**خطوات العمل على مشروع موقع بيع وتأجير السيارات**

**1- تجهيز بيئة العمل  
- اختيار الأدوات المناسبة**

قبل البدء في تنفيذ المشروع، كان من الضروري اختيار الأدوات المناسبة لتطوير الواجهة الأمامية للموقع (Front-End) ومن أهم الأدوات التي تم استخدامها:

**Visual Studio Code** : بيئة تطوير متكاملة (IDE) لكتابة وتحرير الأكواد البرمجية بسهولة.

**Git** : نظام تحكم في الإصدارات يسمح بحفظ التعديلات وتتبعها.

**Cmder** : واجهة سطر أوامر متقدمة للتعامل مع Git وأوامر النظام.

**GitHub** : منصة لاستضافة الكود ومشاركته مع الفريق أو لعرض المشروع النهائي.

**متصفح الويب (Google Chrome, Firefox, Edge)**: لاختبار عمل الموقع والتأكد من توافقه مع المتصفحات المختلفة.

**- تنزيل البرامج المطلوبة وتثبيتها**

تم تنزيل وتثبيت جميع الأدوات السابقة على الجهاز، بالإضافة إلى ضبط الإعدادات اللازمة لكل منها لضمان سهولة وسلاسة العمل.

**2- إنشاء هيكل المشروع وملفاته الأساسية**

**- إنشاء الملفات الأساسية**

تم إنشاء مجلد المشروع، ثم إضافة الملفات الأساسية المطلوبة لتطوير الموقع:

index.html : ملف يحتوي على الهيكل الأساسي للصفحة الرئيسية باستخدام **HTML**

styles.css : ملف مسؤول عن تصميم الموقع وإضافة التنسيقات باستخدام **CSS**

main.js : ملف مخصص لجعل الموقع تفاعليًا باستخدام **JavaScript**

**- تقسيم الملفات داخل المشروع**

لتنظيم المشروع بشكل أفضل، تم إنشاء مجلدات فرعية لتخزين الصور والملفات الأخرى، كما يلي:

Assets : يحتوي على جميع الموارد مثل الصور والخطوط.

Images : يحتوي على صور السيارات المعروضة في الموقع.

**3- إدارة المشروع باستخدام Git و GitHub**

**- إنشاء مستودع GitHub وربط المشروع به**

بعد تجهيز البيئة، قمنا بإنشاء مستودع جديد على GitHub لرفع المشروع ومشاركته. وللقيام بذلك، تم اتباع الخطوات التالية:

- إنشاء حساب على GitHub

- إنشاء مستودع (Repository) جديد وإعطاؤه اسمًا مناسبًا.

- نسخ رابط المستودع لاستخدامه لاحقًا في ربط المشروع  
  
- بمجرد الانتهاء من إعداد الملفات الأساسية، تم تنفيذ الأوامر التالية لحفظ العمل في Git

في سطر الأوامر (Cmder)، بعد اختيار ملف المشروع من خلال الـTerminal

git init

- إضافة جميع الملفات إلى Git

git add \*

حفظ التعديلات مع كتابة رسالة توضيحية:

git commit -m "first commit"

git remote add origin https://github.com/OElmraghy/Graduation-Project.git

هذه الأوامر تقوم بتهيئة المشروع كمستودع Git وربطه بمستودع GitHub

بعدها نقوم برفع المشروع علي Github عن طريق الامر  
git push -u origin main

**4- تصميم الشكل المبدئي للموقع**

**- استخدام أدوات التصميم**

قبل البدء في كتابة الكود، كان من المهم تحديد الشكل العام للموقع من خلال تصميم واجهة المستخدم باستخدام أدوات مثل:

**Figma** لإنشاء التصميمات التفاعلية قبل تحويلها إلى كود.

**- تحديد الألوان والخطوط والعناصر البصرية**

تم تحديد لوحة ألوان مناسبة تعبر عن احترافية الموقع وسهولة الاستخدام. كما تم اختيار خطوط حديثة ومتناسقة مع الهوية البصرية للمشروع.

