



وزارة التعليم العالي

جامعة إقليم سبا

كلية تكنولوجيا المعلومات وعلوم الحاسوب

قسم: تكنولوجيا المعلومات والاتصالات

تصميم وتنفيذ نظام تتبع المركبات باستخدام GPS & GSM

عبر واجهة ويب

إعداد الطالب :

عبد الحكيم فوزي علي محمد حمود

أبو بكر عبد الجبار عبد الله علي سعد

مصطفى خالد صالح صومل

إشراف :

د. أسامة سيف

هذا المشروع مقدم كجزء من متطلبات نيل درجة البكالوريوس من كلية تكنولوجيا

المعلومات وعلوم الحاسوب

قسم تكنولوجيا المعلومات والاتصالات، جامعة إقليم سبا

2024-2025 م

إهداء

بدأنا بهمة وواجهنا أكثر من هم وتخطينا الكثير من الصعوباتوها نحن اليوم نقطف ثمر الصبر بين دفتي هذا المشروع المتواضع، وما كان ليتم ذلك لو لا فضل الله وعونه وتوفيقه، فالحمد لله دائماً وأبداً إلى منارة العلم وعلم الصبر، إلى من كانت لسيرته العطرة المليئة بالصبر والجد والمثابرة وعلو الهمة، دور بارز في تثبيتنا وقت التيه والضعف والشتات، إلى النبي الأمي، إمام العلماء والمتعلمين، إلى سيد الخلق رسولنا الكريم صلى الله عليه وسلم، فاللهم جازه عنا خير الجزاء.

إلى العظيمة التي لا تفتر من الدعاء، إلى القلب الحنون، من كانت حاضرة بروحها معنا في كل المراحل، إلى أمهاطنا الحبيبات العظيمات إلى من سعى وتعب وناضل لأجلنا، إلى من علمنا كيف نقف على أقدامنا متحدين أعاصير الرياح وصخب الحياة وقوتها، إلى آبائنا العظاماء الكرماء إلى من حبهم يسري في عروقنا ويلهج بذكرهم فؤادنا، إلى إخواننا وأخواتنا إلى من علمنا حرفاً، إلى من كانوا سندًا لنا، إلى من لهم الفضل بإرشادنا إلى طريق العلم والمعرفة، إلى دكاترتنا الكرام إلى كل من وقف معنا بدعوة، أو كلمة، أو دعم ولو خفي إلى أصدقائنا وزملائنا، الذين شاركونا التعب والاجتهد، وكانوا العون والسد إلى أنفسنا التي أبىت أن تستسلم، وثبتت رغم العقبات، وسعت بإصرار نحو الهدف إلى هذه الرحلة التي صنعت فيها القوة، ونقشت على قلوبنا معنى الإصرار نهدي هذا العمل عربون شكر ووفاء، لكل من كان له أثر في هذه المسيرة العلمية.

شکر و عرفان

الحمد لله، لك الحمد يا ربنا كما ينبغي لجلال وجهك وعظمي سلطانك، فأنت صاحب الفضل، والمن، وال توفيق، والإحسان ثم إلى كلية تكنولوجيا المعلومات وعلوم الحاسوب، وبالأخص قسم تكنولوجيا المعلومات والاتصالات، والدكتورة الأفضل العاملين فيها، نود أن نعبر عن شكرنا العميق واحترامنا الكبير لهذه الكلية العريقة، ولها هذا القسم المتميز من خلال برامجها التعليمية المتميزة والمحدثة في مجال تكنولوجيا المعلومات والاتصالات، تسهم الكلية في صقل مهارات ومعرفات الطلاب، وتمكينهم من التفوق الأكاديمي والمهني.

إن التركيز على البحث العلمي والابتكار يعزز قدرات الطلاب على مواكبة التطورات السريعة في هذا المجال الحيوي أنتم، الدكاترة الأفاضل، تمثّلون مصدر إلهام للطلاب من خلال خبراتكم المتخصصة، توجّهون وترشدون الطلاب، وتشجعونهم على استكشاف إمكاناتهم وتحقيق أهدافهم الأكاديمية والمهنية إن دعمكم وتوجيهكم القييم يساهم في بناء قادة المستقبل في مجال تكنولوجيا المعلومات والاتصالات نود أن نشكركم على التزامكم الدائم بتقديم تعليم عالي الجودة، وتوفير بيئة تعليمية محفزة وداعمة إن مجاهوداتكم وتقانيكם في تطوير المناهج الدراسية، وإثراء المحتوى العلمي، يعزز من تجربة التعلم للطلاب، ويمكنكم من اكتساب المهارات الالزمة لمواجهة التحديات في سوق العمل ولا ننسى أن نتقدم بالشكر الجليل لإدارة الكلية وطاقمها الإداري، الذين لم يدخروا جهداً في تسهيل كل السبل أمامنا، وتوفير المناخ المناسب لإنجاز مشروعنا.

أخيراً، نؤدّي أن نعرب عن امتناننا العميق لكل فرد يعمل في كلية تكنولوجيا المعلومات وعلوم الحاسوب. تعملون كفريق متكامل لتحقيق رؤية الكلية وتميزها في مجال التعليم والبحث العلمي شاكراً لكم على الجهود الجبارة التي تبذلونها يوماً بعد يوم لتعزيز قدرات الطلاب وتحقيق التميز الأكاديمي.

مع خالص الامتنان والتقدير،،،

الملخص

في ظل تزايد معدلات سرقة السيارات في اليمن، وافتقار العديد من المناطق لـ "إنترنت إنترنت مستقرة أو خدمات تتبع لحظي، بُرِزَت الحاجة إلى تصميم نظام محلي يتيح تتبع المركبات بطريقة ذكية وموثوقة ومنخفضة التكلفة. ومن هذا المنطلق، تم تنفيذ هذا المشروع بعنوان:

"تصميم وتنفيذ نظام تتبع المركبات باستخدام GPS و GSM "، والذي يهدف إلى المساهمة في رفع مستوى الأمان ومراقبة المركبات، خصوصاً في البيئات التي تعاني من ضعف البنية التحتية الرقمية.

يعتمد النظام على وحدة تحديد الموضع الجغرافية GPS NEO-M8N-0-10 للحصول على إحداثيات دقيقة، ووحدة اتصال GSM SIM800L لنقل هذه البيانات إلى المستخدم إما عبر الإنترنت أو من خلال SMS، ويتم التحكم بكل عملية المعالجة عبر وحدة ESP32.

وقد تم تصميم النظام للعمل بثلاثة أوضاع تشغيل رئيسية:

1. وضع الرسائل القصيرة (SMS Mode):

يمكن للمستخدم إرسال رسالة تحتوي على الكلمة المفتاحية "LOC" ليقوم النظام بالرد تلقائياً برسالة تتضمن رابطاً لموقع السيارة على خريطة Google.

2. الوضع اللحظي (Real-Time Mode):

يتم إرسال الموقع بشكل دوري إلى خادم ويُبَرَّجُ خاص بالمشروع من خلال شبكة Wi-Fi (https://ajyalyemen.site/car_system/dashbord) حيث يتم عرض الموقع مباشرةً في واجهة تفاعلية للمستخدم، مع دعم إضافة أكثر من سيارة أو مستخدم من خلال لوحة تحكم إدارية مصممة بلغة PHP و MySQL.

3. الوضع غير المتصل (Offline Mode):

في حالة انقطاع الاتصال بالإنترنت، يتم تخزين البيانات داخل مصفوفة من نوع `<std::vector<GPSData>>` في وحدة ESP32. وعند استعادة الاتصال، يتم رفع هذه البيانات تلقائياً إلى الخادم. ومع ذلك، فإن هذا الوضع لم يتم اختباره ميدانياً حتى لحظة إعداد هذا التوثيق.

بدأ تنفيذ النظام مبدئياً باستخدام منصة خارجية هندية مفتوحة المصدر تدعى CircuitDigest Cloud، إلا أن الفريق قرر لاحقاً تطوير منصة خاصة أكثر مرونة وتحكمًا بالبيانات، بعد ملاحظات المشرف الأكاديمي.

تمت برمجة النظام بالكامل بلغة C ++ ، وتضمن إرسال البيانات عبر بروتوكولات SMS و HTTP و JavaScript و PHP ، مع إمكانية عرض الموقع على الخريطة وتصديره. كما تمت تجربة الوضعين SMS و Real-Time SMS ميدانياً بنجاح، وأثبتت النسخة دقتها وفعاليتها.

يُعد المشروع نواة أولية لتطبيقات أوسع مستقبلاً في مجال تتبع المركبات، مثل تحديد المناطق الآمنة (Safe Zones)، التنبؤ عند الخروج من النطاق، وإدارة أسلوب النقل، مما يجعله حلًّا تقنياً واقعياً يناسب الظروف اليمنية ويخدم المجتمع.

جدول المحتويات

المحتويات

2	إهداء
3	شكر وعرفان.....
4	الملخص
12.....	الفصل الأول:
12.....	المقدمة.....
13.....	1.خلفية المشروع:
13.....	1.2 مشكلة المشروع :
13.....	1.3 فكرة المشروع :
14.....	1.4 أهداف المشروع :
14.....	1.5 نطاق المشروع :
14.....	1.6 أهمية المشروع :
15.....	1.7 منهجية العمل:
16.....	الفصل الثاني :
16.....	الخلفية النظرية والدراسات السابقة
17.....	1.2 الدراسة الأولى:
18.....	2.2 الدراسة الثانية:
19.....	2.3 الخلاصة:
19.....	2.4 أنظمة تتبع المركبات التقليدية :
19.....	2.5 نموذج تصميم النظام :
20.....	2.6 مميزات المشروع :
22.....	الفصل الثالث:
22.....	منهجية تنفيذ المشروع.....
23.....	3.1 منهجية:
23.....	3.2 مرحلة التخطيط:
23.....	3.2.1 تحديد الأهداف:

23.....	3.2.2 تحليل المتطلبات:
24.....	3.3 مكونات المشروع المادية :
24.....	3.3.1 وحدة المعالجة الرئيسية في المشروع :
26.....	3.3.2 وحدة GPS neo-m8n-0-10 :
27.....	3.3.3 وحدة GSM SIM800L :
29.....	3.3.4 وحدة الطاقة :
29.....	3.3.5 الأسلك ولوحة التجارب :
29.....	3.3.6 صندوق الحماية (Enclosure) :
30.....	3.3.6 شريحة SIM :
30.....	3.3.7 خادم ويب (Ajyalyemen.site) :
30.....	3.4 آلية العمل حسب أوضاع التشغيل :
30.....	3.4.1 وضع الرسائل القصيرة (SMS) :
30.....	3.4.2 الوضع غير المتصل (Offline Mode) :
31.....	3.4.3 وضع التتبع اللحظي (Real-Time Mode) :
31.....	3.5 تنظيم البرنامج الأساسي (Main Program Structure) :
32.....	ملاحظات إضافية:
34.....	الفصل الرابع:
34.....	التحليل والتصميم.....
35.....	4.1 الجدوى الإجتماعية للمشروع :
35.....	4.2 التحليل :
36.....	4.3 التصميم :
36.....	4.3.1 التصميم المادي (Hardware Design) :
36.....	4.3.2 التصميم المنطقي (Software Design) :
45.....	4.3.3 التصميم واجهة الويب (Web Interface Design) :
50.....	الفصل الخامس:
50.....	التنفيذ المادي والبرمجي
51.....	5.1 مرحلة التنفيذ :
51.....	5.1.1 تنفيذ الأجهزة (Hardware Implementation) :

53.....	(Software Implementation :)	5.1.2
54.....	الفصل السادس:	
54.....	الاختبار والنتائج	
55.....	6 مرحلة الاختبار :	6.1
55.....	6 اختبار الوضع الأول : SMS Mode :	6.1.1
57.....	6 اختبار الوضع الثاني : Offline Mode :	6.1.2
58.....	6 اختبار الوضع الثالث : Real-Time Mode :	6.1.3
62.....	6 اختبار فعالية الوحدات الإلكترونية :	6.1.4
62.....	6 تحليل النتائج ومقارنة أوضاع التشغيل :	6.1.5
63.....	6 الدروس المستفادة من مرحلة الاختبار :	6.1.6
63.....	6.1.7 التقييم العام للنظام :	
64.....	الفصل السابع :	
64.....	الخاتمة والمراجع	
65.....	7.1 الخاتمة العامة للمشروع :	
65.....	7.2 أهم الإنجازات التي تحققت في هذا المشروع :	
66.....	7.3 التحديات التي واجهتنا :	
66.....	7.4 التوصيات والتعديلات المستقبلية :	
70.....	7.4 المراجع: (References)	
71.....	الفصل الثامن :	
71.....	الملاحق	
72.....	8.1 الكود المصدرى (Source Code)	
80.....	8.2 اكواز الموقع الالكتروني (Web Site)	

قائمة الجداول :

51.....	جدول (1) : جدول يوضح القطع المستخدمة وعملها
62.....	جدول 2 :جدول يوضح إختبار فعالية الوحدات الإلكترونية
62.....	جدول 3 : جدول يبين مقارنة بين اوضاع التشغيل الثلاثة
68.....	جدول 4 :جدول يبين تحسين تكامل العتاد

جدول الأشكال

26.....	الشكل رقم (1) يوضح خصائص وحدة ESP32
26.....	الشكل رقم (2) يوضح لوحة التحكم ESP32
26.....	الشكل رقم (3) يوضح وحدة GPS NEO-M8N-0-10
27.....	الشكل رقم (4) يوضح اهم منافذ وحدة GPS NEO-M8N-0-10
27.....	الشكل رقم (5) صورة أخرى لوحدة GSM SIM800L
27.....	الشكل رقم (6) يوضح وحدة GSM SIM800L
28.....	الشكل رقم(7) يوضح منفذ sim لوحدة GSM SIM800L
29.....	الشكل رقم (8) توضح منافذ وحدة GSM SIM8-00L
37.....	الشكل رقم (9) يوضح مخطط Block Diagram
38.....	الشكل رقم (10) يوضح مخطط Circuit Diagram
39.....	الشكل رقم (11) يوضح مخطط تدفق البيانات (Follow Chart)
42.....	الشكل رقم (12) يوضح مخطط النشاط activity diagram
43.....	الشكل رقم (13) يوضح مخطط الحالة (Use Case)
44.....	الشكل رقم (14) يوضح مخطط التسلسل (Sequence Diagram)
45.....	الشكل رقم (15) يوضح صفحة تسجيل الدخول لموقع الويب
46.....	الشكل رقم(16) يوضح لوحة تحكم المدير
46.....	الشكل رقم (17) يوضح واجهة إضافة مركبة
46.....	الشكل رقم(18) يوضح واجهة إضافة مستخدم
47.....	الشكل رقم(19) يوضح صفحة تتبع الموقع الجغرافي
47.....	الشكل رقم (20) يوضح صفحة تتبع الموقع الجغرافي بإستخدام القمر الصناعي
48.....	الشكل رقم (21) يوضح جدول سجل المواقع خطوط الطول والعرض
49.....	الشكل رقم (22) يوضح مخطط ERD لقاعدة البيانات
52.....	الشكل رقم (23) يوضح صورة النظام بعد التركيب
56.....	الشكل رقم (24) يوضح عند النقر والدخول إلى الخريطة
56.....	الشكل رقم (25) يوضح إرسال رسالة LOC والرد برابط
56.....	الشكل رقم (26) يوضح صورة تظهر وصول رسالة loc في وضع SMS
57....	الشكل رقم (27) يوضح لقطة من السيريرال مونيتور توضح انقطاع الاتصال بالانترنت وتفعيل وضع Off line
59.....	الشكل رقم (28) صورة تظهر اتصال شبكة اويفاي وان الخادم يستجيب ويتم عرض الموقع بموقع الويب
59.....	الشكل رقم(29) يوضح صورة تبين تتبع لحظي للمركبة على خريطة الموقع

الشكل رقم (30) يوضح صورة تبين تتبع لحظي للمركبة على خريطة الموقع باستخدام القمر الصناعي	60
الشكل رقم (31) يوضح صورة أخرى تبين تتبع لحظي للمركبة على خريطة الموقع باقمر الصناعي	60
الشكل رقم (32) يوضح صورة أخرى تبين تتبع لحظي للمركبة على خريطة الموقع	61
الشكل رقم (33) يوضح صورة ثلاثة تبين تتبع لحظي للمركبة على خريطة الموقع بالقمر الصناعي.....	61

الفصل الأول:

المقدمة

1.1 خلية المشروع :

تُعد سرقة المركبات من الجرائم الشائعة في العديد من المدن اليمنية، خصوصاً في ظل الأوضاع الأمنية غير المستقرة، وضعف الإمكانيات التكنولوجية المساعدة على الردع أو التعقب. وفي المقابل، تفتقر البلاد إلى أنظمة متقدمة أو منصات موثوقة تتبع السيارات وتحديد موقعها بدقة، سواء في الوقت الحقيقي أو بعد حدوث حادثة السرقة.

لقد ساهمت التكنولوجيا الحديثة في تقديم حلول مبتكرة لمثل هذه التحديات، من خلال تكامل تقنيات تحديد الموضع الجغرافية (GPS) مع أنظمة الاتصال اللاسلكي (GSM/Wi-Fi)، مما يتيح مراقبة حركة المركبة عن بعد واستلام الإحداثيات بشكل مستمر.

1.2 مشكلة المشروع :

مع انتشار سرقات المركبات، باتت أنظمة التتبع ضرورية لحماية المركبات ومراقبتها. غير أن معظم الحلول التجارية المتوفرة تتطلب اتصالاً دائماً بالإنترنت، أو تكون مكلفة ومعقدة في التركيب والتشغيل. وتزداد المشكلة في البيئات الريفية أو المناطق ذات التغطية الضعيفة، حيث يصعب استخدام التطبيقات الذكية أو المنصات السحابية بسبب انعدام الإنترت أو عدم وجود بنية تحتية مناسبة.

تمثل المشكلة الأساسية في غياب نظام تتبع فعال، متعدد الأوضاع، منخفض التكلفة، ويعمل حتى في غياب الإنترت. فالمستخدم يحتاج إلى طريقة بسيطة لمعرفة موقع مركبته:

سواء من خلال رسالة نصية (SMS Mode)،

أو عبر تخزين المعلومات لحين توفر الشبكة (Offline Mode)،

أو من خلال التتبع اللحظي عند توفر الإنترت (Real-time Mode).

يسعى المشروع إلى سد هذه الفجوة من خلال تقديم نظام هجين يتكيف مع مختلف حالات الاتصال، ويوفر تجربة استخدام مرنة وآمنة.

1.3 فكرة المشروع :

يهدف هذا المشروع إلى تصميم وتنفيذ نظام تتبع مركبات منخفض التكلفة، يعمل في مختلف البيئات (حتى في حالة انقطاع الإنترت)، ويتيح للمستخدم تتبع موقع سيارته بعدة طرق، إما عبر الإنترت من خلال واجهة ويب تفاعلية، أو من خلال استقبال رسالة قصيرة تحتوي على رابط موقع المركبة عند الطلب.

1.4 أهداف المشروع :

1. تصميم دائرة إلكترونية تجمع بين وحدات GPS و GSM مع وحدة ESP32.
2. تمكين المستخدم من تتبع المركبة لحظياً من خلال الإنترنت.
3. تفعيل ميزة إرسال الموقع الجغرافي عبر SMS عند انعدام الإنترنت.
4. دعم الوضع غير المتصل (Offline Mode) لتخزين الإحداثيات مؤقتاً ورفعها لاحقاً.
5. بناء واجهة ويب خاصة لعرض بيانات المركبة على الخريطة مباشرة.
6. تجربة النظام في بيئات مختلفة لاختبار دقة الإحداثيات وسرعة التحديث.

