السيارة' (Car), 'المبلغ' (Amount), 'التاريخ' (Date), 'الحالة' (Status), and 'الإجراء' (Action). Three transactions are shown, with statuses like 'قيد الانتظار' (Waiting), 'قيد المعالجة' (Processing), and 'مكتملة' (Completed). A modal dialog titled 'إتمام الدفعه المسبقه' (Complete prepaid payment) is overlaid, with 'نعم' (Yes) and 'لا' (No) buttons.

الرقم	العميل	السيارة	المبلغ	التاريخ	الحالة	الإجراء
cma488e7k0001kvmmevt1f9a6	Oussama Khelef	EON23	دينار 23000000	2025-04-30	قيد الانتظار	معالجة الدفع
cma54kalg0001kvfd8u2kqibm	Oussama Khelef	EON23	دينار 23000000	2025-05-01	قيد المعالجة	معالجة الدفع
cma817wbz0001kvpq1by0vjam	Oussama Khelef	EON23	دينار 23000000	2025-05-03	مكتملة	

- **الشكل 61** : الواجهة البرمجية لبوابة الدفع (المدفوعات)

• **Sales Manger** : منصة متكاملة لمتابعة أداء المبيعات وتحليل البيانات واتخاذ القرارات التجارية.



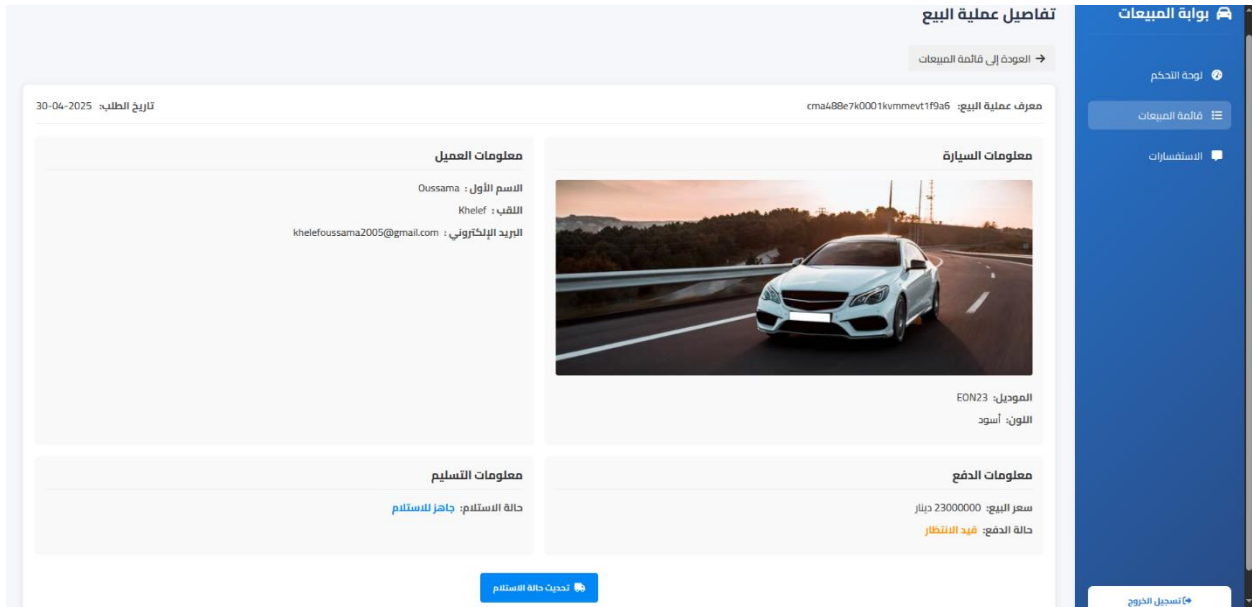
• الشكل 62 : الواجهة البرمجية للوحة التحكم الخاصة بـ (Sales Manger)

The screenshot displays the 'قائمة المبيعات' (Sales List) interface. It features a search bar at the top right with the placeholder text 'أدخل معرف العميل للبحث'. Below the search bar is a table with the following columns: 'المصرف' (Bank), 'العميل' (Customer), 'السيارة' (Car), 'القيمة' (Value), 'التاريخ' (Date), 'الحالة' (Status), 'الاسم' (Name), and 'الإجراءات' (Actions). The table contains three rows of data. On the right side, there is a sidebar with navigation links: 'لوحة التحكم' (Control Panel), 'قائمة المبيعات' (Sales List), and 'البيانات' (Data). At the bottom right, there is a button labeled 'تسجيل الخروج' (Logout).