**- إنشاء Wireframe للموقع**

تم رسم نموذج تخطيطي لكل صفحة من صفحات الموقع، يتضمن:

الصفحة الرئيسية.

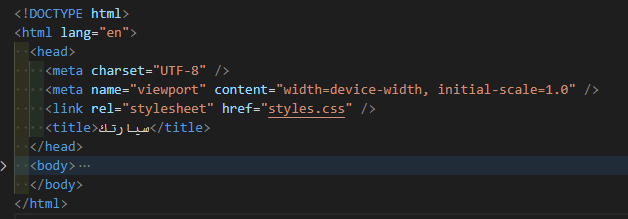
صفحة عرض السيارات.

صفحة تفاصيل السيارة.

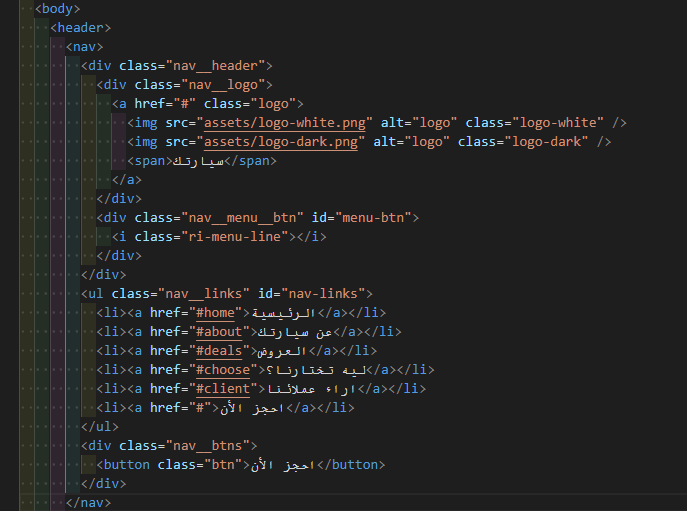
صفحة التواصل.

**5- تنفيذ التصميم باستخدام HTML, CSS, JavaScript**

**- كتابة كود HTML**

تم إنشاء الهيكل الأساسي للموقع باستخدام **HTML5**، مع التركيز على تنظيم العناصر داخل علامات مناسبة او الـ HTML Semantic Elements  
بدأنا بأنشاء البيئة الأساسية لعمل الموقع :-  


- تم ربط ملف التنسيقات عن طريق السطر :   
<link rel="stylesheet" href="styles.css" />

ثم كتبنا الكود لإنشاء الـ Header للموقع وفيه اللوجو للموقع تم انشاءه باستخدام الذكاء الاصطناعي  
واسم الموقع "سيارتي"  
وفيه ازرار الانتقال للأقسام التالية :-  
- الصفحه الرئيسية  
- عن سيارتك  
- العروض  
- ليه تختارنا؟  
- اراء عملاءنا  
- زر احجز الأن  
  
وتم كتابة الكود والبنية الأساسية بالشكل التالي :-  
  
  
  
  
تم مراعاة التسمية الصحيحة للـ classes والـ id لاستخدامها لاختيار العناصر في ملف الـ css  
  
### شرح الكود خطوة بخطوة:

1. \*\*إنشاء عنصر التنقل الرئيسي (Navigation Bar):\*\*

- يتم إنشاء عنصر `<nav>` لاحتواء شريط التنقل الخاص بالموقع.

- شريط التنقل يُنظّم الروابط والأزرار الخاصة بتصفح الموقع.

---

2. \*\*العنوان الرئيسي لشريط التنقل:\*\*

- يتم استخدام عنصر `div` بفئة `nav\_\_header` لإنشاء الجزء العلوي من شريط التنقل.

- يتضمن قسم العنوان:

- \*\*الشعار (Logo):\*\*

- يتم إنشاء شعار داخل `<div class="nav\_\_logo">` يحتوي على رابط باستخدام `<a>` يوجه المستخدم للنقر على الشعار.

- الصور الخاصة بالشعار: يتم إدراج صورتين (شعار أبيض وشعار داكن) مع النص "سيارتك" كجزء من الشعار.