1.5 نطاق المشروع :

يركز المشروع على تصميم نظام مدمج لتنبئ السيارات، يتم تشغيله بواسطة وحدة تحكم ESP32، ويوظف وحدة NEO-M8N-0 لتحديد الموقع، ووحدة GSM SIM800L لإرسال واستقبال الرسائل القصيرة. كما يعتمد على واجهة برمجية (API) مخصصة للتفاعل مع موقع ويب شخصي مستضاف على خادم خاص (ajyalyemen.site) صُمم خصيصاً لهذا الغرض.

يوفر النظام الوظائف التالية:

استقبال أمر من المستخدم عبر رسالة نصية تحتوي على الكلمة المفتاحية ("LOC")، والرد بموقع المركبة على هيئة رابط Google Maps.

تخزين المسارات الجغرافية محلياً عند فقدان الاتصال بالإنترنت، وإرسالها تلقائياً عند استعادة الاتصال.

إرسال الموقع تلقائياً إلى الخادم الشخصي عبر الإنترنت عند توفر اتصال (Real-Time Mode).

ولا يشمل المشروع حالياً:

التحكم عن بعد بالمركبة (مثل تشغيل/إيقاف المحرك).

التكامل مع تطبيقات موبايل خارجية.

تحليلات متقدمة للبيانات أو الذكاء الاصطناعي.

1.6 أهمية المشروع :

تتبع أهمية المشروع من كونه يقدم حلولاً عملياً ومرناً لتنبئ المركبات في بيئات تعاني من ضعف الاتصال، كما هو الحال في اليمن. ويتميز المشروع بما يلي:

الاعتماد على مكونات إلكترونية منخفضة التكلفة وسهلة البرمجة. دعم تتبع الموقع عبر الرسائل القصيرة، مما يعني عن الحاجة للإنترنت في الحالات الطارئة. توفر نظام يعمل في ثلاث حالات اتصال (SMS, Offline, Real-Time) سهولة التكامل مستقبلاً مع قواعد بيانات وتحليلات لتوسيع قدرات النظام. يسهم المشروع في تعزيز مفاهيم الأمن الذكي، ويمثل نواة لأنظمة أكبر لإدارة الأسطيل أو حماية المركبات الخاصة.

7.1 منهجية العمل:

تم اتباع منهجية تنفيذية تبدأ بـ تحديد المتطلبات من حيث الأجهزة المطلوبة وطرق الاتصال. تصميم الدائرة الإلكترونية الخاصة بوحدات ESP32 و GPS و GSM. تطوير البرمجيات لقراءة الإحداثيات والتعامل مع الرسائل والإنترنت. إنشاء موقع ويب باستخدام MySQL و PHP لعرض بيانات التتبع. اختبار النظام ميدانياً في بيئات مختلفة وتحليل النتائج. الاستفادة من تجارب منصات سابقة مثل Circuit Digest Cloud لصياغة نموذج أكثر احترافية وتحكمًا.

8.1 محتوى الفصول:

الفصل الثاني: الدراسات السابقة، مع مقارنة بين النظام المطور ومنصات مفتوحة مثل CircuitDigest.

الفصل الثالث: التحليل النظمي للمشروع، باستخدام مخططات Activity و Use Case و Sequence.

الفصل الرابع: التصميم التقني للمكونات البرمجية والعادية وواجهة المستخدم.

الفصل الخامس: خطوات التنفيذ العملي وتوصيل المكونات البرمجية والفيزيائية.

الفصل السادس: اختبار النظام وتحليل النتائج، والتوصيات المستقبلية.

الفصل السابع: خاتمة المشروع، التحديات، والدروس المستفادة.

الفصل الثاني :

الخلفية النظرية والدراسات السابقة

1.2 الدراسة الأولى:

Kadiri, K. O. & Adegoke, O. (2019),

"Design of a GPS/GSM Based Anti-theft Car Tracker System"

هدفت هذه الدراسة إلى تطوير نظام تتبع يعتمد على وحدات GPS و GSM باستخدام وحدة تحكم من نوع ATmega16 ، يتم فيه إرسال الموقع الجغرافي للسيارة للمستخدم عبر الرسائل القصيرة (SMS) عند طلبه. يتم تثبيت النظام في مكان سري داخل المركبة لغرض الحماية من السرقة. يرسل النظام الموقع بصيغة روابط Google Maps ، مما يتيح تحديد موقع السيارة بسهولة.

أوجه التشابه مع مشروعنا:

- استخدام وحدات GPS و GSM لتحديد الموقع وإرساله.
- الاعتماد على الرسائل القصيرة لتوفير الموقع للمستخدم.
- التركيز على خفض التكلفة وسهولة التركيب.

أوجه الاختلاف:

- نظامهم يعتمد فقط على وضع SMS Mode ، في حين أن مشروعنا يدعم ثلاثة أوضاع:
 - كما في دراستهم (SMS Mode)
 - وضع التخزين بدون إنترنت (Offline Mode)
 - وضع التتبع اللحظي (Real-time Mode) عبر Wi-Fi والموقع الإلكتروني.
- دراستهم لا توفر متابعة لحظية أو تخزين للمسارات، بينما مشروعنا يدمج الميزات الثلاث ضمن نظام واحد.

تعليق نقيدي:

الدراسة قدمت حلًا فعالًا لحماية المركبات من السرقة باستخدام تقنيات بسيطة ومنخفضة التكلفة، لكنها تفتقر للتكامل مع واجهات مرئية (Web Interface) أو التعامل مع فقدان الاتصال بالإنترنت. مشروعنا يتفوّق من ناحية الشمولية وتعدد طرق التشغيل ووجود منصة ويب متكاملة.

2.2 الدراسة الثانية:

Morallo, N. T. (2021),

"Vehicle tracker system design based on GSM and GPS interface using Arduino as platform"

تناولت هذه الدراسة تصميم نظام تتبع باستخدام وحدة Arduino Uno ، ووحدة GPS NEO-6M ووحدة SIM900A. يقوم النظام بإرسال إحداثيات المركبة إلى المستخدم عبر رسالة قصيرة ويتم عرض الموقع على خرائط Google. الدراسة ركزت على تقديم نظام منخفض التكلفة وسهل الاستخدام يمكن تشغيله في جميع الظروف الجوية.

أوجه التشابه مع مشروعنا:

- استخدام نفس الوحدات GPS NEO-6M و SIM900A.
- الاعتماد على Arduino كمتحكم وإن كنت تستخدم ESP32 ، فالفكرة قريبة.
- إرسال الإحداثيات عبر SMS بصيغة قابلة للعرض على خرائط Google.

أوجه الاختلاف:

- يقتصر نظامهم على SMS فقط ولا يدعم تتبع مباشر أو تخزين عند انقطاع الاتصال.
- لم يتم تضمين أي واجهة ويب أو تطبيق خارجي.
- مشروعنا أكثر تطوراً من حيث تعدد الأوضاع وتكامل النظام مع واجهة ويب حية.

تعليق نقدي:

تميزت الدراسة ببساطة التصميم وانخفاض التكلفة، لكنها افتقرت إلى دعم الحالات المتقدمة مثل التتبع اللحظي أو معالجة فقدان الإنترن特. مشروعنا يعالج هذه النواقص من خلال دعم التخزين المؤقت وعرض المسارات السابقة عبر واجهة الويب.

2.3 الخلاصة:

من خلال تحليل الدراسات السابقة، يمكننا ملاحظة أن معظم الأنظمة تركز على نمط واحد من التشغيل، خصوصاً نمط SMS Mode إلا أن مشروعنا قدم حلاً أكثر شمولية ومرنة، حيث دمج بين التتبع اللحظي عبر الإنترنت، التخزين عند انقطاع الاتصال، والتفاعل عبر الرسائل القصيرة، مع دعم واجهة ويب تعرض موقع المركبة على الخريطة في الوقت الفعلي. هذا التكامل يجعل مشروعنا مناسباً للتطبيق العملي في البيئات المتعددة ويزيد من قيمته المضافة مقارنة بالدراسات السابقة.

2.4 أنظمة تتبع المركبات التقليدية :

تعتمد أنظمة تتبع المركبات التجارية التقليدية على تقنيات متقدمة مثل أجهزة GPS عالية الدقة، وشبكات الاتصال الخلوي (3G/4G/5G) إلى جانب خوادم سحابية مخصصة لتخزين البيانات وتحليلها، مع واجهات مستخدم احترافية سواء عبر تطبيقات الهاتف أو منصات الويب.

رغم هذه الإمكانيات، إلا أن هناك عدة عوائق تحدّ من استخدام هذه الأنظمة في بعض المناطق، وخاصة في الدول النامية مثل اليمن، ومنها :

ارتفاع التكلفة: تتطلب هذه الأنظمة اشتراكات شهرية ورسوماً مقابل خدمات التخزين السحابي والتحليلات.

الحاجة إلى تغطية إنترنت مستمرة: معظمها لا يعمل بدون اتصال دائم بالشبكة.

تعقيد الإعداد والتشغيل: قد تحتاج إلى دعم فني لإعداد النظام وصيانته.

ضعف قابلية التخصيص: يصعب تعديل هذه الأنظمة لتناسب مع احتياجات محلية خاصة، سواء على مستوى اللغة أو آلية العمل.

لهذا ظهرت الحاجة إلى أنظمة مفتوحة وقابلة للتخصيص، مثل المشروع الحالي، الذي يقدم حلّاً بسيطاً، عملياً، وقابلًا للتنفيذ بتكلفة منخفضة.

2.5 نموذج تصميم النظام :

يرتكز نموذج التصميم المقترن على التكامل بين ثلات وحدات أساسية:

وحدة تحديد الموقع GPSNEO-M8N-0: توفر إحداثيات دقيقة ومحدثة.

وحدة الاتصال GSMSIM800L: ترسل الإحداثيات عند الطلب عبر رسائل SMS.

وحدة التحكم (ESP32): تقوم بقراءة البيانات من وحدة GPS، ومعالجتها، ثم إرسالها عبر الإنترن特 أو الرسائل القصيرة حسب الوضع التشغيلي.

آلية التشغيل حسب كل وضع:

: SMS Mode .1

يستقبل ESP32 رسالة "LOC".

يقرأ الموقع من GPS.

يُرسل رابط Google Maps إلى رقم المستخدم عبر SMS.

: Offline Mode.2

في حال انقطاع الإنترن特، تخزن الإحداثيات محلياً داخل `<vector<GPSData>`. عند استعادة الاتصال، يتم رفع البيانات إلى قاعدة بيانات الموقع.

: Real-time Mode.3

عند توفر اتصال Wi-Fi، يُرسل الموقع فوراً إلى السيرفر عبر واجهة API. تُعرض البيانات مباشرة على لوحة تحكم العميل في واجهة الويب.

2.6 مميزات المشروع :

يتميز النظام المقترن بعدد من الخصائص التي تجعله مناسباً للتطبيق في البيئات المتعددة، وخصوصاً في المناطق التي تعاني من ضعف البنية التحتية الرقمية أو انقطاع الإنترنط.

أولى هذه المميزات هي المرونة التشغيلية، حيث يوفر المشروع ثلاثة أوضاع تشغيل مختلفة هي: وضع الرسائل النصية (SMS Mode)، والوضع غير المتصل (Offline Mode)، ووضع التتبع اللحظي (Real-time Mode). هذا التنويع يتيح للنظام التكيف مع حالات الاتصال المختلفة، ويمنح المستخدم حرية أكبر في الاستخدام.

الميزة الثانية هي عدم الاعتماد الكامل على الإنترنط. يمكن للنظام الاستمرار في أداء مهامه الأساسية (مثل إرسال الموقع عند الطلب) باستخدام الرسائل القصيرة فقط، مما يجعله عملياً في البيئات التي تفتقر إلى اتصال دائم أو مستقر بالشبكة.

ومن أبرز جوانب القوة في المشروع هي واجهة العرض المخصصة، حيث تم تصميم موقع ويب خاص على السيرفر يتيح للمستخدم تسجيل الدخول واستعراض مواقع المركبات على خريطة Google

بشكل مباشر. يدعم الموقع اللغة العربية ويوفر تجربة استخدام بسيطة وسلسة، مع إمكانية تصدير الموقع.

كما يقدم النظام آلية للتخزين المؤقت في حال انقطاع الإنترن特، حيث تُخزن الإحداثيات في الذاكرة المحلية، ويتم رفعها تلقائياً عند استعادة الاتصال دون تدخل يدوي.

ويُعد النظام سهل الاستخدام، حيث لا يتطلب من المستخدم معرفة تقنية متقدمة للتعامل معه، كما أن آلية الطلب عبر رسالة "LOC" تجعل التعامل مع النظام بديهياً.

من الناحية التقنية، يعتمد المشروع على مكونات إلكترونية منخفضة التكلفة وسهلة البرمجة مثل NEO-M8N و SIM800L و ESP32.

أخيراً، فإن المشروع يتمتع بقابلية التوسيعة، حيث يمكن في المستقبل دمج ميزات إضافية مثل تحديد "المنطقة الآمنة" (Geo-Fencing)، إرسال تنبيةات عند الخروج عن المسار، أو تطوير تطبيق موبايل متكامل يعرض الموقع ويرسل تنبيةات مباشرة إلى الهاتف.

الفصل الثالث:

منهجية تنفيذ المشروع

3.1 المنهجية:

تم اتباع المنهج الوصفي التطبيقي في هذا المشروع، حيث بدأ الفريق بدراسة وتحليل أنظمة التتبع الحالية، ثم تحديد المشكلات التي تواجه المستخدم اليمني، خصوصاً في المناطق التي تعاني من ضعف في تغطية الإنترن特. وبناءً على هذا التحليل، تم تصميم النظام ليشمل أوضاع تشغيل متعددة تلائم بيئات الاتصال المختلفة.

اعتمد تنفيذ المشروع على مراحل واضحة تبدأ بالخطيط، تليها مرحلة التصميم، ثم البرمجة والتوصيل العملي، وأخيراً الاختبار والتقييم، مع الاستفادة من أدوات التطوير مفتوحة المصدر.

3.2 مرحلة التخطيط:

3.2.1 تحديد الأهداف:

في هذه المرحلة تم تحديد الأهداف الأساسية للمشروع، والتي شملت:
تصميم نظام تتبع مركبات يمكنه العمل دون إنترنت.
دعم التتبع اللحظي من خلال واجهة ويب شخصية.

تمكين المستخدم من طلب موقع المركبة عبر SMS واستلام رابط خرائط.
تقديم بديل منخفض التكلفة لأنظمة التجارية المعقدة.

3.2.2 تحليل المتطلبات:

تم تحليل متطلبات المشروع إلى فئتين رئيسيتين:
أولاً: المتطلبات البرمجية:

تطوير كود برمجي بلغة C++ يعمل على متحكم ESP32.
التعامل مع وحدتي GPS و GSM وتحليل البيانات المستلمة.
استخدام مكتبة HTTPClient لرفع البيانات إلى السيرفر.

إنشاء API بلغة PHP لاستقبال البيانات وتخزينها في MySQL.
تصميم واجهة ويب لعرض موقع المركبة على الخريطة لحظياً.

ثانياً: المتطلبات العتادية:

وحدة ESP32 باعتبارها المتحكم الرئيسي.

وحدة GPS من نوع NEO-M8N-0-10 لقراءة الإحداثيات بدقة.

وحدة GSM SIM800L لإرسال واستقبال الرسائل النصية.

مصدر طاقة مستقر (بطارية 12V).

كابلات وأسلاك توصيل، مع إمكانية استخدام Breadboard في التجميع الأولي.

3.3 مكونات المشروع المادية :

يعتمد النظام المقترن على مجموعة من الوحدات الإلكترونية الأساسية التي تعمل بتكامل تام لتنفيذ عمليات التتبع والإرسال. وفيما يلي شرح مفصل لكل مكون:

3.3.1 وحدة المعالجة الرئيسية في المشروع :

ESP32

وهي وحدة متقدمة تحتوي على معالج ثنائي النواة بسرعة عالية، مزودة بوحدات Wi-Fi وBluetooth، وتحتوي على بطاقة ذاكرة MicroSD بسعة 32 جيجابايت. وتتميز بسهولة البرمجة وكثرة المنافذ الرقمية، مما يجعلها مناسبة للتعامل مع أكثر من وحدة.

تُعد وحدة ESP32 من أشهر وحدات التحكم الدقيقة المستخدمة في مشاريع إنترنت الأشياء (IoT)، كما أنها تقدم دعمًا ملائمًا للبرمجة، مما يجعلها خيارًا ممتازًا لبناء حلول IoT.

سعة تصاالت 4G معاهرة وتدعم الاتصال اللاسلكي Wi-Fi و Bluetooth.

أهم المعاصفات والخصائص :

المعالجات: شرائي النواة (Dual-core Xtensa LX6) بسرعة تصل إلى 240MHz
الاتصال اللاسلكي Bluetooth v4.2 Wi-Fi 802.11 b/g/n

عدد المنافذ الرقمية (GPIO) أكثر من 30 منها يمكن تخصيصها لمهام مختلفة.

منافذ ADC محول تناظري إلى رقمي: حتى 18 قناة مثل GPIO36 إلى GPIO39

منافذ DAC محول رقمي إلى تناظري: قناتان GPIO25 وGPIO26 منافذ UART: تدعم حتى 3 منافذ UART، مثلًا، GPIO3/U0 RXD، GPIO1/U0 TXD.

دعم لـ وتهـنـكـولـات الـاتـصالـ المـخـتـلـفـة:

CRIC10, CRIC20 & CRIL

C919Q31/SRA C919Q32/SCL 13C

卷之三

حتى 10 نقاط لمس مثل GPIO4 على Touch0

دعم لمنافذ PWM: لجميع منافذ GPIO تقريباً، مما يتيح التحكم في المحركات والإضاءة

جهد التشغيل 3.3V

سهولة التوصيل والبرمجة: من خلال منفذ USB مدمج.

تم استخدام لوحة ESP32 كوحدة المعالجة الرئيسية لربط وحدة

GPS NEO-M8N-0-10) لتحديد الموقع، مع وحدة (SIM800L

الرسائل النصية SMS ، مما يجعلها الخيار الأمثل لمشروع تتبع السيارات.

: المُنافذ المستخدمة في لوحة ESP32

تم استخدام مجموعة من المُنافذ الرقمية في لوحة ESP32 لتوصيل وحدتي GPS وGSM وكانت

كالآتي :

GPIO16 (TX) وGPIO17 (RX) : تم تخصيصهما للتواصل التسلسلي (UART2) مع وحدة

GSM SIM800L لإرسال واستقبال الرسائل النصية SMS

GPIO4 (TX) وGPIO5 (RX) : تم استخدامهما للتواصل مع وحدة NEO-M8N-0-10

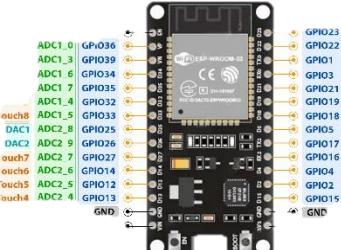
GPIO25 وGPIO26 : باستخدام بروتوكول UART1 ، حيث تُرسل الوحدة إحداثيات الموقع إلى

منافذ احتياطية تدعم الإخراج التاظري (DAC) يمكن استخدامها لاحقاً في تطوير المشروع لإضافة

وظائف مثل التنبيهات الصوتية أو إشارات تناظرية أخرى.