المصرف	العميل	السيارة	القيمة	التاريخ	الحالة	الاسم	الإجراءات
cma48be7h0001kvmmeyt1f9a6	Oussama Khelef	EON23	230000000 دينار	2025-04-30	مقبول	داغر	عرض
cma54kalg0001kvfd8u2kqbm	Oussama Khelef	EON23	230000000 دينار	2025-05-01	مقبول	داغر	عرض
cma817wbz0001kvpeq1by0jam	Oussama Khelef	EON23	230000000 دينار	2025-05-03	مقبول	داغر	عرض

• الشكل 63 : الواجهة البرمجية لقائمة المبيعات الخاصة بـ (Sales Manger)

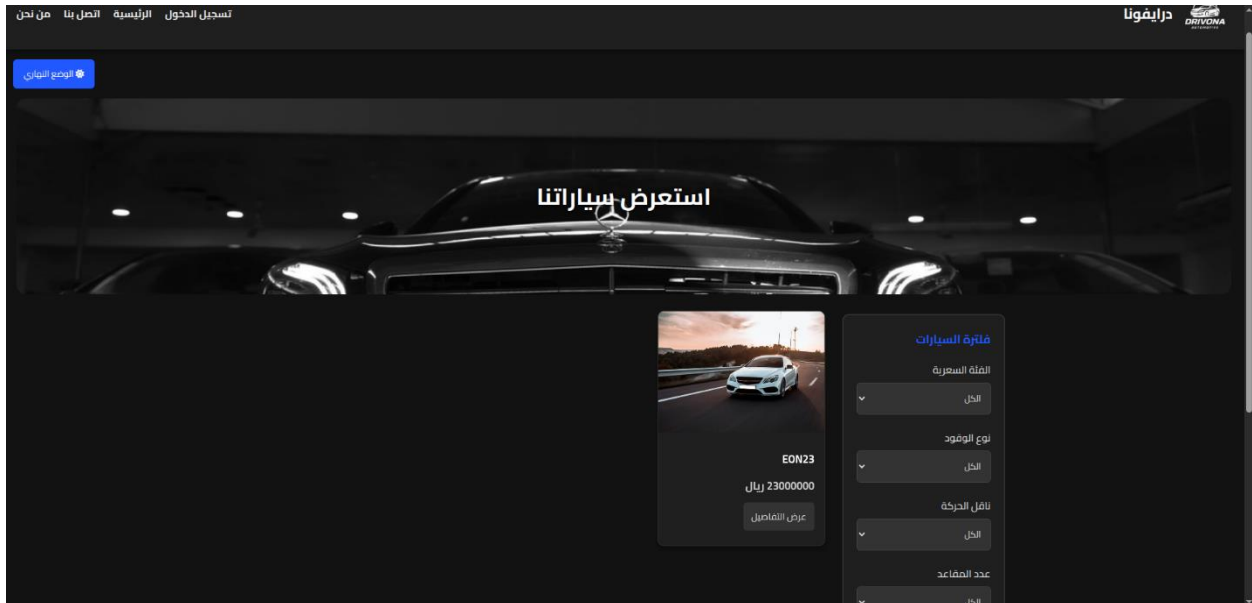
الفصل الثالث : التنفيذ



• الشكل 64 : الواجهة البرمجية لعرض السيارة من قائمة المبيعات

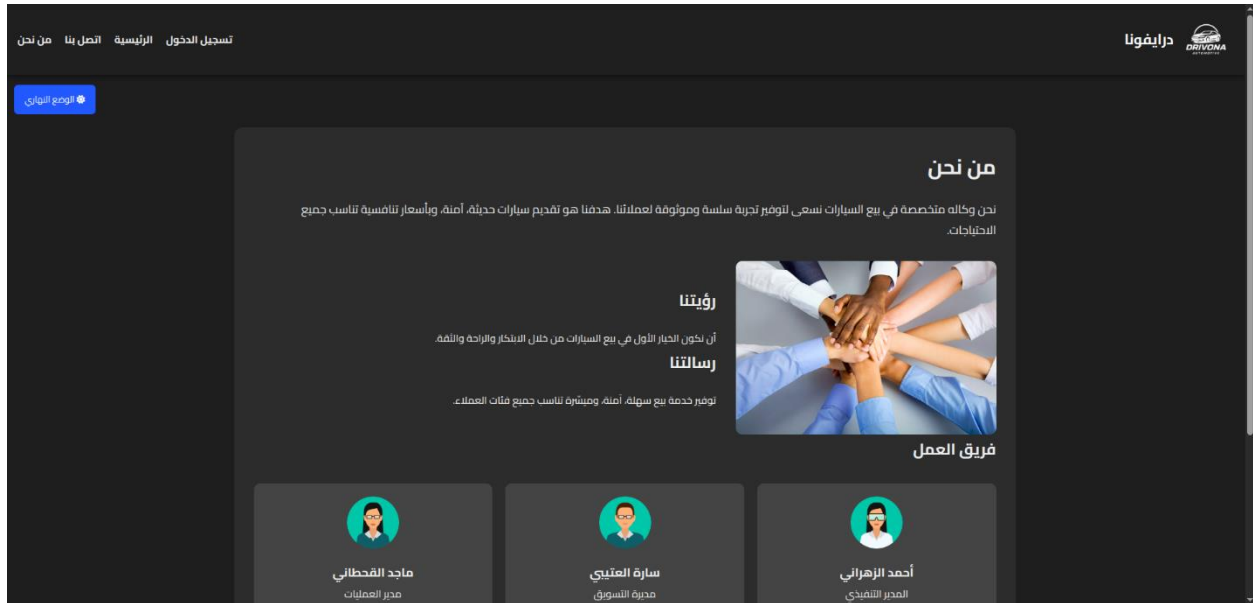
3.6.4 واجهة الصفحة الرئيسية :

• Homepage :



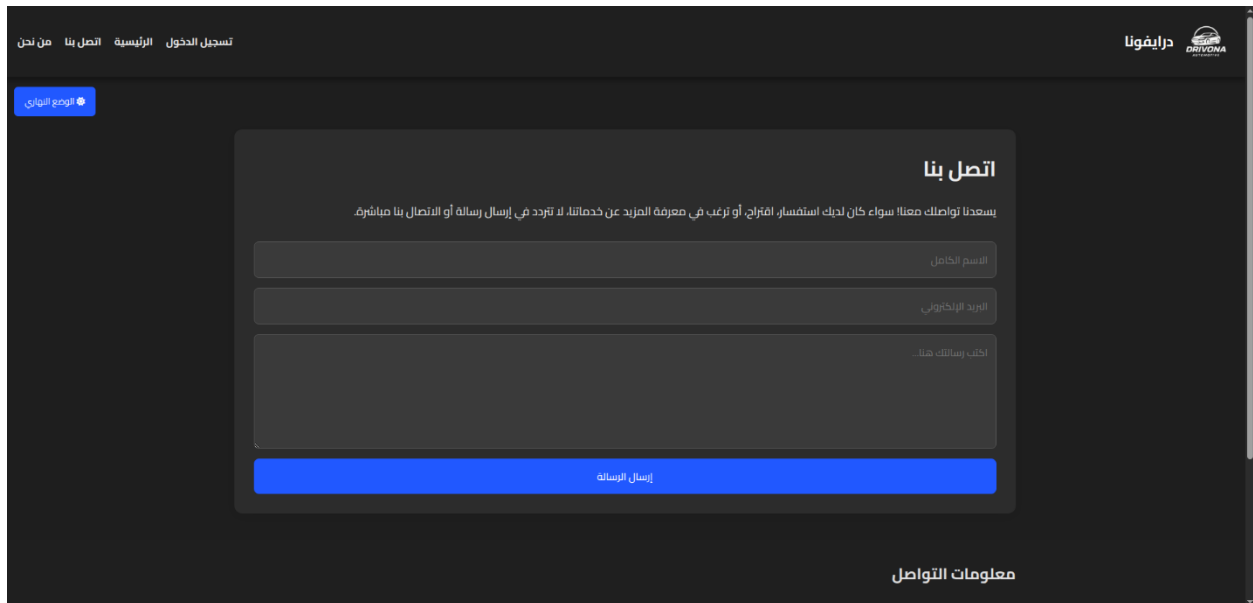
• الشكل 65 : الواجهة البرمجية لاستعراض السيارات في الصفحة الرئيسية للموقع