- يُستخدم النص البديل (alt) لتعريف الصور إذا لم تُعرض.

---

3. \*\*زر القائمة (Menu Button):\*\*

- يتم إضافة زر باستخدام `<div>` بفئة `nav\_\_menu\_\_btn`.

- يحتوي الزر على أيقونة تُستخدم لتفتح القائمة (عبر رمز CSS `ri-menu-line`).

---

4. \*\*قائمة الروابط (Navigation Links):\*\*

- يتم إنشاء قائمة غير مرتبة `<ul>` بفئة `nav\_\_links` تضم روابط التنقل:

- \*\*كل رابط:\*\* يتم إنشاؤه باستخدام `<li>` يحتوي على رابط `<a>` يوجه المستخدم إلى القسم المطلوب داخل الصفحة عبر `id`.

- أمثلة على الروابط:

- "الرئيسية" تفتح القسم `#home`.

- "عن سيارتك" تفتح القسم `#about`.

- "العروض" تفتح القسم `#deals`.

- "احجز الأن" هو رابط للحجز.

---

5. \*\*زر الحجز (Booking Button):\*\*

- يتم إضافة زر للحجز داخل `<div class="nav\_\_btns">`.

- الزر يُستخدم لجعل عملية الحجز سهلة وواضحة.

---

6. \*\*التنسيق العام:\*\*

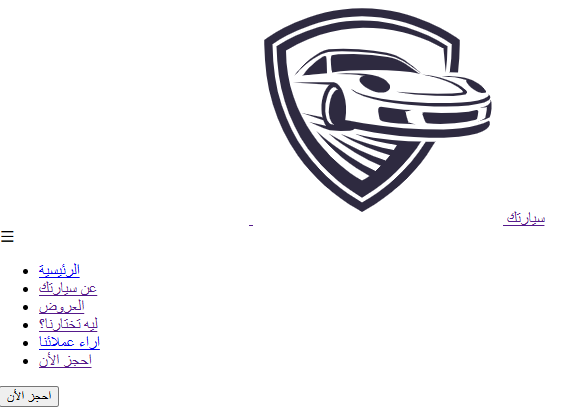
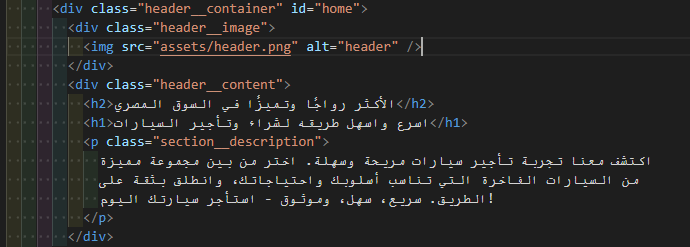
- يعتمد التنسيق على فئات CSS مثل `nav\_\_header`, `nav\_\_logo`, `nav\_\_links` لتحديد مواقع العناصر وتنسيقها.

- الشعار، الأزرار، والروابط مُنظمة بطريقة متناسقة لدعم تجربة المستخدم.  
  
### الهدف:

هذا الكود يُنظّم شريط التنقل في الصفحة لتمكين المستخدمين من التنقل بسهولة بين أقسام الموقع. التصميم يُركز على الجمالية والوضوح، مع إبراز الشعار وزر الحجز.

### المخرجات:

شريط التنقل سيكون واضحًا وسهل الاستخدام للمستخدمين، مما يعزز التصفح السريع والتفاعل مع الموقع.  
  
- تم أيضا استخدام مكتبه ايقونات Remixicon  
عن طريق ربطها بالـ html من خلال التعليمات المتاحة علي الموقع الرسمي للمكتبة  
وهو :   
 <link href="https://cdn.jsdelivr.net/npm/remixicon@4.5.0/fonts/remixicon.css"rel="stylesheet"/>

لأستخدمها في الموقع ومنها أيقونه القائمة لتظهر في حالة استخدام الموقع علي شاشة صغيرة كالموبايل  
ويظهر شكل البنية الان علي الموقع بهذا الشكل   
  
- بدون أي تنسيقات بعد .   
  