GPIO21(SDA) وGPIO22(SCL)

منافذ تدعم بروتوكول I2C في حال تم استخدام مستشعرات إضافية مثل شاشة OLED



الشكل رقم (2) يوضح لوحة التحكم

ESP32 FEATURES & SPECS		
ESP32 TECHNICAL SPECIFICATIONS		
No.	Component Name	
1	Microprocessor	
2	Processor Core	
3	Operating Voltage	
4	DC Current on 3.3V Pin	
5	DC Current on UO Pins	
6	Maximum Operating Frequency	
7	Frequency Oscillators	
8	Timers	
ESP32 PINOUT		
1	DAC	
2	ADC	
3	Capacitive Touch Sensors	
4	LED PWM	
ESP32 COMMUNICATION PROTOCOLS		
1	Wi-Fi	
2	Bluetooth	
3	Bluetooth Low Energy	
4	UART Protocol	
5	I2C Protocol	
6	DSI Protocol	
7	CAN Protocol	
ESP32 BUILTIN MEMORY		
No.	Parameter Name	Parameter Value
1	SRAM	52KB
2	ROM/Flash Memory	4MB
3	RTC SRAM	1KB

الشكل رقم (1) يوضح خصائص وحدة ESP32

3.3.2 وحدة GPS neo-m8n-0-10

تستخدم هذه الوحدة للحصول على الموقع الجغرافي الدقيق (خط الطول والعرض والارتفاع والسرعة) عبر الأقمار الصناعية. وتعد من الوحدات الدقيقة التي تدعم أنظمة متعددة لتحديد الموقع مثل GPS و GLONASS.

مميزات وحدة GPS NEO-8M

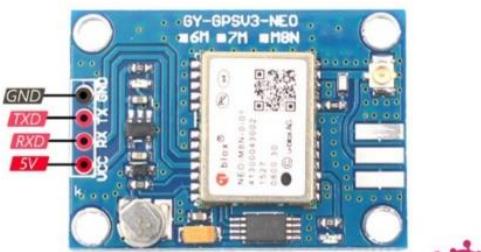
وحدة GPS NEO-8M هي وحدة GPS متطورة مبنية على uBlox m8N، وتدعم بروتوكول اتصال UART بهوائي نشط. يمكن ربط هذه الوحدة بسهولة بوحدة تحكم دقيقة. تحتوي هذه الوحدة على بطارية قابلة لإعادة الشحن، ويمكن توصيلها مباشرةً بجهاز كمبيوتر باستخدام محول USB إلى TTL.

يستطيع جهاز NEO-8M استقبال المعلومات ثم حساب الموقع الجغرافي بدقة عالية وسرعة فائقة. بالإضافة إلى دعمه لشبكات BeiDou و Galileo و GPS/QZSS و GLONASS، يحتوي الجهاز على ذاكرة داخلية لحفظ الإعدادات. NEO-8M متوافق مع Arduino ويمكن استخدامه في أي مشروع.



الشكل رقم (3) يوضح وحدة GPS NEO-M8N-0-10

وحدة GPS GY-NEO-8M



الشكل رقم (4) يوضح أهم منافذ وحدة GPS NEO-M8N-0-10

يحتوي هذا المستشعر على 4 دبابيس:

VIN: مصدر طاقة الوحدة - 5 فولت

GND: الأرض

RX: استقبال البيانات عبر البروتوكول التسلسلي

TX: إرسال البيانات عبر البروتوكول التسلسلي

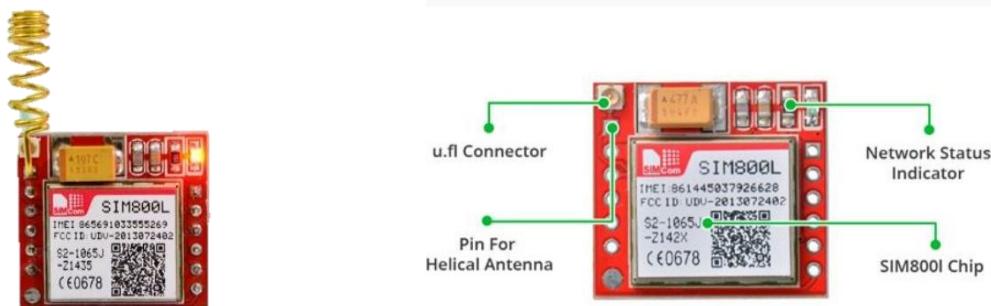
: GSM SIM800L 3.3.3

تُستخدم هذه الوحدة لإرسال واستقبال الرسائل القصيرة عبر شبكة الهاتف المحمول. في هذا المشروع، يتم إرسال إحداثيات GPS من خلال رسالة نصية SMS إلى رقم المستخدم المسجل مسبقاً.

وحدة GSM SIM800L هي مودم GSM مصغر يمكن استخدامه في مجموعة متنوعة من مشاريع إنترنت الأشياء. يمكنك استخدام هذه الوحدة للقيام بأي شيء تقريباً يمكن للهاتف المحمول العادي القيام به، مثل إرسال رسائل نصية قصيرة، وإجراء مكالمات هاتفية، والاتصال بالإنترنت عبر GPRS، وغير ذلك الكثير.

إضافةً إلى ذلك، تدعم الوحدة شبكات GSM/GPRS رباعية النطاق، مما يعني أنها ستعمل في أي مكان تقريباً في العالم. تحتوي الوحدة على شريحة SIM800L GSM خلوية من Simcom.

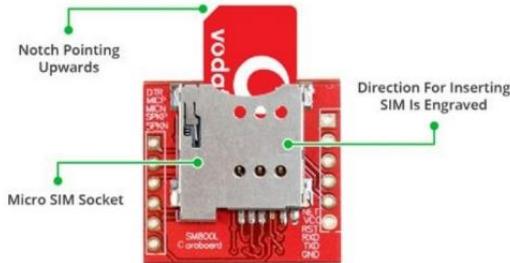
يتراوح جهد تشغيل الشريحة بين 3.4 فولت و 4.4 فولت، مما يجعلها خياراً مثالياً للتزويد المباشر ببطاريات LiPo. هذا يجعلها خياراً ممتازاً للتضمين في المشاريع ذات المساحات



الشكل رقم (5) صورة أخرى لوحدة GSM SIM800L

الشكل رقم (6) يوضح وحدة GSM SIM800L

تحتوي على منفذ لشريحة SIM في الخلف! أي بطاقة Micro SIM 2G ستعمل بشكل مثالي. عادةً ما تكون الطريقة الصحيحة لإدخال بطاقة SIM محفورة على سطح منفذ SIM



الشكل رقم(7) يوضح منفذ sim لوحدة sim800L

السمات :

على الرغم من صغر حجم هذه الوحدة بشكل لا يصدق - بوصة مربعة واحدة فقط - إلا أنها تحتوي على عدد مذهل من الميزات. من بينها:

يدعم النطاق الرباعي: PCS1900 و EGSM900 و DCS1800 و GSM850

اتصل بأي شبكة GSM عالمية باستخدام أي شريحة 2G

إجراء واستقبال المكالمات الصوتية باستخدام مكبر صوت خارجي $8\ \Omega$ و ميكروفون إلكترون

إرسال واستقبال الرسائل النصية القصيرة

إرسال واستقبال بيانات (TCP/IP، HTTP) GPRS وما إلى ذلك

مسح واستقبال البث الإذاعي FM

قوة الإرسال:

الفئة 4 (2 واط) لـ GSM850

الفئة 1 (1 واط) لـ DCS1800

موصلات FL لهواتف الخلايا

يقبل بطاقة Micro SIM

مؤشرات حالة LED

تحتوي وحدة SIM800L على مؤشر LED يظهر حالة شبكتك الخلوية. سيومض المؤشر بمعدلات مختلفة حسب حالة الشبكة.

هذا هو دبوس مصدر الطاقة. يُرجى العلم أن نطاق جهد التشغيل لشريحة SIM800L يتراوح بين 3.4 و 4.4 فولت، لذا فإن توصيل هذه الوحدة بمخرج أردوينو 5 فولت قد يُسبب تلفاً لها. فهي لا

تعمل حتى بجهد 3.3 فولت! استخدم بطارية ليثيوم بوليمر أو محول تيار مستمر - تيار مستمر (DC-DC Buck).

RST (إعادة تعيين) هو دبوس إعادة الضبط. إذا كانت الوحدة في مكان سيء للغاية، فاسحب هذا الدبوس إلى الوضع المنخفض لمدة ١٠٠ ملي ثانية لإجراء إعادة ضبط كاملة.

RXD (المستقبل) يستخدم هذا الدبوس لإرسال الأوامر إلى الوحدة. هذا الدبوس يعمل تلقائياً بالبود، لذا فإن معدل البود الذي ترسل به أمر "AT" بعد إعادة الضبط هو معدل البود المستخدم.

GND هو دبوس الأرض.



الشكل رقم (8) توضح منافذ وحدة GSM SIM800L

3.3.4 وحدة الطاقة :

يتم تغذية النظام ببطارية ليثيوم 12 فولت، كما تم استخدام منظم جهد لتوفير 5 فولت آمن إلى الوحدات المختلفة لضمان عملها بشكل مستقر.

3.3.5 الأسلام ولوحة التجارب :

تم استخدام أسلاك توصيل ولوحة تجارب لربط المكونات مع بعضها في مرحلة التصميم والاختبار الأولية قبل الانتقال إلى تثبيت نهائي داخل صندوق الحماية.

3.3.6 صندوق الحماية (Enclosure) :

تم وضع المكونات داخل صندوق حماية بلاستيكي للحفاظ على الدوائر من العوامل الخارجية كالرطوبة والغبار، ولتحقيق المشروع داخل المركبة بسهولة.

: SIM شريحة 3.3.6

يتم تركيبها داخل وحدة SIM800L، وتُستخدم لإرسال الرسائل القصيرة إلى رقم المستخدم عند طلب الموقع، وكذلك لتفعيل خاصية الاتصال بالشبكة إذا تم لاحقاً دعم نقل البيانات عبر GSM.

: (Ajyalyemen.site) خادم ويب 3.3.7

تم استخدام استضافة خاصة تم رفع ملفات واجهة الموقع عليها، وهي المسئولة عن استقبال البيانات عبر API HTTP وتخزينها في قاعدة بيانات MySQL وعرضها على الخريطة لحظياً من خلال واجهة العميل.

: 3.4 آلية العمل حسب أوضاع التشغيل

يعتمد نظام تتابع المركبات في هذا المشروع على ثلاثة أوضاع تشغيل رئيسية، تم تصميمها لتناسب مع اختلاف ظروف الاتصال بشبكة الإنترنت واحتياجات المستخدم.

: (SMS) وضع الرسائل القصيرة 3.4.1

يُعد هذا الوضع مصمماً ليعمل بشكل مستقل تماماً عن الإنترنط، ويتيح للمستخدم معرفة موقع السيارة في الوقت الفعلي من خلال إرسال رسالة قصيرة فقط، مما يجعله مثالياً للمناطق ذات التغطية الضعيفة.

آلية العمل:

1. يقوم المستخدم بإرسال رسالة نصية تحتوي على الكلمة المفتاحية LOC إلى رقم الشريحة المثبتة في وحدة SIM800L.
2. تتلقى وحدة ESP32 الرسالة عبر المنفذ التسلسلي وتتعرف على الكلمة المفتاحية.
3. يتم جلب الإحداثيات من وحدة GPS وتحويلها إلى رابط Google Maps.
4. يتم إرسال الرابط إلى المستخدم عبر رسالة SMS باستخدام وحدة GSM.

: (Offline Mode) 3.4.2 الوضع غير المتصل

في هذا الوضع، يتعامل النظام مع الحالات التي يكون فيها الاتصال بالإنترنت غير متوفراً مثل انقطاع Wi-Fi أو ضعف الشبكة .(وبدلاً من فقدان البيانات، يقوم النظام بتخزينها مؤقتاً في الذاكرة حتى يعود الاتصال).

آلية العمل:

1. تتحقق وحدة ESP32 من حالة الاتصال بالإنترنت.
2. في حال عدم وجود اتصال، يتم تخزين بيانات (GPS الموقع + الوقت) في مصفوفة من نوع `vector<GPSData>`.
3. عند عودة الاتصال، يتم إرسال جميع البيانات الموجلة تلقائياً إلى الخادم عبر HTTP.
4. بعد التأكيد من رفع البيانات بنجاح، يتم حذفها من الذاكرة.

3.4.3 وضع التتبع اللحظي (Real-Time Mode) :

يُستخدم هذا الوضع عند توفر اتصال ثابت بالإنترنت عبر نقطة اتصال هاتف أو شبكة Wi-Fi ويتيح إرسال الإحداثيات فوراً إلى الخادم ليتم عرضها على خريطة Google في واجهة الموقع

آلية العمل:

1. تقرأ وحدة ESP32 البيانات من وحدة GPS كل فترة زمنية محددة (30 ثانية مثلاً).
2. يتم التحقق من تغيير الموقع بمسافة معينة مثل (30 متر) قبل الإرسال لتوفير البيانات.
3. يتم إرسال الإحداثيات إلى واجهة API في الخادم عبر HTTP GET.
4. يتم عرض الموقع تلقائياً في واجهة المستخدم الخاصة بالعميل على الموقع.

3.5 تنظيم البرنامج الأساسي (Main Program Structure) :

تمت برمجة وحدة ESP32 باستخدام Arduino IDE ، وتم اعتماد هيكلية منظمة للكود تسهل من فهمه وتطويره لاحقاً. ويكون البرنامج من الأجزاء التالية:

• **الدالة: setup()**

تُستخدم لتهيئة النظام عند بدء التشغيل، وتشمل تهيئة وحدات GPS و GSM و Wi-Fi، وضبط الاتصال التسلسلي، وتهيئة المتغيرات الضرورية التي يعتمد عليها النظام.

• **الدالة: loop()**

هي الحلقة الرئيسية التي تعمل باستمرار، وتقوم باستقبال الرسائل النصية من المستخدم، وقراءة بيانات GPS، والتحقق من وضع التشغيل النشط-Real (SMS / Offline / Real-time)، ثم تنفيذ الإجراء المناسب لكل وضع.

• **الدالة: processGPSData()**

تقوم بتحليل جمل NMEA المستلمة من وحدة GPS، لاستخراج المعلومات الأساسية مثل خطوط الطول والعرض، والتاريخ، والوقت، وعدد الأقمار الصناعية.

• **الدالة: sendGPSData()**

مسؤولة عن إرسال البيانات الجغرافية إلى الخادم (server) باستخدام بروتوكول HTTP، وُتستخدم في وضع التتبع اللحظي أو عند رفع البيانات المؤجلة في وضع Offline.

• **الدالة: convertAndPrintLocalDateTime()**

تقوم بتحويل التوقيت القائم من وحدة GPS الذي يكون بصيغة (UTC إلى توقيت محلي (GMT+3)، ليُعرض للمستخدم بشكل واضح ومناسب.

• **الدالة: haversine()**

تُستخدم لحساب المسافة بين آخر نقطة تم إرسالها والموقع الحالي، وذلك لتقليل الإرسال المتكرر عندما تكون المركبة ثابتة أو تتحرك ضمن مسافة صغيرة (أقل من 30 متراً مثلاً).

ملاحظات إضافية:

• يتم تحديد الفاصل الزمني بين كل إرسال باستخدام متغير مثل (uploadInterval) مثلاً كل 30 ثانية.

• يُخزن الموقع الأخير المُرسل في متغير lastSentPosition لاستخدامه في دالة haversine().

• في وضع Offline ، تُخزن البيانات داخل مصفوفة من نوع <vector<GPSData>، وعند توفر الاتصال يتم رفعها ثم حذفها.

- تحليل رسائل المستخدم يتم عبر دوال مثل `SerialAT.readString()` أو `Serial.read()` للتحقق مما إذا كانت تحتوي على الكلمة المفاتيحية "LOC" ، والرد بإحداثيات الموقع عند الحاجة.

الفصل الرابع:
التحليل والتصميم

4.1 الجدوى الإجتماعية للمشروع :

يُعد هذا المشروع ذا جدوى اجتماعية واضحة، حيث يسهم بشكل مباشر في تعزيز الأمان المجتمعي من خلال تمكين مالكي المركبات من تتبع مرکباتهم في الوقت الحقيقي أو عند انقطاع الإنترنط، مما يرفع من فرص استرجاع السيارات المسروقة أو المفقودة، خصوصاً في البيئات التي تعاني من ضعف البنية التحتية الأمنية كحال العديد من المحافظات اليمنية مثل محافظة مأرب.

يسهم النظام أيضاً في خفض القلق والضغط النفسي لدى الأفراد نتيجة غياب وسائل التتبع المتاحة، ويوفر وسيلة ميسّرة وبسيطة لفئات متعددة من المجتمع، بما في ذلك من لا يمتلكون معرفة تقنية عالية، عبر رسائل SMS.

كما يُعتبر المشروع خطوة نحو نشر ثقافة استخدام التقنيات الحديثة لأغراض السلامة والحماية، وهو ما يدعم أهداف التنمية المجتمعية، ويعزز من اعتماد المجتمع على حلول تقنية بديلة منخفضة الكلفة وفعالة في ذات الوقت.

4.2 التحليل :

في هذه المرحلة تم تحليل فكرة النظام بشكل منهجي، بهدف بناء تصور دقيق حول آلية تتبع المركبات اعتماداً على تقنيات الاتصال اللاسلكي (GPS و GSM) ووحدة تحكم ذكية (ESP32). يرتكز النظام على توفير وسيلة تتبع عملية وسهلة الاستخدام، تُمكّن المستخدم من تحديد موقع المركبة بدقة عبر الرسائل القصيرة (SMS)، دون الحاجة إلى تطبيقات أو اتصال دائم بالإنترنت.

يتكون النظام من ثلاثة أوضاع تشغيل رئيسية:

SMS Mode: يتيح للمستخدم إرسال أمر "LOC" إلى الشريحة المدمجة داخل النظام، والذي يرد تلقائياً برسالة SMS تتضمن رابطاً إلى موقع السيارة على Google Maps.

Offline Mode: عند انقطاع الإنترنط، يقوم النظام بتخزين الإحداثيات تلقائياً داخل مصفوفة بيانات، ويتم إرسالها لاحقاً إلى خادم الويب الخاص بمجرد استعادة الاتصال.

Real-time Mode: يسمح بمتابعة المركبة لحظياً على خريطة تفاعلية من خلال الموقع عبر الإنترنط، باستخدام مفتاح API خاص بالمستخدم.

خلال عملية التحليل، تمأخذ النقاط التالية بعين الاعتبار:

توفر تغطية GPS: لضمان دقة الإحداثيات.

قوة شبكة GSM: لتفادي تأخير أو فشل إرسال الرسائل.

كفاءة استهلاك الطاقة: لضمان استمرار عمل النظام في الظروف الميدانية.

سهولة البرمجة والتوسعة المستقبلية: لتحديث الأوامر أو إضافة ميزات إضافية

4.3 التصميم :

4.3.1 التصميم المادي (Hardware Design) :

يرتكز التصميم المادي على توصيل العناصر الإلكترونية بطريقة منظمة ومستقرة، وفق التكوين التالي:

وحدة التحكم ESP32: المعالج المركزي الذي يدير الاتصالات ويوجه الأوامر.