• About Us :



• الشكل 66 : الواجهة البرمجية 'من نحن' لظهار فريق العمل في موقع Drivona

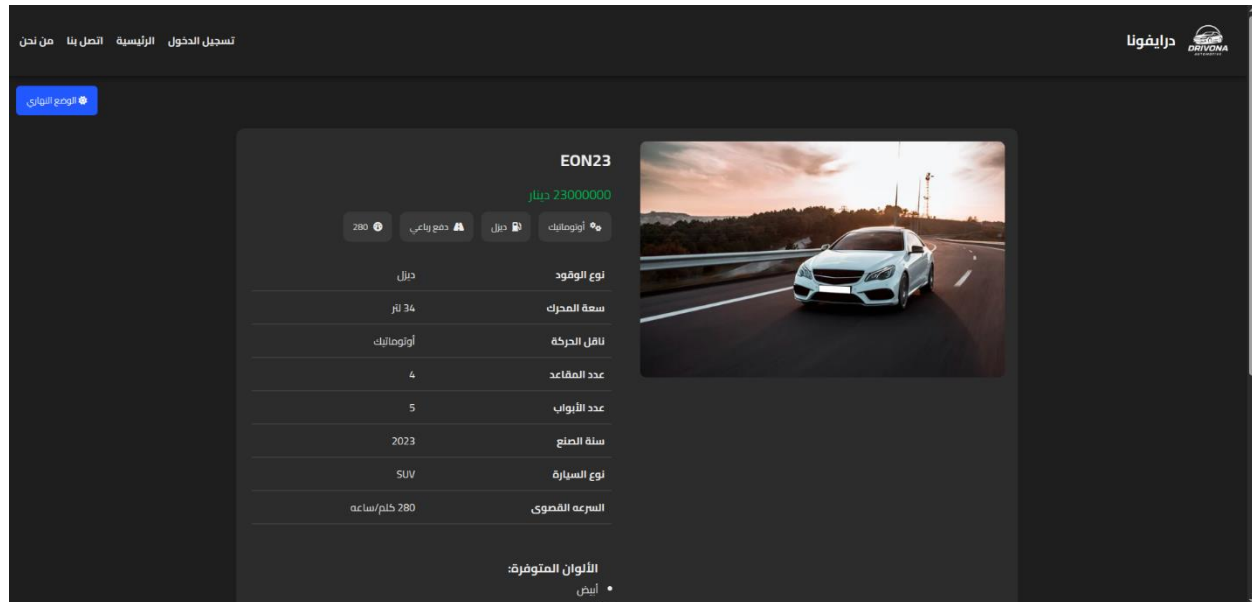
• Contact :



• الشكل 67 : الواجهة البرمجية للتواصل مع الطاقم العمل في موقع Drivona

4.6.4 واجهات عرض السيارة :

• Car Details :



. الشكل 68 : الواجهة البرمجية لجميع تفاصيل السيارة قبل عملية الشراء

الخاتمة

يُمثّل هذا المشروع نظامًا متكاملًا لبيع السيارات عبر الإنترنت، يجمع بين سهولة الاستخدام والقوة التقنية، حيث يوفر واجهات مخصصة لكل من:

- العملاء (لتصفح السيارات وإتمام عمليات الشرك)
- المشرفين وفرق المبيعات والمالية (لإدارة العمليات وتحليل البيانات)

تم تصميم النظام ليكون:

- ✓ آمنًا (باستخدام طبقات مصادقة متعددة)
- ✓ مرناً (يدعم الدفع الإلكتروني السهل)
- ✓ قابلاً للتطوير (بفضل بنيته المعيارية واستخدام أحدث التقنيات)

بدمج هذه العناصر، يُقدّم الموقع تجربة شاملة تلبي احتياجات جميع الأطراف، من العملاء إلى الموظفين، مع ضمان كفاءة العمليات وشفافية المعاملات.

S

المصادر

- <https://www.typescriptlang.org/docs/> : TypeScript
- <https://supabase.com> : Supabase
- <https://expressjs.com> : Express.js
- <https://nodejs.org/docs/latest/api/> : NodeJS
- <https://app.diagrams.net> : Draw.IO UML Creator
- <https://www.kaggle.com> : Kaggle To obtain data about commercial
- <https://railway.com> : Railway
- <https://resend.com> : Resend