- بعدها كتابة بنية الصفحه الرئيسية للموقع , وتم اختيار صور سيارات من موقع PurePNG  
  
  
  
### شرح الكود خطوة بخطوة:

1. \*\*إنشاء الحاوية الرئيسية (Header Container):\*\*

- يتم إنشاء عنصر `div` بالفئة `header\_\_container` لإعداد الحاوية الرئيسية لقسم الهيدر (Header).

- يتم إضافة المعرف (ID) `home` لتمييز القسم بسهولة وتحديده في التنقل داخل الصفحة.

---

2. \*\*إضافة صورة الهيدر:\*\*

- داخل الحاوية، يتم إنشاء عنصر `div` بفئة `header\_\_image` لاحتواء صورة الهيدر.

- يتم إدراج الصورة باستخدام وسم `<img>`:

- \*\*src:\*\* يُحدد مصدر الصورة الموجود في مجلد `assets` باسم `header.png`.

- \*\*alt:\*\* نص بديل يُظهر وصفًا للصورة في حالة عدم تحميلها.

---

3. \*\*إضافة المحتوى النصي (Header Content):\*\*

- يتم إنشاء عنصر `div` آخر بالفئة `header\_\_content` لإضافة النصوص المتعلقة بهذا القسم.

---

4. \*\*العناوين الرئيسية:\*\*

- \*\*عنوان فرعي (`<h2>`):\*\*

- يحتوي النص "الأكثر رواجًا وتميزًا في السوق المصري"، لإبراز التميز والمكانة.

- \*\*عنوان رئيسي (`<h1>`):\*\*

- يحتوي النص "اسرع واسهل طريقة لشراء وتأجير السيارات"، لجذب الانتباه إلى فكرة الخدمة الأساسية.

---

5. \*\*الوصف النصي:\*\*

- يتم استخدام وسم الفقرة `<p>` مع الفئة `section\_\_description` لإضافة نص وصفي جذّاب:

- يوضح التجربة التي توفرها الخدمة: السهولة، الراحة، والثقة.

- يدعو المستخدمين إلى استئجار السيارات بطريقة مريحة.

---

### الهدف:

هذا القسم مُصمم ليكون الجزء التقديمي للموقع (Header Section)، حيث يعرض الفكرة الرئيسية للخدمة بشكل مباشر وجذاب باستخدام:

- صورة تعبر عن الخدمة.

- نصوص تسويقية قصيرة وفعّالة.

- تصميم منظم يجمع بين الجمالية وسهولة القراءة.

### المخرجات:

عند عرض هذا القسم على الصفحة، سيُقدم تجربة أولى قوية تجذب الزوار وتشجعهم على اكتشاف باقي أجزاء الموقع.  
ويكون شكل الصفحه الرئيسية بدون أي تنسيقات كالتالي:-  


- بعدها بدأنا فعمل صفحه كيف نعمل او عن سيارتك :- **A screenshot of a computer program

AI-generated content may be incorrect.**1. \*\*إنشاء قسم خاص بالمحتوى:\*\*

- الكود يبدأ بـ `<section>` ويُعطي القسم هوية (ID) تُسمى `about` لتحديده بسهولة.

- يُضاف إلى القسم تنسيق باستخدام الفئة (class) `section\_\_container` و`about\_\_container`.

2. \*\*العنوان الرئيسي للقسم:\*\*

- الوسم `<h2>` يُستخدم لإضافة عنوان رئيسي بعنوان "كيف نعمل".

- الفئة `section\_\_header` تُحدد تنسيقًا خاصًا للعناوين.

3. \*\*وصف نصي للقسم:\*\*

- الوسم `<p>` يحتوي على وصف موجز يُشرح سهولة عملية تأجير السيارة.

- الفئة `section\_\_description` تُستخدم لإضافة تنسيق مخصص للنصوص الوصفية.

4. \*\*إنشاء شبكة (Grid) لتنظيم المحتوى:\*\*

- يتم استخدام عنصر `<div>` بفئة `about\_\_grid` لإنشاء شبكة تحتوي على بطاقات (Cards) لعرض خطوات العملية.