وحدة GPS NEO-M8N-0-10: لقراءة الإحداثيات الجغرافية بدقة عالية.

وحدة GSM SIM800L: لإرسال واستقبال الرسائل النصية عبر شبكة الهاتف.

منظم الجهد 7805 وديود الحماية: لضمان ثبات الجهد وتفادي تلف الدوائر.

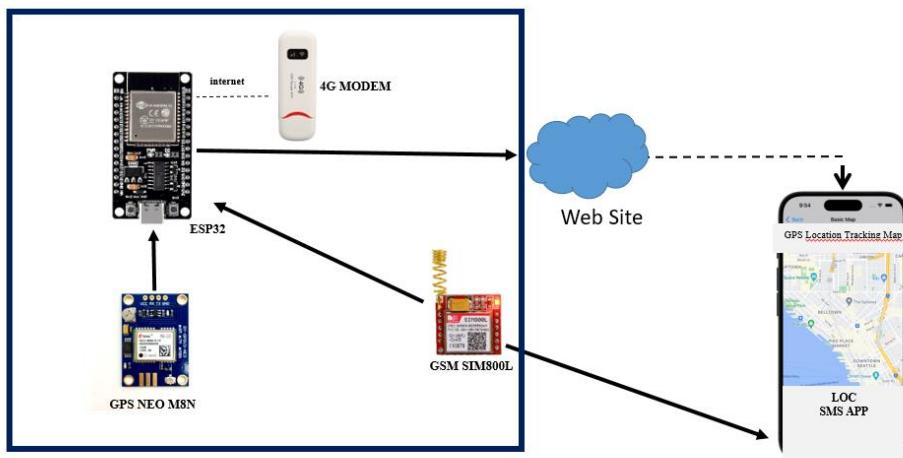
مصدر طاقة 12 فولت: لتشغيل النظام بكفاءة واستقرارية.

تم اختيار هذه المكونات بناءً على معايير الكفاءة، التوفير، وانخفاض التكلفة، مع الحرص على تقليل التداخل وضمان التشغيل المستقر والطاقة الآمنة.

4.3.2 التصميم المنطقي (Software Design) :

يعتمد التصميم البرمجي على تسلسل دقيق يبدأ من استقبال الأوامر عبر الرسائل النصية، وتمر بتحليلها، ثم تتنفيذ المهام المطلوبة حسب وضع التشغيل النشط (SMS – Offline – Real-time).

: Block Diagram 4.3.2.1



الشكل رقم (9) يوضح مخطط Block Diagram ().

يوضح المخطط الهيكلي (Block Diagram) تفاعل المكونات التالية:

1. وحدة GPS NEO-M8N :

مسئولة عن استقبال الإحداثيات من الأقمار الصناعية.

تُرسل البيانات بصيغة NMEA إلى ESP32 عبر UART.

تحلّل باستخدام دوال مخصصة لاستخراج خطوط الطول والعرض.

2. وحدة التحكم ESP32 :

تستقبل البيانات وتتحقق من وضع التشغيل.

تنفذ أحد السيناريوهات الثلاثة:

.SMS Mode: يرد على المستخدم برسالة تحتوي على رابط Google Maps

.Real-time Mode: يرفع الإحداثيات إلى السيرفر باستخدام HTTP API Key و

.Offline Mode: يخزن البيانات مؤقتاً ويرسلها لاحقاً عند توفر الإنترنت.

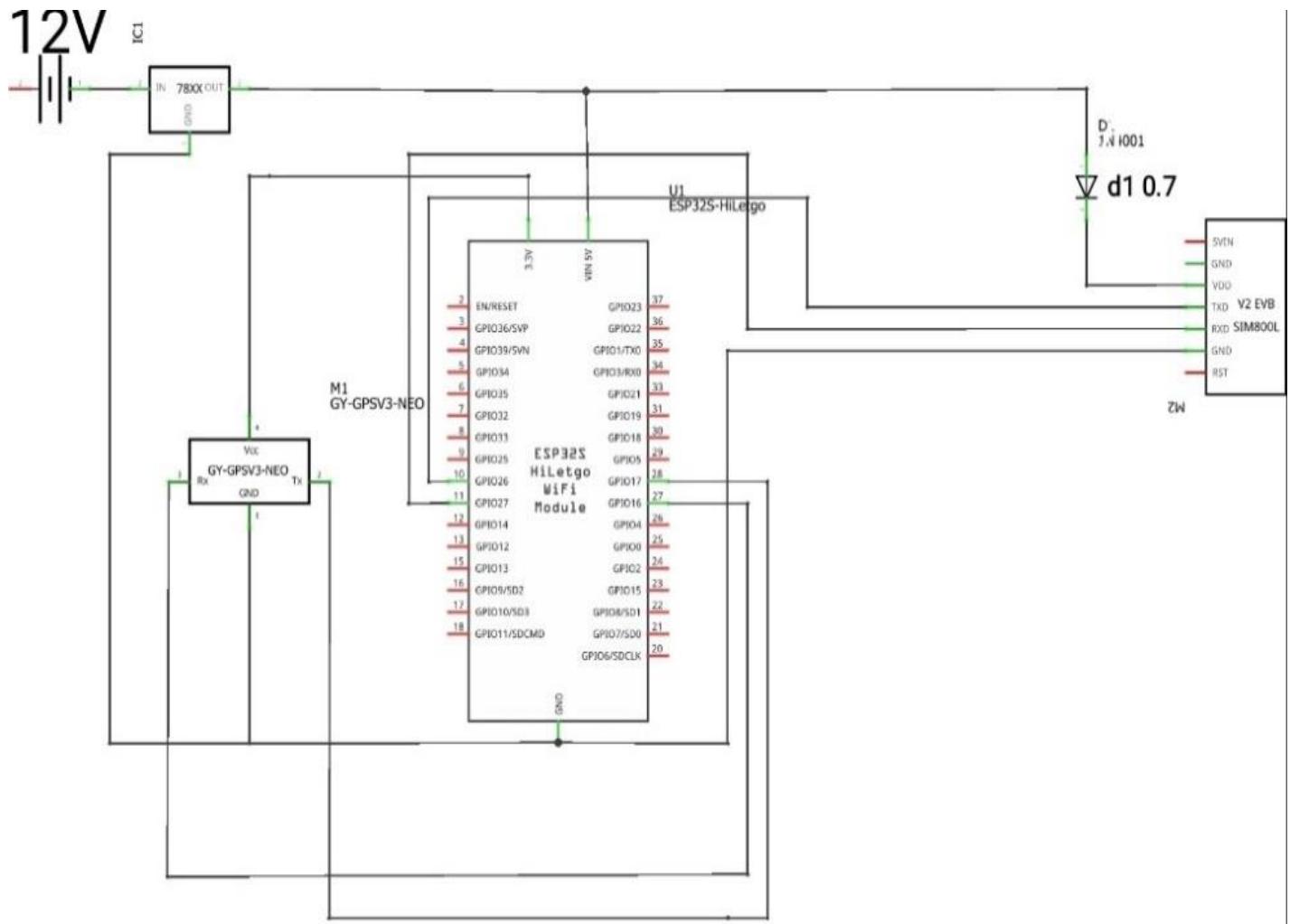
3. وحدة GSM SIM800L :

تستقبل رسائل المستخدم.

تُرسل الإحداثيات عبر SMS عند الطلب.

لا تعتمد على الإنترنٌت.

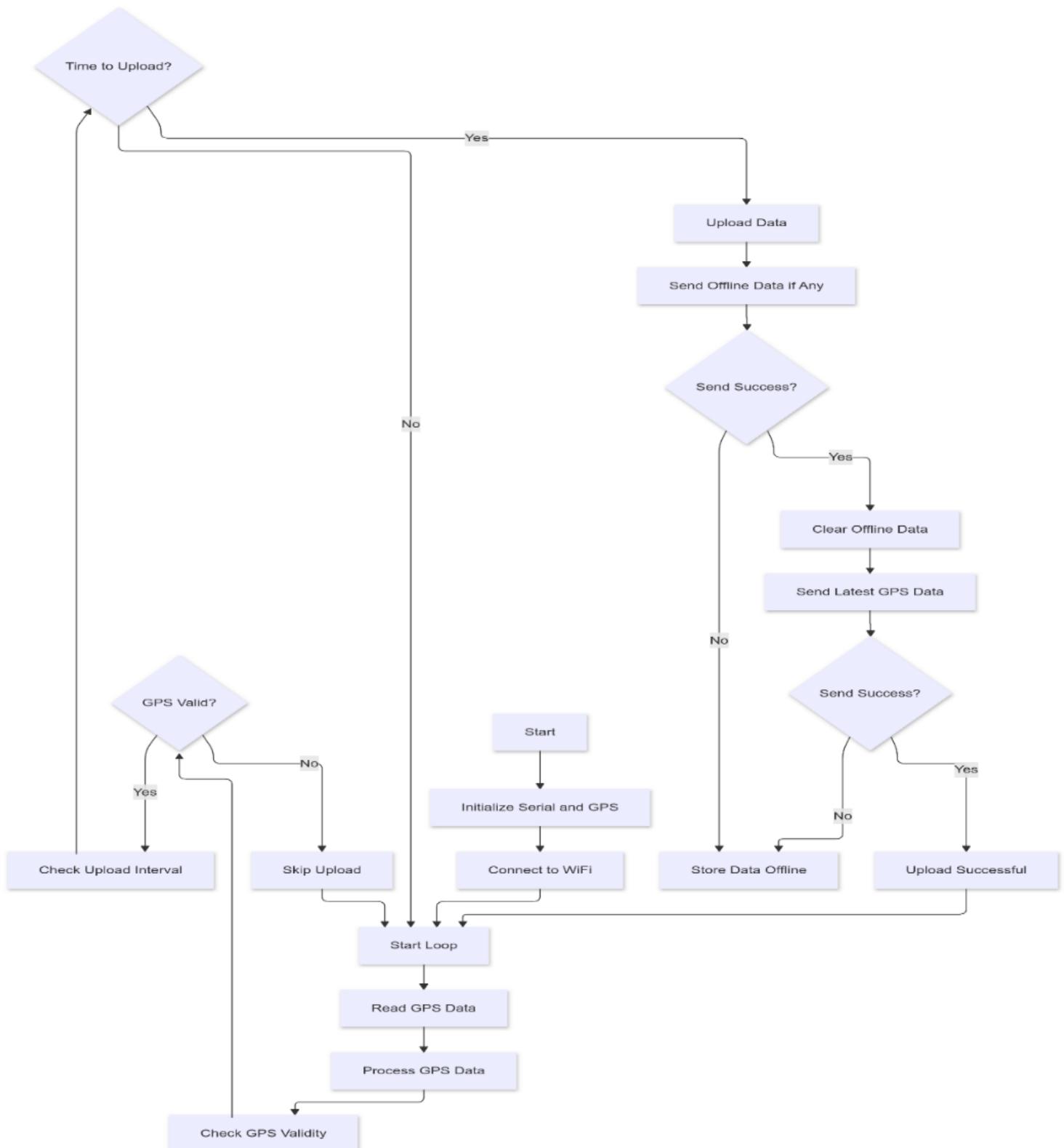
: Circuit Diagram 4.3.2.2



fritzing

الشكل رقم (10) يوضح مخطط Circuit Diagram

4.3.2.3 مخطط التدفق (Flowchart) :



الشكل رقم (11) يوضح مخطط تدفق البيانات (Follow Chart)

شرح مخطط تدفق النظام : (Flowchart Explanation)

يمثل هذا المخطط خطوات عمل نظام تتبع المركبات الذي يعتمد على وحدة GPS لجمع الإحداثيات، ويستخدم شبكة WiFi أو وحدة GSM لرفع البيانات إلى الخادم. يحتوي النظام على آلية لمعالجة البيانات في ثلاث حالات رئيسية: التهيئة، التحقق من صلاحية البيانات، ثم الرفع أو التخزين عند انقطاع الاتصال. وفيما يلي شرح تفصيلي لكل جزء من أجزاء المخطط:

1. مرحلة التهيئة (Initialization)

Start: بداية تشغيل النظام عند تزويده بالطاقة.

Initialize Serial and GPS: يتم تهيئة الاتصال التسلسلي (Serial) وتشغيل وحدة تحديد المواقع (GPS).

Connect to WiFi: يحاول النظام الاتصال بشبكة WiFi المعرفة مسبقاً.

Start Loop: بعد التهيئة، يبدأ البرنامج في الدخول في حلقة مستمرة لتنفيذ المهام بشكل دوري.

Read GPS Data: يتم قراءة بيانات الموقع (Latitude, Longitude) من وحدة GPS.

Process GPS Data: تتم معالجة البيانات مثل تحويل الإحداثيات إلى صيغة مفهومة وتنسيقها.

2. التحقق من صلاحية بيانات GPS

Check GPS Validity: يتم فحص ما إذا كانت البيانات المقرؤة من وحدة GPS صحيحة وغير فارغة.

إذا كانت البيانات غير صالحة: يتم تجاوز عملية الرفع ويتم الرجوع إلى بداية الحلقة.

إذا كانت البيانات صالحة:

Check Upload Interval: يتم فحص ما إذا كان قد مر وقت كافٍ منذ آخر عملية رفع.

?Time to Upload

إذا كانت الإجابة "نعم": يتم الانتقال إلى عملية رفع البيانات.

إذا كانت الإجابة "لا": يتم تأجيل عملية الرفع مؤقتاً والعودة إلى الحلقة.

3. مرحلة رفع البيانات إلى الخادم (Data Upload)

Upload Data: يتم محاولة رفع البيانات إلى الخادم عبر الإنترنت.

Send Offline Data if Any: في حال وجود بيانات مخزنة سابقاً أثناء انقطاع الاتصال، يتم إرسالها أولاً.

?Send Success

إذا تم رفع البيانات بنجاح:

Clear Offline Data: يتم حذف البيانات المخزنة القديمة.

Send Latest GPS Data: إرسال أحدث بيانات GPS التي تم قراءتها.

?Send Success

إذا نجحت عملية الإرسال: يتم اعتبارها Upload Successful.

إذا فشلت: يتم Store Data Offline (تخزين البيانات محلياً لإرسالها لاحقاً).

إذا فشلت عملية إرسال البيانات المخزنة أو لم يكن هناك اتصال:

يتم Store Data Offline مباشرةً.

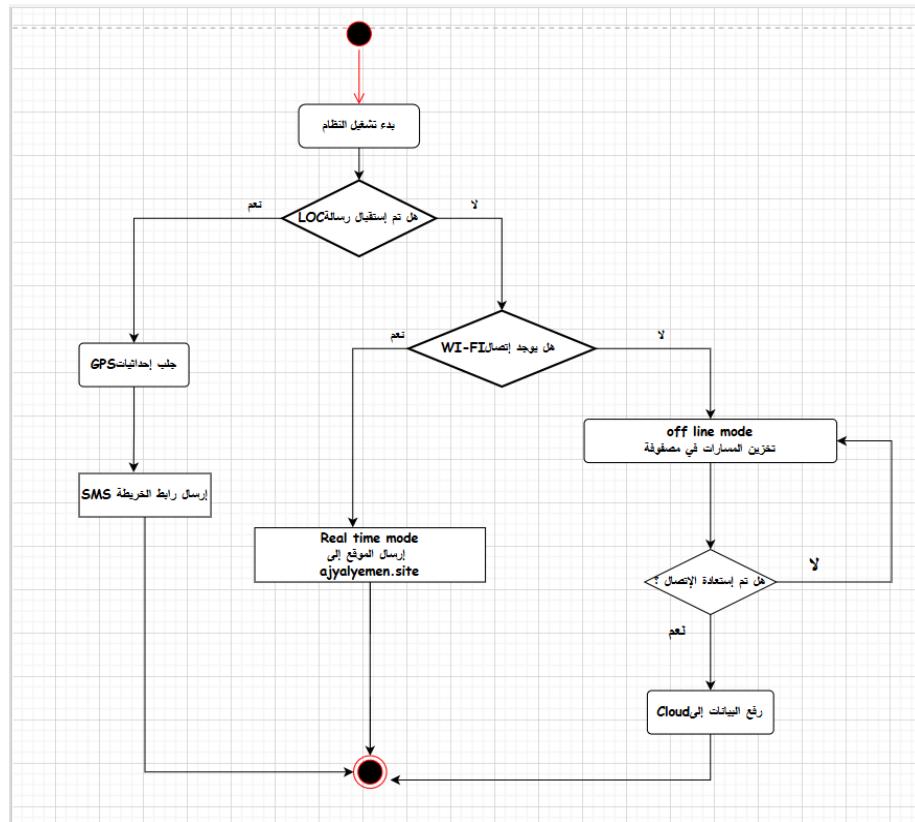
ملاحظة عامة:

يعتمد النظام على آلية ذكية لمعالجة انقطاع الاتصال، حيث يتم تخزين البيانات مؤقتاً في الذاكرة الداخلية، وإرسالها تلقائياً بمجرد توفر الاتصال بالشبكة.

4.3.2.4 مخطط النشاط (Activity Diagram) :

يُظهر هذا المخطط تسلسل العمليات داخل النظام وفقاً لسير العمل الآتي:

1. النظام يبدأ التشغيل.
2. التحقق من وجود اتصال بالشبكة.
3. التتحقق من وجود أمر وارد (مثل LOC).
4. قراءة الإحداثيات من GPS.
5. تنفيذ الإجراء المناسب:
 - .SMS Mode في حالة إرسال رسالة في حالة SMS Mode
 - .Offline Mode في حالة تخزين البيانات في حالة Offline Mode
 - .Real-time Mode في حالة إرسال فوري إلى السحابة في حالة Real-time Mode
6. انتظار الطلب التالي.



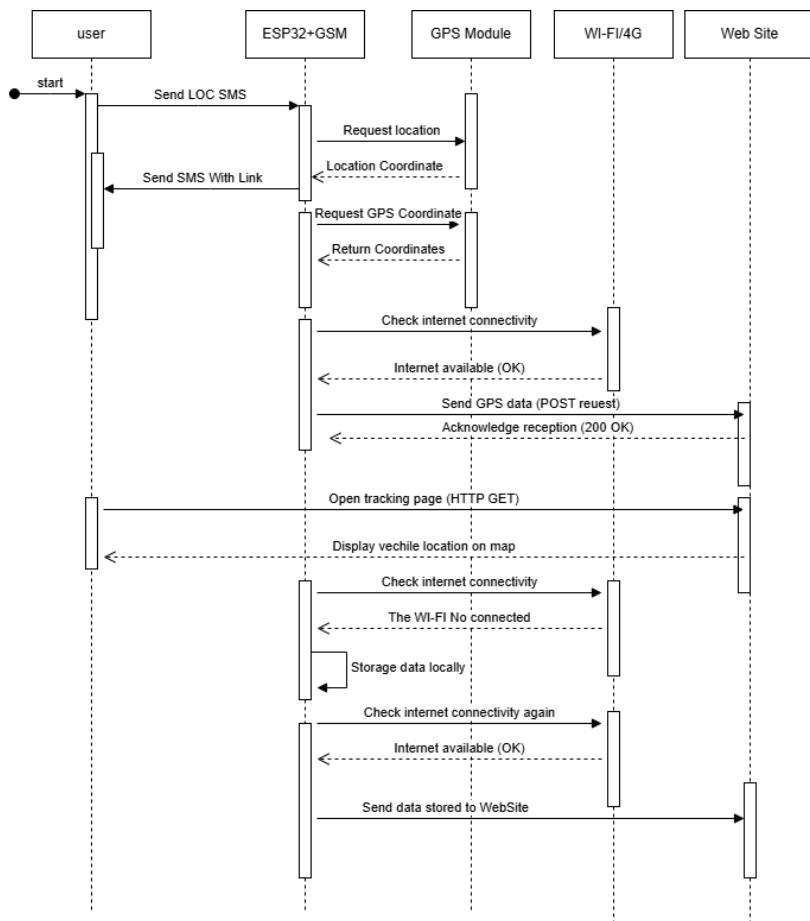
الشكل رقم (12) يوضح مخطط النشاط

4.3.2.5 مخطط الحالات : (Use Case Diagram)



الشكل رقم (13) يوضح مخطط الحالة (Use Case)

4.3.2.6 مخطط التسلسل : Sequence Diagram



الشكل رقم (14) يوضح مخطط التسلسل (Sequence Diagram)

تسلسل الاحداث بين مكونات النظام : (Sequence Diagram Explanation)

يوضح الشكل التالي تسلسل العمليات التي تحدث بين المستخدم ووحدة ESP32+GSM ووحدة GPS وموقع الويب. يبدأ النظام بتلقي رسالة SMS من المستخدم لطلب الموقع، ثم يتم التعامل مع كل وضع من أوضاع التشغيل الثلاثة (SMS Mode – Real-time Mode – Offline Mode) وفقاً لحالة الاتصال بالشبكة. يتم تخزين البيانات محلياً عند انقطاع الإنترنэт وإرسالها تلقائياً إلى الخادم عند استعادة الاتصال. كما يتم تحديث الموقع اللحظي للمركبة كل 30 ثانية في وضع Real-time، فقط في حال تحركت السيارة أكثر من 30 متراً لتقليل التكرار.

: (Web Interface Design) تصميم واجهة الويب 4.3.4

تم تطوير واجهة ويب مخصصة لتوفير وسيلة تفاعلية لمراقبة وتتبع المركبة، مع مراعاة سهولة الاستخدام وتجربة المستخدم. هذه الواجهة ترتبط بالسيفر الخاص بالمشروع وتعرض بيانات التتبع لحظياً بالإضافة إلى أدوات الإدارة. تتكون الواجهة من:

: 4.3.3.1 صفحة تسجيل الدخول

هي البوابة الرئيسية للوصول إلى النظام، حيث يتم التحقق من بيانات المستخدم (اسم المستخدم وكلمة المرور). يسمح النظام بالدخول حسب الصالحيات (مدير أو عميل).

The screenshot shows a login form titled 'نظام تتبع السيارات' (Car Tracking System). The form has two fields: 'البريد الإلكتروني' (Email) containing 'Hakeem@gmail.com' and 'كلمة المرور' (Password) containing '.....'. A blue 'تسجيل الدخول' (Login) button is at the bottom. The URL in the browser bar is 'ajyalyemen.site/car_system/dashboard/'.

الشكل رقم (15) يوضح صفحة تسجيل الدخول لموقع الويب

: 4.3.3.2 لوحة التحكم (Dashboard)

بعد تسجيل الدخول، يتم تحويل المستخدم إلى لوحة التحكم. يظهر للمستخدم حسب نوعه:

- المدير: يمكنه إضافة مستخدمين، تحديد الصالحيات، وإضافة المركبات.
- العميل: يمكنه فقط عرض موقع مركبته ومتابعتها.

مرحباً بك، مدير النظام!

هنا يمكنك إدارة المستخدمين والسيارات وعرض جميع بيانات التتبع.

إدارة السيارات

إضافة سيارة

إجراءات	المستخدم	معرف السيارة	اسم السيارة	
<button>حذف</button> <button>تعديل</button>	احمد خالد	1	مرسيدس	

إدارة المستخدمين

إضافة مستخدم

إجراءات	الدور	البريد الإلكتروني	الاسم
<button>حذف</button> <button>تعديل</button>	عميل	ahmed@gmail.com	احمد خالد
<button>تعديل</button>	مدير	Hakeem@gmail.com	عبد الحكيم
<button>حذف</button> <button>تعديل</button>	عميل	Saleh@gmail.com	صالح

الشكل رقم(16) يوضح لوحة تحكم المدير

تيح لوحة التحكم إضافة مستخدم جديد وتحديد نوعه (عميل أو مدير)، بالإضافة إلى إمكانية إدخال بيانات المركبة معرف المركبة يجب أن يكون رقم 1 كما هو مذكور في كود ESP32 .

X

إضافة مستخدم جديد

الاسم

البريد الإلكتروني

كلمة المرور

الدور

إضافة
إلغاء

الشكل رقم(18) يوضح واجهة إضافة مستخدم

X

إضافة سيارة جديدة

اسم السيارة

معرف السيارة (يجب أن يكون فريداً)

المستخدم

إضافة
إلغاء

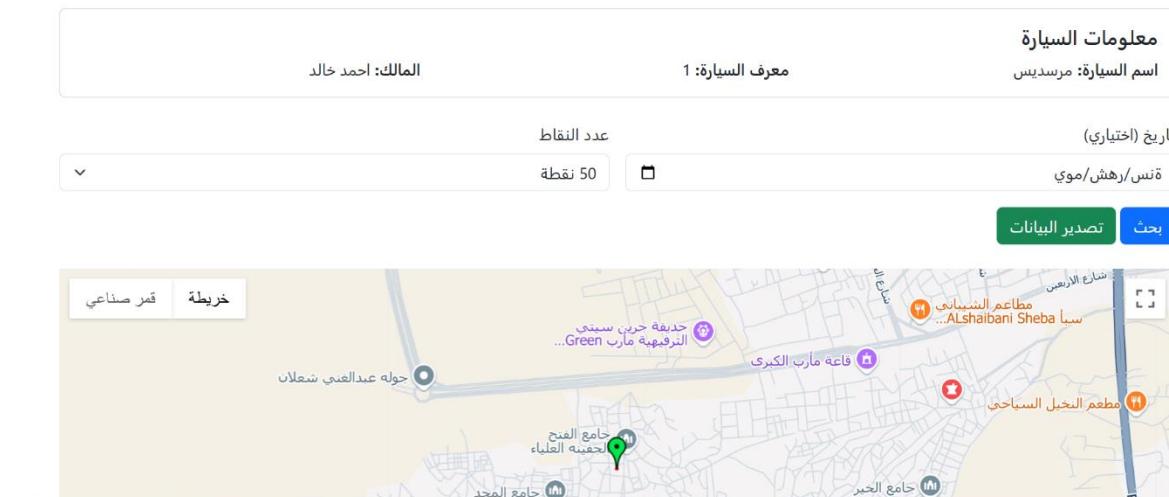
الشكل رقم (17) يوضح واجهة إضافة مركبة

4.3.3.4 صفحة تتبع المركبة :

عند النقر على "عرض الموقع"، يتم فتح صفحة تعرض:

- خريطة تفاعلية تعرض موقع المركبة الحالي باستخدام مكتبة Leaflet.
 - جدول تحت الخريطة يحتوي على بيانات الموقع السابقة (خط الطول، العرض، التوقيت).

سجل مواقع السيارة



الشكل رقم(19) يوضح صفحة تتبع الموقع الجغرافي



الشكل رقم (20) يوضح صفحة تتبع الموقع الجغرافي باستخدام القمر الصناعي

سجل المواقع

رابط الخريطة	خط الطول	خط العرض	التاريخ والوقت
عرض على الخريطة	45.306445	15.448889	18:07:37 2025-07-01
عرض على الخريطة	45.306437	15.448885	18:00:00 2025-07-01
عرض على الخريطة	45.306437	15.448891	17:52:30 2025-07-01
عرض على الخريطة	45.306446	15.448868	19:51:28 2025-06-30
عرض على الخريطة	45.306451	15.44887	19:45:15 2025-06-30

الشكل رقم (21) يوضح جدول سجل المواقع خطوط الطول والعرض

4.3.3.5 تصدير المواقع إلى ملف إكسل:

تم دمج زر يسمح للمستخدم بتصدير جدول الإحداثيات إلى ملف CSV قابل للتنزيل، وذلك لتوثيق الرحلات أو حفظها لأغراض لاحقة.

4.3.3.6 التقنيات المستخدمة في تصميم الواجهة :

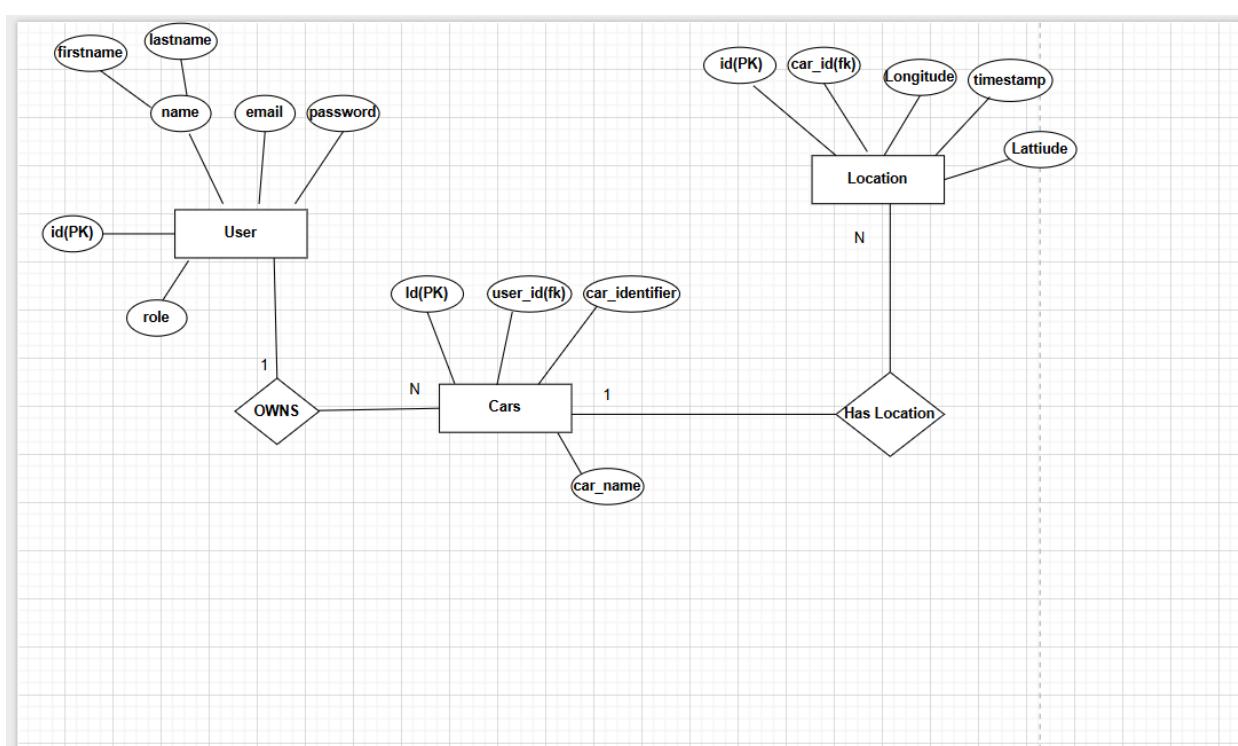
تم استخدام الأدوات والتقنيات التالية:

- HTML / CSS / JavaScript لهيكلة الواجهة وتصميمها.
- مكتبة Leaflet.js لعرض الخريطة.
- خادم PHP لتخزين واسترجاع البيانات.
- استضافة على سيرفر Hostinger.

4.3.3.7 تصميم قاعدة البيانات :

تم تصميم قاعدة البيانات لتناسب مع مكونات النظام الرئيسية (المستخدمين ، السيارات ، الموقع) يحتوي هذا على المخطط الكياني للعلاقات بين الجداول (ERD) الذي يوضح العلاقات بين الكيانات بشكل مرئي .

مخطط الكيانات وال العلاقات (ERD) :



الشكل رقم (22) يوضح مخطط ERD لقاعدة البيانات

الفصل الخامس:

التنفيذ المادي والبرمجي

5.1 مرحلة التنفيذ :

يتناول هذا الفصل الجانب العملي للمشروع من حيث تنفيذ المكونات المادية وتطوير البرمجيات اللازمة لتشغيل النظام، بهدف إنتاج نموذج أولي فعال لنظام تتبع المركبات يعمل بثلاثة أوضاع مختلفة، اعتماداً على تقنيات GPS و Wi-Fi و GSM. وقد تم تنفيذ النظام على مراحلتين متكمالتين: تنفيذ الأجهزة (Hardware) وتطوير البرمجيات (Software).

5.1.1 تنفيذ الأجهزة (Hardware Implementation) :

أولاً: المكونات المستخدمة

تم اختيار المكونات المادية بعناية لضمان توافرها في السوق، وانخفاض تكلفتها، وقدرتها على العمل ضمن بيئه المشروع. وتشمل:

رقم	المكون	الوظيفة
1	وحدة ESP32	وحدة تحكم متقدمة ثنائية النواة تدير عمليات الاتصال والتحكم بوحدات GPS و GSM
2	وحدة GPS NEO-M8N-0-10	مسؤولة عن تحديد الموقع الجغرافي بدقة عبر الأقمار الصناعية
3	وحدة GSM SIM800L	لإرسال واستقبال الرسائل القصيرة SMS وربط النظام بشبكة الهاتف المحمول
4	منظم جهد 7805	لضبط الفولت الداخل إلى وحدة GSM إلى 5 فولت
5	ديود حماية 1N4007	لخفض الجهد من 5 فولت إلى 4.2 ليتناسب مع GSM.
6	مصدر طاقة 12V	لتغذية النظام بشكل مستقر
7	أسلاك توصيل ولوحة تجارب	لاختبار التوصيلات قبل اللحام النهائي

جدول (1) : جدول يوضح القطع المستخدمة وعملها

ثانياً: آلية التوصيل :

تم توصيل وحدة GPS على المنفذ UART1 RX=16، TX=17 بالمنفذ المخصص لـ SIM800L على المنفذ UART2 RX=27، TX=26 مع خافض جهد ومنظم.

تم استخدام دايمود حماية ومنظم جهد لحماية وحدة SIM800L من الجهد الزائد.

تم توصيل النظام بمصدر طاقة خارجي بجهد 12 فولت لضمان استمرارية التشغيل.

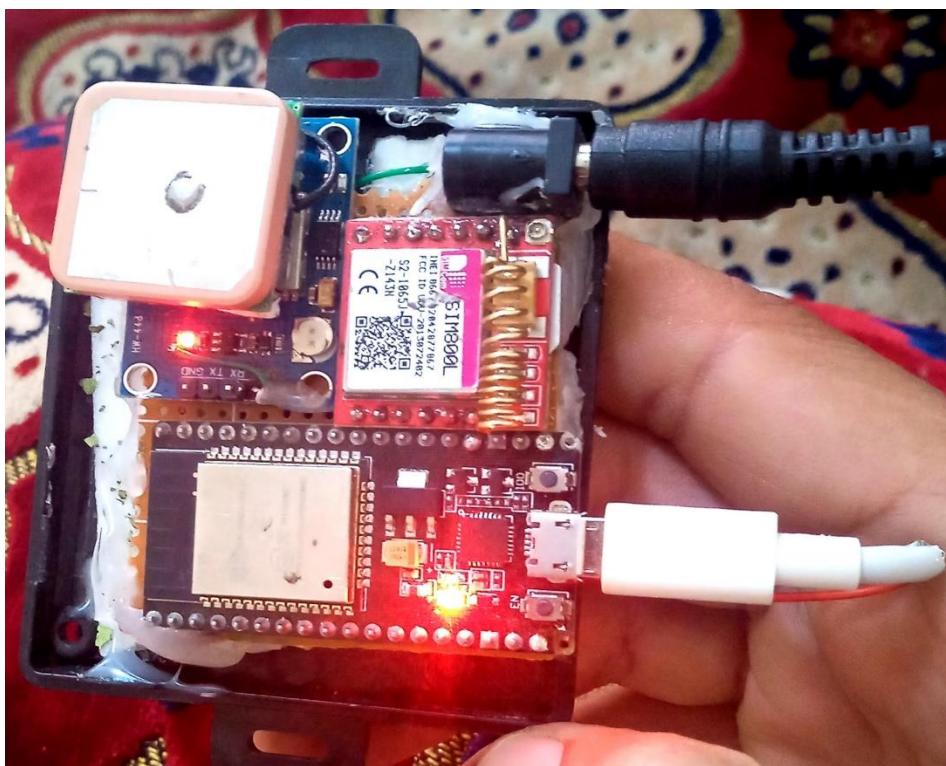
ثالثاً: اعتبارات التوصيل :

تقليل التشویش الكهرومغناطيسي باستخدام توصيلات قصيرة ومعزولة.

التأكد من ثبات الفولت المناسب لوحدات GPS و GSM.

تأمين التهوية الجيدة للدوائر الإلكترونية.

5.1.1.1 صورة النظام بعد التوصيل :



الشكل رقم (23) يوضح صورة النظام بعد التركيب

5.1.2 تنفيذ البرمجيات : (Software Implementation)

أولاً: بيئة التطوير

تم استخدام برنامج Arduino IDE لبرمجة وحدة ESP32.

تم استخدام PHP و MySQL لتطوير واجهة الموقع الإلكتروني وقاعدة البيانات.

ثانياً: البرمجة على ESP32

تم كتابة كود برمجي مهيكل يدير أوضاع التشغيل الثلاثة:

SMS Mode: يستقبل رسالة "LOC" ويرسل الإحداثيات في رابط Google Maps.

Offline Mode: يخزن البيانات في متغير <vector<GPSData>> عند فقد الاتصال.

Real-Time Mode: يرسل البيانات إلى الموقع مباشرة كل 30 ثانية باستخدام HTTP.

تم استخدام المكتبات التالية:

WiFi.h و HTTPClient.h: للاتصال بالإنترنت والخوادم.

GPS++TinyGPS: لتحليل بيانات NMEA من وحدة GPS.

ArduinoJson.h: لتنسيق البيانات بصيغة JSON عند الحاجة.

ثالثاً: تصميم الموقع الإلكتروني

تم تطوير موقع ويب .

يحتوي الموقع على:

لوحة تحكم Admin: لإضافة مستخدمين وسيارات ومعرفات فريدة.

لوحة العميل: تعرض الموقع الجغرافي للمركبة على خريطة Google.

نظام تسجيل دخول: لضمان الخصوصية.

واجهة API (مثل track.php) لاستقبال البيانات من ESP32

رابعاً: قاعدة البيانات

تم استخدام MySQL لتنظيم وحفظ بيانات التتبع:

.users, cars, locations: الجداول تشمل

كل سجل يحتوي على: id, car_identifier, latitude, longitude, timestamp

الفصل السادس:

الاختبار والنتائج

6.1 مرحلة الاختبار :

تهدف هذه المرحلة إلى تقييم كفاءة النظام المقترن من خلال اختبارات ميدانية تغطي أوضاع التشغيل الثلاثة (Real-Time – Offline – SMS)، والتأكد من دقة بيانات GPS، وسلامة إرسال واستقبال البيانات سواء عبر الرسائل القصيرة أو الإنترنت، إضافة إلى اختبار سيناريوهات فقد الاتصال، وقدرة النظام على التعافي وタイミング البيانات المؤجلة.