5. \*\*البطاقة الأولى – اختيار الموقع:\*\*

- يتم إنشاء بطاقة باستخدام `<div class="about\_\_card">`.

- تحتوي البطاقة على:

- رمز أو أيقونة باستخدام وسم `<span>` وعلامة `<i>` (باستخدام فئة CSS خاصة بالأيقونات `ri-map-pin-2-fill`).

- عنوان البطاقة باستخدام `<h4>` بعنوان "اختر المكان".

- نص وصفي يُشرح اختيار مواقع الاستلام.

6. \*\*البطاقة الثانية – تحديد تاريخ الاستلام:\*\*

- نفس الهيكل المستخدم في البطاقة الأولى.

- النصوص تُوضح أهمية تحديد التاريخ والوقت لاستلام السيارة.

- تُستخدم أيقونة مختلفة بفئة `ri-calendar-event-fill`.

7. \*\*البطاقة الثالثة – إتمام عملية الحجز:\*\*

- تحتوي البطاقة على:

- أيقونة خاصة بالسيارات باستخدام الفئة `ri-roadster-fill`.

- عنوان "احجز سيارتك".

- نص وصفي يُؤكد سهولة إتمام الحجز ببضع خطوات فقط.

8. \*\*التنسيق العام:\*\*

- يتم التحكم في تصميم العناصر باستخدام الفئات (CSS Classes) المخصصة مثل `section\_\_container`, `about\_\_card`, `about\_\_grid`.

- الشبكة (Grid) تُنظّم البطاقات الثلاثة بطريقة مرنة ومتصلة.

### الهدف:

هذا القسم يُوضح خطوات تأجير السيارة بطريقة مرنة ومنظمة باستخدام تصميم الشبكة والبنية النصية الواضحة. يمكن للمستخدمين بسهولة فهم العمليات الثلاثة الأساسية: اختيار الموقع، تحديد التاريخ، وإتمام الحجز.  
  
  
  
  
  
  
  
  
 **- إضافة التنسيقات باستخدام CSS**

تم تحسين مظهر الموقع باستخدام **CSS3**، مع تطبيق تقنيات مثل:

**Flexbox** و **Grid** لتوزيع العناصر بشكل مرن ومتناسق.

**Media Queries** لجعل الموقع متجاوبًا مع الشاشات المختلفة.

**- إضافة التأثيرات والتفاعلات باستخدام JavaScript**

تمت إضافة ميزات تفاعلية مثل:

عرض تفاصيل السيارة عند النقر عليها.

**6- اختبار وتحسين أداء الموقع**

**- اختبار توافق الموقع مع المتصفحات**

تم اختبار الموقع على عدة متصفحات لضمان عمله بكفاءة على:

**Google Chrome**

**Mozilla Firefox**

**Microsoft Edge**

**Safari**

**- تحسين سرعة تحميل الموقع**

لضمان أداء سريع، تم تطبيق:

ضغط الصور لتقليل حجمها دون التأثير على الجودة.

استخدام **Lazy Loading** لتحميل الصور عند الحاجة فقط.

**7- نشر الموقع واختباره مع المستخدمين**

**- رفع المشروع على GitHub Pages**

تم نشر الموقع عبر **GitHub Pages** باتباع الخطوات:

فتح إعدادات المستودع على GitHub

تفعيل خيار **GitHub Pages**

تحديد الفرع الرئيسي لعرض الموقع مباشرة.

**- اختبار الموقع مع المستخدمين وأجهزة مختلفة**

بعد النشر، تمت تجربة الموقع على أجهزة مختلفة وجمع ملاحظات المستخدمين لإجراء التحسينات النهائية.

**8 - كتابة التحديات التي واجهت الفريق**

تم توثيق المشاكل التي تمت مواجهتها أثناء تطوير الموقع، مثل:

* توافق التصميم مع جميع الأجهزة.
* تحسين سرعة تحميل الصفحات.
* جعل التفاعلات سلسة وسريعة.