6.1.1 اختبار الوضع الأول : SMS Mode

الوصف: في هذا الوضع، يعتمد النظام على استقبال أوامر المستخدم عبر رسالة نصية، ويقوم بالرد بإحداثيات الموقع عبر رابط خريطة. لا يتطلب هذا الوضع وجود اتصال بالإنترنت.

إجراءات الاختبار:

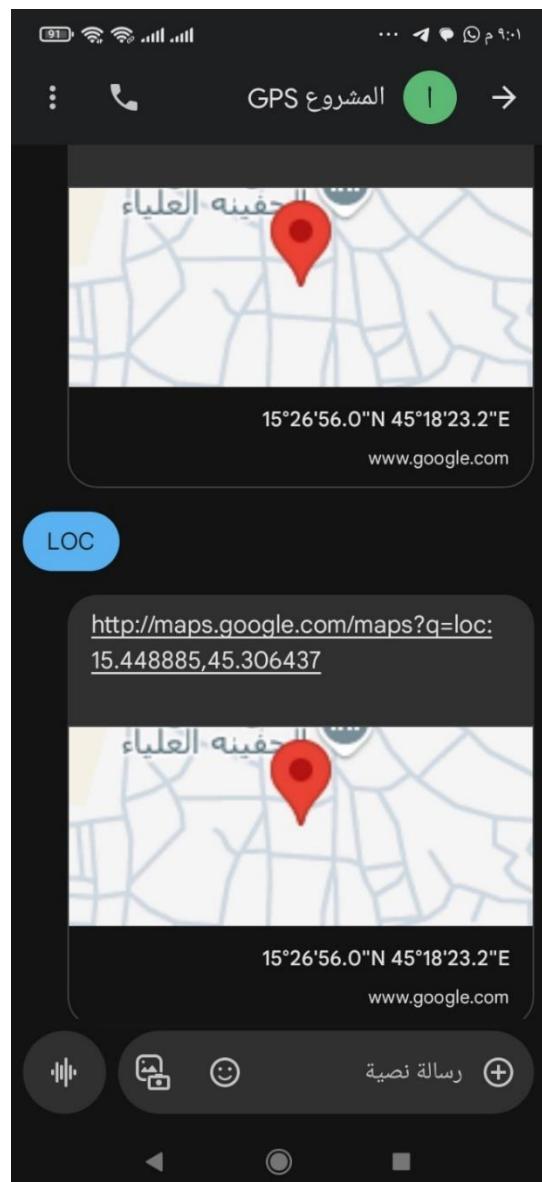
1. إرسال رسالة تحتوي على الأمر LOC إلى شريحة SIM800L.
2. استقبال الرسالة عبر ESP32 وتحليل محتواها.
3. قراءة الإحداثيات من وحدة GPS.
4. تكوين رابط Google Maps وإرساله للمستخدم في رسالة قصيرة SMS

النتائج:

تم استقبال الأوامر بنجاح واستجابة النظام خلال أقل من 5 ثوانٍ.
الإحداثيات كانت دقيقة وفتحت مباشرة في تطبيق Google Maps.
تمت التجربة في بيئات مختلفة (مفتوحة - حضرية) وكانت نتائج GPS دقيقة.
لم تظهر أية مشاكل في إرسال الرسائل في وجود تغطية GSM جيدة.



الشكل رقم (24) يوضح عند النقر والدخول إلى الخريطة



الشكل رقم (25) يوضح إرسال رسالة LOC والرد برباط

صورة في برنامج Arduino Ide تظهر وصول الرسالة للنظام وانه سيرسل الرابط للرقم المضمن بالكود :

```
Serial Monitor x
Message (Enter to send message to 'ESP32 Dev Module' on 'COM7')

+CMPT: "+967778086270","","25/07/01,21:00:48+12"
LOC
sent
>
>

+CMGS: 209
OK
 Skipping send: distance (meters) = 0.00
```

الشكل رقم (26) يوضح صورة تظهر وصول رسالة loc في وضع SMS

6.1.2 اختبار الوضع الثاني: Offline Mode

الوصف: عند فقدان الاتصال بالإنترنت، يحتفظ النظام بالإحداثيات مؤقتاً داخل مصفوفة (`<std::vector<GPSData>>`)، ثم يعيد إرسالها تلقائياً إلى الخادم عند استعادة الاتصال.

إجراءات الاختبار:

1. فصل الاتصال بالواي فاي يدوياً.
2. مراقبة قراءة GPS كل 30 ثانية.
3. فحص عملية تخزين البيانات في المتغير المؤقت.
4. إعادة الاتصال بالواي فاي.
5. مراقبة رفع البيانات المؤجلة إلى السيرفر.

النتائج:

تم تخزين الإحداثيات بشكل دوري خلال فترة انقطاع الاتصال.

لم يتم فقدان أي سجل من البيانات.

عند استعادة الاتصال، تم إرسال كل البيانات إلى الموقع وعرضها على الخريطة تلقائياً.

تمت مزامنة البيانات بدقة مع تسجيل الوقت (`timestamp`) لكل نقطة.

الآن نلاحظ في Arduino ide في برنامج Serial Monitor عند انقطاع الاتصال بالإنترنت يتم التخزين محلياً

```
Final HTTP Response code: -1
Error on HTTP request: connection refused
Error code: -1
Off line mode . Storing data locally.
```

الشكل رقم (27) يوضح لقطة من السيرالي مونيتور توضح انقطاع الاتصال بالإنترنت وتفعيل وضع Off line

```
Final HTTP Response code: -1
Error on HTTP request: connection refused
Error code: -1
Off line mode . Storing data locally.
WiFi not connected. Storing data locally.
E (90707) wifi:sta is connecting, return error
ets Jul 29 2019 12:21:46
```

تُظهر هذه اللقطة من نافذة المراقبة التسلسليّة (Serial Monitor) أن النّظام واجه فشلاً في إرسال البيانات إلى الخادم بسبب فقدان الاتصال بالإنترنت. نتيجة لذلك، قام النّظام تلقائياً بتفعيل وضع التخزين المحلي (Offline Mode)، حيث يتم الاحتفاظ بالإحداثيات الجغرافية في الذاكرة المحليّة لوحدة ESP32.

من أهم الرسائل الموضحة في الصورة:

- Error on HTTP request: connection refused: يشير إلى أن الاتصال بـ track.php لم ينجح.
- Off line mode. Storing data locally.: تأكيد أن النّظام انتقل تلقائياً إلى وضع التخزين المحلي.
- WiFi not connected.: يؤكد أن الشبكة اللاسلكية غير متاحة في تلك اللحظة.

عند عودة الاتصال بالشبكة، يقوم النّظام تلقائياً بإرسال جميع البيانات التي تم تخزينها أثناء فترة الانقطاع، مما يضمن عدم فقدان أي موقع من مسار المركبة.

6.1.3 اختبار الوضع الثالث: Real-Time Mode :

الوصف: يُرسل النّظام الموقع الجغرافي بشكل لحظي كل 30 ثانية إلى الخادم عبر واجهة API مخصصة، حيث يتم تخزين البيانات وعرضها مباشرة على خريطة Google.

إجراءات الاختبار:

1. الاتصال بشبكة Wi-Fi
2. تشغيل النّظام ومراقبة عملية الإرسال.
3. التحقق من نجاح طلبات HTTP (رمز الاستجابة 200/201)
4. فحص ظهور البيانات على لوحة التحكم الخاصة بالمستخدم.

النتائج:

أرسلت البيانات بشكل تلقائي ودون تأخير ملحوظ.

لم يحدث ازدواج في البيانات أو تكرار.

خريطة الموقع أظهرت التتبع اللحظي بنجاح.

النظام تعامل بكفاءة مع إعادة الاتصال التلقائي عند انقطاع الشبكة مؤقتاً.

صورة في برنامج Arduino Ide تظهر نجاح الاتصال بشبكة الواي فاي وانه يرسل للموقع بالاحداثيات

```
Serial Monitor ×

Message (Enter to send message to 'ESP32 Dev Module' on 'COM7')

WiFi connected!
OK
OK

Invalid GPS data. Skipping upload.
Invalid GPS data. Skipping upload.
Invalid GPS data. Skipping upload.
Full GET URL: https://ajyalymen.site/car_system/api/track.php?car_identifier=1&latitude=15.448885&longitude=45.306437&api_key=YemenCars123
Final HTTP Response code: 200
Response: {"status": "success"}
Live MODE ON, SHOW ON MAP.

)
}

Daemon Offline
```

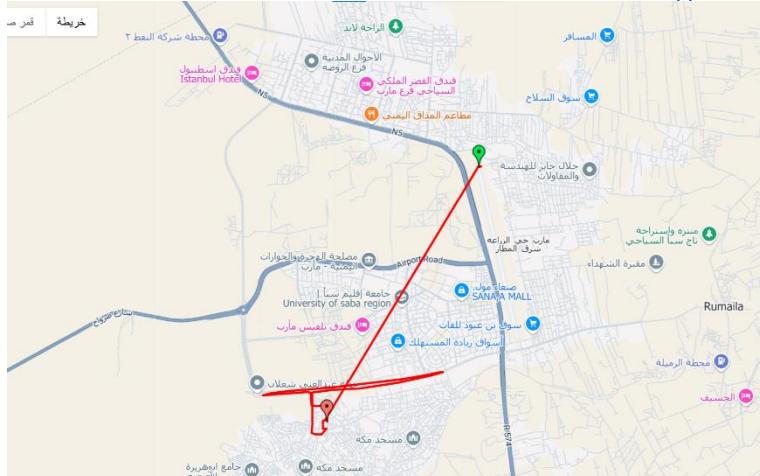
الشكل رقم (28) صورة تظهر اتصال شبكة اويفاي وان الخادم يستجيب ويتم عرض الموقع بموقع الويب

صورة تتبع حقيقة المركبة أثناء الاختبار

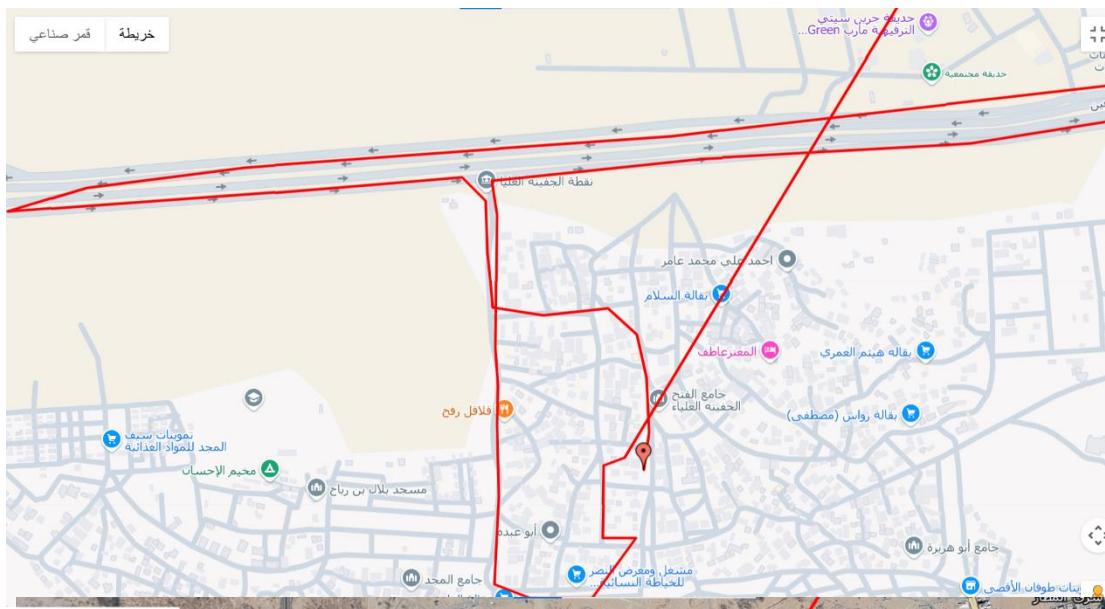
تُظهر هذه الصورة المأخوذة من الخريطة اللحظية للنظام خلال التجربة الحقيقية، المسار الذي قطعته المركبة في بيئة سكنية. تمثل العلامة الخضراء نقطة البداية للمركبة، بينما تشير العلامة الحمراء إلى نقطة التوقف النهائية. أما الخط الأحمر المتصل، فيمثل المسار الذي تتبعه المركبة كما تم تتبعه لحظياً بواسطة النظام.

تم تفعيل وضع التتبع اللحظي من خلال اتصال وحدة Wi-Fi بشبكة ESP32 وإرسال البيانات الجغرافية إلى الخادم الخاص عبر واجهة track.php. بعد التحقق من نجاح الاتصال بالسيفري، تم تتبع السيارة فعليًا، وتحديث موقعها على الخريطة في الزمن الحقيقي باستخدام واجهة الموقع المصممة على الرابط:

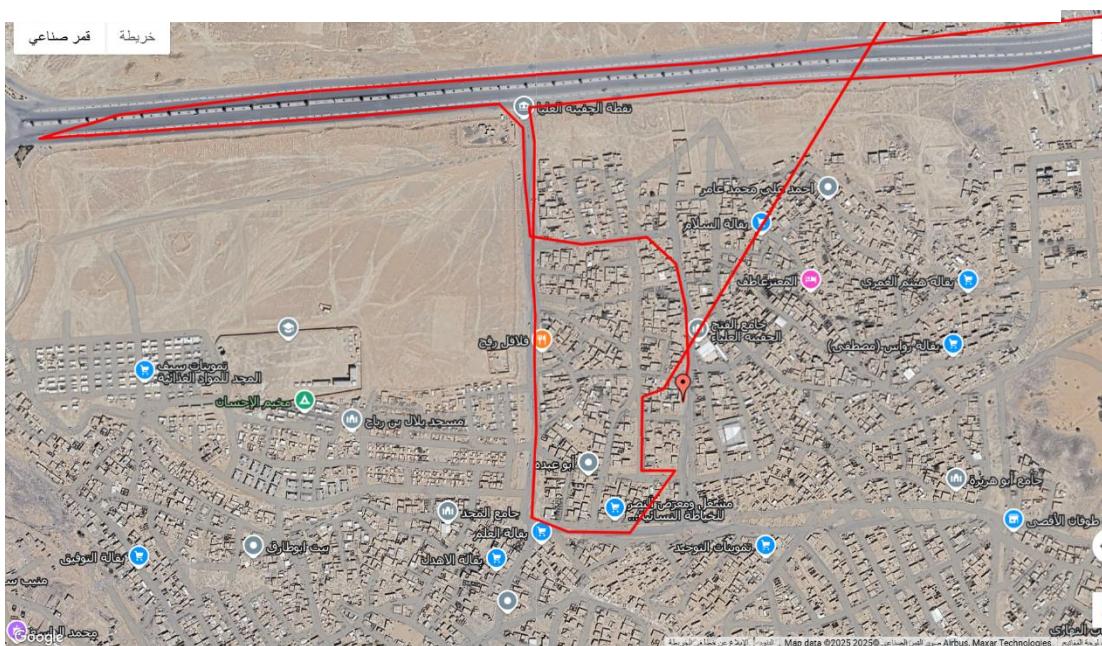
https://ajyalyemen.site/car_system/dashboard



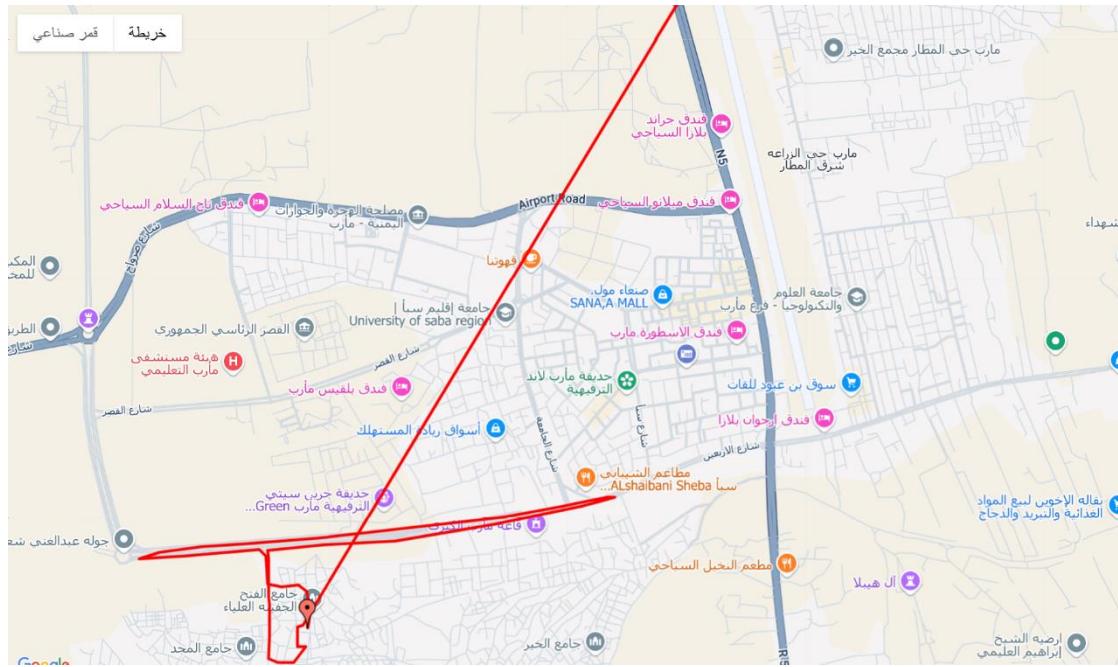
الشكل رقم(29) يوضح صورة تبين تتبع لحظي للمركبة على خريطة الموقع



الشكل رقم (30) يوضح صورة تتبع لحظي للمركبة على خريطة الموقع باستخدام القراء الصناعي



الشكل رقم (31) يوضح صورة أخرى تتبع لحظي للمركبة على خريطة الموقع بأقمار الصناعي



الشكل رقم (32) يوضح صورة أخرى تبين تتبع لحظي للمركبة على خريطة الموقع



الشكل رقم (33) يوضح صورة ثالثة تبين لحظي للمركبة على خريطة الموقع بالقمر الصناعي

6.1.4 اختبار فعالية الوحدات الإلكترونية :

الوحدة	النتيجة
وحدة GPS	حددت الموقع بدقة (حتى 3 أمتار)، خاصة في الأماكن المفتوحة.
وحدة GSM	أرسلت الرسائل بسرعة، ولكن الأداء تأثر في الأماكن ذات تغطية ضعيفة.
وحدة ESP32	عملت بكفاءة عالية طوال فترة التشغيل دون أي تجميد أو إعادة تشغيل.
مزود الطاقة	تغذى النظام من مصدر 12V عبر منظم 7805 بدون مشاكل حرارية أو استقرارية.

جدول 2 : جدول يوضح اختبار فعالية الوحدات الإلكترونية

6.1.5 تحليل النتائج ومقارنة أوضاع التشغيل :

تم تقييم أداء النظام من خلال مقارنة الأوضاع التشغيلية الثلاثة Offline Mode ، SMS Mode و Real-Time Mode ، اعتمادًا على مجموعة من المعايير وهي: زمن الاستجابة، دقة الموقع، الاعتمادية، الحاجة للاتصال بالإنترنت، واستهلاك الطاقة. يوضح الجدول التالي ملخصاً مقارناً لأداء كل وضع:

جدول 3 : جدول يبين مقارنة بين أوضاع التشغيل الثلاثة

المؤشر	SMS Mode	Offline Mode	Real -Time Mode
الحاجة إلى الإنترنٌت	لا	لا (يُخزن محلياً لحين الاتصال)	نعم
زمن الاستجابة	10 – 3 ثوانٍ	يرفع عند عودة الاتصال	7 – 4 ثوانٍ
نسبة النجاح	100%	98%	95%
دقة تحديد الموقع	1.5 – 2 متراً	±1.5 متراً	±1 متراً
الاعتمادية	عالية	عالية	متوسطة (حسب جودة الاتصال)
استهلاك الطاقة	منخفض	منخفض	مرتفع نسبياً

6.1.6 الدروس المستفادة من مرحلة الاختبار :

من خلال إجراء الاختبارات في بيئات متعددة (داخل مبانٍ، مناطق مفتوحة، تحت تغطية شبكة ضعيفة)، تم استخلاص عدد من الدروس المهمة:

1. كفاءة التشغيل متعدد الأوضاع(Hybrid Operation)

منح المشروع قدرة على التكيف مع بيئات الاتصال المختلفة، وهو ما يُعد ميزة تنافسية مقارنة بالأنظمة أحادية النمط.

2. SMS Mode

أثبتت نجاحاً كاملاً في إرسال الإحداثيات بدقة عبر الرسائل النصية فقط، مما يجعله مناسباً للاستخدام في المناطق التي تقترن إلى الإنترنت أو أثناء الطوارئ.

3. Offline Mode

قدم هذا الوضع حلّ عملياً لتخزين البيانات مؤقتاً أثناء انقطاع الاتصال، ثم رفعها لاحقاً دون فقدان أي موقع أو ترتيب زمني.

4. Real-Time Mode

قدم تجربة مستخدم سلسة وفورية في عرض الموقع على الخريطة، إلا أن الأداء كان يتتأثر بسرعة الاتصال وجودته.

5. الاستقرار العام للنظام

أظهر المشروع أداءً مستقراً حتى في ظروف تشغيل صعبة (مثل الحرارة، تحريك الجهاز، أو ضعف الشبكة)، مما يعزز قابليته للتطبيق العملي خارج المختبر.

6.1.7 التقييم العام للنظام :

بناءً على نتائج التحليل الميداني، يمكن القول إن النظام:

- حقق أهدافه الأساسية في توفير وسيلة موثوقة لتتبع المركبات بثلاثة أوضاع ذكية.
- أظهر تكاملاً ممتازاً بين الوحدات الإلكترونية.(ESP32, GPS, GSM)
- قدم واجهة ويب فعالة من حيث الوظيفة وسهولة الاستخدام على الرابط:
https://ajalyemen.site/car_system/dashboard
- سهولة تطويره مستقبلاً لتضمين مزايا إضافية مثل:
عرض السرعة، إشعارات التنبيه، تطبيق موبايل، إلخ.

الفصل السابع :

الخاتمة والمراجع

7.1 الخاتمة العامة للمشروع :

في ختام هذا المشروع، تمكنا من تصميم وتنفيذ نظام تتبع مركبات ذكي وفعال باستخدام وحدة التحكم ESP32، مع الاعتماد على تقنيات GPS وGSM. جاء هذا المشروع استجابةً لحاجة حقيقية في البيئة اليمنية التي تعاني من ضعف تغطية الإنترن特 وشبكات الاتصالات في بعض المناطق، حيث تم بناء نظام قادر على العمل في ثلاثة أوضاع تشغيل مختلفة لضمان الاستمرارية والدقة في تتبع المركبات في جميع الظروف.

7.2 أهم الإنجازات التي تحققت في هذا المشروع :

دعم ثلاثة أوضاع تشغيل متكاملة:

- وضع الرسائل القصيرة (SMS Mode): لتحديد الموقع الجغرافي عند توفر شبكة الاتصالات.
- الوضع غير المتصل (Offline Mode): لحفظ المسارات عند انقطاع الإنترنط، ورفعها لاحقاً عند الاتصال.
- الوضع اللحظي (Real-time Mode): لإرسال البيانات مباشرة إلى موقع الويب ومتابعتها بشكل مباشر كل 30 ثانية.
- تكامل مكونات الأجهزة بنجاح: باستخدام ESP32 كوحدة تحكم، GPS NEO-M8N-0-0 لتحديد الموقع، و SIM800L لإرسال واستقبال الرسائل.
- معالجة مشكلة ضعف الشبكة: عبر نظام هجين لا يعتمد كلياً على الإنترنط، مما يجعله مناسباً للبيئة المحلية خاصة في المناطق الصحراوية أو الجبلية.
- قابلية التوسيعة المستقبلية: مثل تطوير تطبيق موبايل وربط النظام بمنصات خارجية أو استخدام مودمات أقوى مثل 4G أو STARLINK في الأماكن التي لا تتوفر فيها شبكة.

7.3 التحديات التي واجهتنا :

- ضعف تغطية شبكة YOU في بعض الأماكن: مما استدعي تطوير النظام ليعمل أيضًا في الوضع غير المتصل.
- القيود المالية: التي حالت دون تنفيذ بعض المزايا مثل شراء مودم فضائي أو تطوير تطبيق موبايل متكامل في هذه المرحلة. ضعف تغطية شبكة YOU في بعض الأماكن: مما استدعي تطوير النظام ليعمل أيضًا في الوضع غير المتصل.
- القيود المالية: التي حالت دون تنفيذ بعض المزايا مثل شراء مودم فضائي أو تطوير تطبيق موبايل متكامل في هذه المرحلة.
- صعوبة ربط النظام بموقع الويب الذي تم تصميمه بالبداية لعرض الموقع عليه.

7.4 التوصيات والتعديلات المستقبلية :

أولاً: تعزيز قابلية الاتصال في البيئات اليمنية نظراً لضعف تغطية الإنترنت والاتصالات في معظم المناطق اليمنية مقارنةً بدول مثل السعودية، يوصى بما يلي:

استخدام وحدات GSM 4G مثل SIM7600 لتحسين الاتصال في Real-Time Mode .
استخدام أنظمة أقمار صناعية مثل Starlink لتوفير اتصال مستقر في المناطق النائية.
ملاحظة: تم تنفيذ النظام باستخدام نقطة وصول الهاتف المحمول نظراً لضعف الموارد المتاحة.

ثانياً: تحسين أوضاع التشغيل

:SMS Mode

يمكن تحسين تجربة الاستخدام بإرسال رسائل تأكيد بعد تنفيذ الأوامر.

إضافة صلاحيات للمستخدم (مثلاً: من يمكنه إرسال أوامر).

:Offline Mode

استخدام بطاقة تخزين خارجية (SD Card) لتوسيع سعة البيانات المؤقتة.

تشفيير البيانات المخزنة لحماية الخصوصية.

(الوضع اللحظي): Real-Time Mode

النظام في نسخته الحالية يعتمد على موقع ويب تم تطويره بشكل خاص، ويُستضاف على نطاق site ajyalyemen.site بـESP32 بإرسال الإحداثيات الجغرافية عبر واجهة API إلى هذا الموقع باستخدام بروتوكول HTTP، ويتم عرض موقع السيارة لحظياً على الخريطة ضمن لوحة تحكم مخصصة لكل مستخدم.

ملاحظات وتوصيات مستقبلية:

توسيع قدرات لوحة التحكم بإضافة إمكانية تتبع عدة مركبات في نفس الوقت.

تحسين استجابة الموقع في الهاتف المحمولة وتصميم واجهة (Responsive UI).

تمكين المستخدم من تصدير البيانات التاريخية (CSV / Excel) عبر الواجهة.

إضافة خريطة حرارية (Heatmap) للمسارات المتكررة.

ربط الخادم بنظام إشعارات (SMS أو Email) عند دخول المركبة إلى "منطقة غير آمنة".

تحسين أمان API عبر استخدام رموز تحقق (JWT أو OAuth).

توثيق النظام API Documentation ليسهل دمجه مع أنظمة أخرى مستقبلاً.

دعم التبديل التلقائي بين Wi-Fi وGSM لرفع البيانات، حسب توفر الاتصال.

هذه التعديلات تُسهم في ترسيخ استقلالية النظام وتعزيز الاعتماد على منصة محلية تمتلكها وتحكم بها الجهة المنفذة للمشروع.

ثالثاً: تحسين تكامل العتاد (Hardware Integration) :

الوصيات المستقبلية	الوحدة
إضافة هوائي خارجي لتحسين الإشارة داخل المبني أو في الأماكن المحاطة بالحواجز.	GPS neo-m8n-0-10
استبداله بوحدات تدعم الجيل الرابع مثل SIM7600 لضمان تغطية أوسع.	SIM800L
تحديث المكتبات والبرمجيات المستخدمة بانتظام لدعم أداء أفضل واستهلاك أقل للطاقة.	ESP32

جدول 4 : جدول يبين تحسين تكامل العتاد

رابعاً: واجهة المستخدم والتحكم :

تصميم تطبيق موبايل لعرض البيانات دون الحاجة لمتصفح.

لوحة تحكم تفاعلية تسمح بـ:

عرض آخر موقع مباشر.

تصدير سجل المسارات.

التحكم في أوضاع التشغيل.

إدارة الحسابات والمركبات.

خامسًا: حماية وأمان البيانات :

تشفير البيانات المرسلة عبر الرسائل النصية والإنترنت باستخدام خوارزميات خفيفة مثل AES أو XOR.

استخدام API Tokens لربط الأجهزة بالموقع بشكل آمن.

منع الوصول غير المصرح به عبر المصادقة الثانية (2FA) في لوحة التحكم.

خامساً: دمج وضع الرسائل القصيرة (SMS Mode) مع نظام التتبع اللحظي (Real-Time SMS Mode) :

في الوضع الحالي، يعتمد النظام على الاتصال بالإنترنت لإرسال إحداثيات المركبة إلى الخادم وعرضها على الموقع الإلكتروني.

إلا أن هذا الوضع غير فعال في المناطق التي لا تتوفر فيها تغطية إنترنت، مما يجعل نظام التتبع غير نشط في تلك الحالات.

يُقترح في النسخ المستقبلية من النظام تطوير تكامل بين SMS Mode و Real-Time Web Tracking، بحيث يتم:

إرسال رسالة نصية من المستخدم إلى الجهاز تحتوي على كلمة "LOC".

يرد الجهاز برسالة SMS تحتوي على رابط مباشر إلى موقع المركبة على Google Maps. يتم استقبال الرسالة في هاتف ذكي (خاص بالمشرف أو مركز التحكم).

يقوم تطبيق جوال (Android App) أو أداة وسيطة بقراءة الرسالة تلقائياً، واستخراج الإحداثيات من الرابط.

تُرسل هذه الإحداثيات مباشرة إلى الخادم (Hostinger) ليتم عرضها على خريطة النظام في الموقع، كما لو كانت أرسلت من ESP32 مباشرة.

بهذه الطريقة، حتى في حالة انقطاع الإنترت عن المركبة، يمكن استمرار تتبعها من خلال SMS وتحويل الإحداثيات إلى الخادم تلقائياً، مما يوفر موثوقية أعلى للنظام.

7.4 المراجع : (References)

1. Last Minute Engineers. (n.d.). *SIM800L GSM/GPRS Module with Arduino*. Retrieved July 2, 2025, from
<https://lastminuteengineers.com/sim800l-gsm-module-arduino-tutorial>
2. Techiesms)YouTube(.)2023(. Build your own ESP32 GPS tracker with live tracking – Easy | Free API. Retrieved from: <https://www.youtube.com>
3. The Engineering Projects. (2020, December). ESP32 Pinout, Datasheet, Features & Applications. Retrieved from
<https://www.theengineeringprojects.com/2020/12/esp32-pinout-datasheet-features-applications.html>
4. Last Minute Engineers. (n.d.). NEO-6M GPS Module with Arduino – Getting Started. Retrieved from
<https://lastminuteengineers.com/neo6m-gps-arduino-tutorial>

الفصل الثامن :

الملحقات

8.1 الكود المصدر (Source Code) :

```
#include <WiFi.h>
#include <HTTPClient.h>
#include <ArduinoJson.h>
#include <vector> // Include vector library for storing offline GPS data
#include <WiFiClientSecure.h>
#include <math.h>
// WiFi credentials
const char* ssid = "yyytrrew"; // WiFi SSID - Replace your own
const char* password = "715411422777"; // WiFi Password - Replace your own
// Server details for sending GPS data
const char* serverUrl = "https://www.circuitdigest.cloud/geolinker"; // Server URL
const char* apiKey =
"kGegKPqW4Kof"; //12 character API key
for authentication
//GPS module connection using UART1
const int RXPin = 16; // GPS TX connected to ESP32 RX
const int TXPin = 17; // GPS RX connected to ESP32 TX
#define TINY_GSM_MODEM_SIM800
#define SerialMon Serial //for serial mowntionr
#define gpsSerial Serial1 //for comminicate with gps modual
#define SerialAT Serial2 // for comminicate with sim8001 modual
String number = "+967778086270";
String textMessage;

// ESP32 and SIM8001 pins
#define gsm_TX 26
#define gsm_RX 27
// Structure to hold detailed GPS data
struct GPSRawData {
    double latitude;
    char latitudeDir;
    double longitude;
    char longitudeDir;
    int satellites;
    double altitude;
    int hours, minutes, seconds;
    int day, month, year;
    String timestamp; // Combined timestamp with date and time
};
// Structure to store simplified GPS data for transmission
struct GPSData {
    double latitude;
    double longitude;
    String timestamp;
};
String phoneNumber = "";
```

```

String message = "";
bool isReceivingMessage = false;
// Upload interval and last upload timestamp
const unsigned long uploadInterval = 30000; // Data sent every 30 seconds
unsigned long lastUploadTime = 0;
// LED indicators for network and GPS status
bool gpsDataValid = false; // Flag to check if GPS data is valid
GPSData latestGPSData; // Stores the latest valid GPS data
GPSRawData latestGPSRawData; // Stores raw GPS data
// Buffer to store GPS data when offline
std::vector<GPSData> offlineData;
// Add these defines/variables near your WiFi credentials
const char* apiEndpoint =
"https://ajyalyemen.site/car_system/api/track.php"; // Change to your server
URL
const char* carIdentifier = "1"; // Unique identifier for this car
const char* apiKeyWebsite = "YemenCars123"; // Must match the PHP API
// Add these variables to track the last sent coordinates
double lastSentLatitude = 0.0;
double lastSentLongitude = 0.0;
bool hasSentBefore = false;
// Returns distance in meters between two lat/lng points
double haversine(double lat1, double lon1, double lat2, double lon2) {
    const double R = 6371000; // Earth radius in meters
    double dLat = radians(lat2 - lat1);
    double dLon = radians(lon2 - lon1);
    double a = sin(dLat/2) * sin(dLat/2) +
               cos(radians(lat1)) * cos(radians(lat2)) *
               sin(dLon/2) * sin(dLon/2);
    double c = 2 * atan2(sqrt(a), sqrt(1-a));
    return R * c;
}
void setup() {
    SerialMon.begin(115200);
    delay(1000);
    SerialMon.println("Wait ...");
    gpsSerial.begin(9600, SERIAL_8N1, RXPin, TXPin); // Initialize GPS module
    SerialAT.begin(9600, SERIAL_8N1, gsm_TX,gsm_RX); //Initialize GSM module
    delay(7000);
    SerialMon.println(" success");
    SerialAT.println("AT+CMGF=1\r\n"); //Sets the GSM Module in Text Mode
    delay(1000);
    SerialAT.println("AT+CNMI=2,2,0,0,0\r\n");
    delay(1000);
    //SerialAT.println("AT+CMGS=\"" + number + "\"\r"); //Mobile phone number
    to send message
    //delay(1000);
    //String smsMessage = "GPS TRACKING RADY";SSSSSS
    //SerialAT.println(smsMessage);
}

```

```

//delay(100);
//SerialAT.println((char)26); // ASCII code of CTRL+Z
//SerialAT.write(0x1a);
//delay(1000);
// Connect to WiFi
WiFi.begin(ssid, password);
SerialMon.print("Connecting to WiFi");
//rst:0xc (SW_CPU_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
while (WiFi.status() != WL_CONNECTED) {
    SerialMon.print(" not connect");
}
SerialMon.println("\nWiFi connected!");
}

void loop() {
    while (SerialAT.available() > 0) {
        textMessage = SerialAT.readString();
        Serial.println(textMessage);
        // delay(15);
    }
    if (textMessage.indexOf("LOC") >= 0) {
        // delay(100);
        textMessage = "";
        String text;
        String latitudee = String(latestGPSData.latitude, 6);
        String longitudee = String(latestGPSData.longitude, 6);
        text += "http://maps.google.com/maps?q=loc:" + latitudee + "," +
longitudee;

        SerialAT.println("AT+CMGS=\"" + number + "\"\r");
        delay(1000);
        SerialAT.println(text);
        delay(100);
        //SerialAT.println((char)26); // ASCII code of CTRL+Z
        SerialAT.write(0x1a);
        delay(1000);
        SerialAT.println("AT+CMGD=1,4");
        SerialMon.println("sent");
    }
    static String gpsData = "";
    // Read and process GPS data from the module
    while (gpsSerial.available()) {
        char c = gpsSerial.read();
        gpsData += c;
        if (c == '\n') { // When a full GPS sentence is received
            processGPSData(gpsData);
            gpsData = "";
        }
    }
}

```

```

// Periodically send GPS data to the server
if (millis() - lastUploadTime >= uploadInterval) {
    lastUploadTime = millis();
    if (gpsDataValid) { // Only send if GPS data is valid
        // Skip sending if distance is less than threshold
        const double MIN_DISTANCE_METERS = 30.0; // Change as needed
        if (hasSentBefore) {
            double dist = haversine(latestGPSData.latitude,
latestGPSData.longitude, lastSentLatitude, lastSentLongitude);
            if (dist < MIN_DISTANCE_METERS) {
                SerialMon.print("Skipping send: distance (meters) = ");
                SerialMon.println(dist);
                return;
            }
        }
        if (WiFi.status() == WL_CONNECTED) {
            // Send any stored offline data first
            bool offlineUploadSuccess = true;
            for (auto& data: offlineData) {
                if (!sendGPSData(data)) { // Stop if sending fails
                    offlineUploadSuccess = false;
                    break;
                }
            }
            if (offlineUploadSuccess) {
                offlineData.clear(); // Clear offline buffer if all data is
uploaded
            }
        }

        // Send the latest GPS data
        if (!sendGPSData(latestGPSData)) {
            SerialMon.println("Off line mode . Storing data locally.");
            offlineData.push_back(latestGPSData); // Store data for later
upload
        } else {
            SerialMon.println("Live MODE ON, SHOW ON MAP.");
            // Update last sent coordinates
            lastSentLatitude = latestGPSData.latitude;
            lastSentLongitude = latestGPSData.longitude;
            hasSentBefore = true;
        }
    } else {
        // If WiFi is offline, store GPS data for later upload
        SerialMon.println("WiFi not connected. Storing data locally.");
        offlineData.push_back(latestGPSData);
        WiFi.disconnect();
        WiFi.reconnect();
    }
}

```

```

        } else {
            SerialMon.println("Invalid GPS data. Skipping upload.");
        }
    }

}

// Function to send GPS data to the server
bool sendGPSData(GPSData data) {
    HTTPClient http;

    // Build the full URL with parameters for GET
    String fullUrl = String(apiEndpoint) + "?car_identifier=" +
String(carIdentifier) +
        "&latitude=" + String(data.latitude, 6) +
        "&longitude=" + String(data.longitude, 6) +
        "&api_key=" + String(apiKeyWebsite);

    SerialMon.println("Full GET URL: " + fullUrl);

    http.begin(fullUrl);
    http.setFollowRedirects(HTTPC_STRICT_FOLLOW_REDIRECTS);
    http.setUserAgent("Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.36");

    // Set timeout to prevent hanging
    http.setTimeout(10000); // 10 seconds timeout

    int httpResponseCode = http.GET();
    SerialMon.println("Final HTTP Response code: " +
String(httpResponseCode));

    if (httpResponseCode > 0) {
        if(httpResponseCode == 301 || httpResponseCode == 302 ||
httpResponseCode == 307 || httpResponseCode == 308){
            String newUrl = http.getLocation();
            SerialMon.println("Redirected to: " + newUrl);
        }
        String response = http.getString();
        SerialMon.println("Response: " + response);
    } else {
        SerialMon.println("Error on HTTP request: " +
http.errorToString(httpResponseCode));
        SerialMon.println("Error code: " + String(httpResponseCode));
    }

    http.end();
    return (httpResponseCode == 200 || httpResponseCode == 201);
}

```

```

// Function to process received GPS data
void processGPSData(String raw) {
    if (raw.startsWith("$GNGGA")) { // GPGGA contains location and satellite
        parseGPGGA(raw);
        convertAndPrintLocalDateTime(); // Convert UTC to local time
    } else if (raw.startsWith("$GPRMC")) { // GPRMC contains date and speed
        parseGPRMC(raw);
    }
}

// Function to parse GPGGA data and extract latitude, longitude, and
// altitude
void parseGPGGA(String gpgga) {
    String tokens[15];
    int tokenIndex = 0;

    // Split GPGGA string into individual tokens
    int startIndex = 0;
    for (int i = 0; i < gpgga.length(); i++) {
        if (gpgga[i] == ',' || gpgga[i] == '*') {
            tokens[tokenIndex++] = gpgga.substring(startIndex, i);
            startIndex = i + 1;
        }
    }
}

// Extract meaningful GPS data
if (tokenIndex > 1) {
    String utcTime = tokens[1];
    latestGPSRawData.hours = utcTime.substring(0, 2).toInt();
    latestGPSRawData.minutes = utcTime.substring(2, 4).toInt();
    latestGPSRawData.seconds = utcTime.substring(4, 6).toInt();

    latestGPSRawData.latitude = nmeaToDecimal(tokens[2]);
    latestGPSData.latitude = nmeaToDecimal(tokens[2]);
    latestGPSRawData.latitudeDir = tokens[3].charAt(0);
    latestGPSRawData.longitude = nmeaToDecimal(tokens[4]);
    latestGPSData.longitude = nmeaToDecimal(tokens[4]);
    latestGPSRawData.longitudeDir = tokens[5].charAt(0);

    latestGPSRawData.satellites = tokens[7].toInt();
    latestGPSRawData.altitude = tokens[9].toDouble();

    if (latestGPSRawData.satellites >= 4) { // Ensure enough satellites are
        available
        if (latestGPSData.latitude != 0 || latestGPSData.longitude != 0) {
            gpsDataValid = true;
        } else {

```

```

        gpsDataValid = false;
    }
} else {
    gpsDataValid = false;
}
}

// Function to parse GPRMC data and extract the date
void parseGPRMC(String gprmc) {
    String tokens[15];
    int tokenIndex = 0;

    // Split GPRMC string into tokens
    int startIndex = 0;
    for (int i = 0; i < gprmc.length(); i++) {
        if (gprmc[i] == ',' || gprmc[i] == '*') {
            tokens[tokenIndex++] = gprmc.substring(startIndex, i);
            startIndex = i + 1;
        }
    }
}

// Extract date information
if (tokenIndex > 9) {
    String utcDate = tokens[9];
    latestGPSRawData.day = utcDate.substring(0, 2).toInt();
    latestGPSRawData.month = utcDate.substring(2, 4).toInt();
    latestGPSRawData.year = 2000 + utcDate.substring(4, 6).toInt();
}
}

// Function to convert UTC time and date to local time and update the
// timestamp
void convertAndPrintLocalDateTime() {
    int offsetHours = 3;
    int offsetMinutes = 1;
    latestGPSRawData.minutes += offsetMinutes;
    latestGPSRawData.hours += offsetHours;
    if (latestGPSRawData.minutes >= 60) {
        latestGPSRawData.minutes -= 60;
        latestGPSRawData.hours++;
    }
    if (latestGPSRawData.hours >= 24) {
        latestGPSRawData.hours -= 24;
        latestGPSRawData.day++;
    }
    char timeBuffer[20];
    sprintf(timeBuffer, sizeof(timeBuffer), "%04d-%02d-%02d %02d:%02d:%02d",

```

```

        latestGPSRawData.year, latestGPSRawData.month,
latestGPSRawData.day,
        latestGPSRawData.hours, latestGPSRawData.minutes,
latestGPSRawData.seconds);
latestGPSRawData.timestamp = String(timeBuffer);
latestGPSData.timestamp = String(timeBuffer);

//SerialMon.print("TimeStamp: ");
//SerialMon.println(latestGPSRawData.timestamp);
}

// Function to determine the number of days in a given month
int daysInMonth(int month, int year) {
    if (month == 2) {
        // Check for leap year
        return (year % 4 == 0 && (year % 100 != 0 || year % 400 == 0)) ? 29 :
28;
    }
    return (month == 4 || month == 6 || month == 9 || month == 11) ? 30 : 31;
}
// Function to convert NMEA coordinates to decimal degrees
double nmeaToDecimal(String nmeaCoord) {
    if (nmeaCoord == "") return 0.0;
    double raw = nmeaCoord.toDouble();
    int degrees = int(raw / 100);
    double minutes = raw - (degrees * 100);
    return degrees + (minutes / 60.0);
}

```

8.2 ا��اد الموقع الالكتروني (Web Site) :

يحتوي هذا الملحق على أهم ملفات موقع الويب البرمجي الذي يستقبل بيانات GPS من وحدة MySQL32 ويعرضها لحظياً على خريطة تفاعلية. الموقع مبني بلغة PHP، ويعتمد على قاعدة بيانات، إضافة إلى Google Maps API لعرض الموقع الجغرافي للمركبات.

سنقوم بعرض وشرح أهم ملفات المشروع التالية:

track.php .1

admin_dashboard.php .2

customer_dashboard.php .3

login.php .4

logout.php .5

: ESP32 - استقبال البيانات من track.php .1

```
<?php
// Database connection
$mysqli = new mysqli("localhost", "u222784783_ahmed_car", "Qweff112233+", "u222784783_car_system");

if ($mysqli->connect_errno) {
    http_response_code(500);
    echo json_encode(["status" => "error", "message" => "Database connection failed"]);
    exit;
}

// Set your API key here (must match what the ESP32 sends)
$API_KEY = "YemenCars123";

// Get POST data
$car_identifier = $_GET['car_identifier'] ?? '';
$latitude = $_GET['latitude'] ?? '';
$longitude = $_GET['longitude'] ?? '';
$timestamp = $_GET['timestamp'] ?? date('Y-m-d H:i:s');
$api_key = $_GET['api_key'] ?? '';

// Input validation
if ($api_key !== $API_KEY) {
    http_response_code(401);
    echo json_encode(["status" => "error", "message" => "Invalid API key"]);
    exit;
}
```

```

}

if (!$car_identifier || !is_numeric($latitude) || !is_numeric($longitude)) {
    http_response_code(400);
    echo json_encode(["status" => "error", "message" => "Missing or invalid
parameters"]);
    exit;
}

// Find car_id by car_identifier
$stmt = $mysqli->prepare("SELECT id FROM cars WHERE car_identifier = ?");
$stmt->bind_param("s", $car_identifier);
$stmt->execute();
$stmt->bind_result($car_id);
if (!$stmt->fetch()) {
    http_response_code(404);
    echo json_encode(["status" => "error", "message" => "Car not found"]);
    exit;
}
$stmt->close();

// Insert Location
$stmt = $mysqli->prepare("INSERT INTO locations (car_id, latitude,
longitude, timestamp) VALUES (?, ?, ?, ?)");
$stmt->bind_param("idss", $car_id, $latitude, $longitude, $timestamp);
if ($stmt->execute()) {
    echo json_encode(["status" => "success"]);
} else {
    http_response_code(500);
    echo json_encode(["status" => "error", "message" => "Failed to insert
location"]);
}
$stmt->close();
$mysqli->close();
?>

```

هذا الملف هو نقطة الدخول التي تتصل بها وحدة ESP32.

يستقبل البيانات عبر GET من خلال عنوان URL.

يتتحقق من صحة مفتاح API ومعرف السيارة، ثم يدخل الإحداثيات في جدول locations.

- ملف admin_dashboard.php - لوحة تحكم المدير :

```
<?php
session_start();
if (!isset($_SESSION['user_id']) || $_SESSION['role'] !== 'admin') {
    header('Location: index.php');
    exit;
}
// $mysqli = new mysqli("localhost", "root", "", "car_system");
$mysqli = new mysqli("localhost", "u222784783_ahmed_car", "Qweff112233+", "u222784783_car_system");

$error_message = '';
$success_message = '';

// إضافة مستخدم جديد
if (isset($_POST['add_user'])) {
    $name = trim($_POST['name']);
    $email = trim($_POST['email']);
    $password = password_hash($_POST['password'], PASSWORD_DEFAULT);
    $role = $_POST['role'];

    التحقق من عدم وجود البريد الإلكتروني مسبقاً //
    $check_stmt = $mysqli->prepare("SELECT id FROM users WHERE email = ?");
    $check_stmt->bind_param("s", $email);
    $check_stmt->execute();
    $check_stmt->store_result();
    if ($check_stmt->num_rows > 0) {
        $error_message = 'البريد الإلكتروني موجود مسبقاً';
    } else {
        $stmt = $mysqli->prepare("INSERT INTO users (name, email, password, role) VALUES (?, ?, ?, ?)");
        $stmt->bind_param("ssss", $name, $email, $password, $role);
        if ($stmt->execute()) {
            $success_message = 'تم إضافة المستخدم بنجاح';
        } else {
            $error_message = 'خطأ في إضافة المستخدم';
        }
        $stmt->close();
    }
    $check_stmt->close();
}

// تعديل مستخدم
if (isset($_POST['edit_user'])) {
    $user_id = intval($_POST['user_id']);
    $name = trim($_POST['name']);
    $email = trim($_POST['email']);
    $role = $_POST['role'];
}
```

```

        التحقق من عدم وجود البريد الإلكتروني مسبقاً (باستثناء المستخدم الحالي) //
$check_stmt = $mysqli->prepare("SELECT id FROM users WHERE email = ? AND
id != ?");
$check_stmt->bind_param("si", $email, $user_id);
$check_stmt->execute();
$check_stmt->store_result();
if ($check_stmt->num_rows > 0) {
    $error_message = 'البريد الإلكتروني موجود مسبقاً';
} else {
    if (!empty($_POST['password'])) {
        $password = password_hash($_POST['password'], PASSWORD_DEFAULT);
        $stmt = $mysqli->prepare("UPDATE users SET name=?, email=?,
password=?, role=? WHERE id=?");
        $stmt->bind_param("ssssi", $name, $email, $password, $role,
$user_id);
    } else {
        $stmt = $mysqli->prepare("UPDATE users SET name=?, email=?,
role=? WHERE id=?");
        $stmt->bind_param("sssi", $name, $email, $role, $user_id);
    }
    if ($stmt->execute()) {
        $success_message = 'تم تحديث بيانات المستخدم بنجاح';
    } else {
        $error_message = 'خطأ في تحديث بيانات المستخدم';
    }
    $stmt->close();
}
$check_stmt->close();
}

// حذف مستخدم
if (isset($_POST['delete_user'])) {
    $user_id = intval($_POST['user_id']);
    if ($mysqli->query("DELETE FROM users WHERE id = $user_id")) {
        $success_message = 'تم حذف المستخدم بنجاح';
    } else {
        $error_message = 'خطأ في حذف المستخدم';
    }
}

// إضافة سيارة جديدة
if (isset($_POST['add_car'])) {
    $car_name = trim($_POST['car_name']);
    $car_identifier = trim($_POST['car_identifier']);
    $user_id = intval($_POST['user_id']);

        التتحقق من عدم وجود معرف السيارة مسبقاً

```

```

    $check_stmt = $mysqli->prepare("SELECT id FROM cars WHERE car_identifier
= ?");
    $check_stmt->bind_param("s", $car_identifier);
    $check_stmt->execute();
    $check_stmt->store_result();
    if ($check_stmt->num_rows > 0) {
        $error_message = '!معرف السيارة موجود مسبقاً';
    } else {
        $stmt = $mysqli->prepare("INSERT INTO cars (car_name,
car_identifier, user_id) VALUES (?, ?, ?)");
        $stmt->bind_param("ssi", $car_name, $car_identifier, $user_id);
        if ($stmt->execute()) {
            $success_message = 'تم إضافة السيارة بنجاح';
        } else {
            $error_message = '!خطأ في إضافة السيارة';
        }
        $stmt->close();
    }
    $check_stmt->close();
}

// تعديل سيارة
if (isset($_POST['edit_car'])) {
    $car_id = intval($_POST['car_id']);
    $car_name = trim($_POST['car_name']);
    $car_identifier = trim($_POST['car_identifier']);
    $user_id = intval($_POST['user_id']);

    // التحقق من عدم وجود معرف السيارة مسبقاً (باستثناء السيارة الحالية)
    $check_stmt = $mysqli->prepare("SELECT id FROM cars WHERE car_identifier
= ? AND id != ?");
    $check_stmt->bind_param("si", $car_identifier, $car_id);
    $check_stmt->execute();
    $check_stmt->store_result();
    if ($check_stmt->num_rows > 0) {
        $error_message = '!معرف السيارة موجود مسبقاً';
    } else {
        $stmt = $mysqli->prepare("UPDATE cars SET car_name=?, car_identifier=?,
user_id=? WHERE id=?");
        $stmt->bind_param("ssii", $car_name, $car_identifier, $user_id,
$car_id);
        if ($stmt->execute()) {
            $success_message = 'تم تحديث بيانات السيارة بنجاح';
        } else {
            $error_message = '!خطأ في تحديث بيانات السيارة';
        }
        $stmt->close();
    }
    $check_stmt->close();
}

```

```

}

// حذف سيارة
if (isset($_POST['delete_car'])) {
    $car_id = intval($_POST['car_id']);
    if ($mysqli->query("DELETE FROM cars WHERE id = $car_id")) {
        $success_message = 'تم حذف السيارة بنجاح';
    } else {
        $error_message = 'خطأ في حذف السيارة';
    }
}

// جلب المستخدمين
$users = $mysqli->query("SELECT * FROM users");
// جلب السيارات مع اسم المستخدم
$cars = $mysqli->query("SELECT cars.*, users.name as user_name FROM cars
JOIN users ON cars.user_id = users.id");
?>
<!DOCTYPE html>
<html lang="ar" dir="rtl">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>لوحة تحكم المدير</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap rtl.min.css" rel="stylesheet">
</head>
<body>
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
    <div class="container-fluid">
        <a class="navbar-brand" href="#">لوحة تحكم المدير</a>
        <div class="d-flex">
            <a href="logout.php" class="btn btn-outline-light">تسجيل الخروج</a>
        </div>
    </div>
</nav>
<div class="container mt-5">
    <?php if ($error_message): ?>
        <div class="alert alert-danger alert-dismissible fade show"
role="alert">
            <?= $error_message ?>
            <button type="button" class="btn-close" data-bs-
dismiss="alert"></button>
        </div>
    <?php endif; ?>
    <?php if ($success_message): ?>
        <div class="alert alert-success alert-dismissible fade show"
role="alert">

```

```

        <?= $success_message ?>
            <button type="button" class="btn-close" data-bs-
dismiss="alert"></button>
        </div>
    <?php endif; ?>
<h1>!مرحباً بك، مدير النظام</h1>
<p>. هنا يمكنك إدارة المستخدمين والسيارات وعرض جميع بيانات التتبع</p>
<div class="row">
    <div class="col-md-6">
        <h3>ادارة المستخدمين</h3>
        <button class="btn btn-success mb-2" data-bs-toggle="modal"
data-bs-target="#addUserModal">إضافة مستخدم</button>
        <table class="table table-bordered table-striped">
            <thead>
                <tr>
                    <th>الاسم</th>
                    <th>البريد الإلكتروني</th>
                    <th>الدور</th>
                    <th>إجراءات</th>
                </tr>
            </thead>
            <tbody>
                <?php $users->data_seek(0); while($user = $users-
>fetch_assoc()): ?>
                <tr>
                    <td><?= htmlspecialchars($user['name']) ?></td>
                    <td><?= htmlspecialchars($user['email']) ?></td>
                    <td><?= $user['role'] === 'admin' ? 'مدير' : 'عميل' ?>
                    </td>
                    <td>
                        <!-- تعديل -->
                        <button class="btn btn-warning btn-sm" data-bs-
toggle="modal" data-bs-target="#editUserModal<?= $user['id']
?>">تعديل</button>
                        <!-- حذف -->
                        <?php if($_SESSION['user_id'] != $user['id']){
?>
                            <form method="POST" style="display:inline-block"
onsubmit="return confirm('هل أنت متأكد من الحذف؟');">
                                <input type="hidden" name="user_id"
value="<?= $user['id'] ?>">
                                <button type="submit" name="delete_user"
class="btn btn-danger btn-sm">حذف</button>
                            </form>
                        <?php } ?>
                    </td>
                </tr>
            <!-- Modal مستخدم -->

```

```

        <div class="modal fade" id="editUserModal<?= $user['id'] ?>" tabindex="-1" aria-labelledby="editUserModalLabel<?= $user['id'] ?>" aria-hidden="true">
            <div class="modal-dialog">
                <div class="modal-content">
                    <form method="POST">
                        <div class="modal-header">
                            <h5 class="modal-title">تعديل بيانات المستخدم<?= $user['id'] ?></h5>
                            <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="إغلاق"></button>
                        </div>
                        <div class="modal-body">
                            <input type="hidden" name="user_id" value="<?=$user['id'] ?>">
                            <div class="mb-3">
                                <label class="form-label">الاسم</label>
                                <input type="text" class="form-control" name="name" value="<?= htmlspecialchars($user['name']) ?>" required>
                            </div>
                            <div class="mb-3">
                                <label class="form-label">البريد الإلكتروني</label>
                                <input type="email" class="form-control" name="email" value="<?= htmlspecialchars($user['email']) ?>" required>
                            </div>
                            <div class="mb-3">
                                <label class="form-label">كلمة المرور الجديدة (اتركها فارغة إذا لا تريد التغيير)</label>
                                <input type="password" class="form-control" name="password" minlength="6" required>
                            </div>
                            <div class="mb-3">
                                <label class="form-label">الدور</label>
                                <select class="form-select" name="role" required>
                                    <option value="customer" <?=$user['role'] === 'customer' ? 'selected' : '' ?>>عميل</option>
                                    <option value="admin" <?=$user['role'] === 'admin' ? 'selected' : '' ?>>مدير</option>
                                </select>
                            </div>
                            <div class="modal-footer">
                                <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">إلغاء</button>
                                <button type="submit" name="edit_user" class="btn btn-primary">حفظ التعديلات</button>
                            </div>
                        </div>
                    </form>
                </div>
            </div>
        </div>
    
```

```

                </div>
            </div>
        </div>
        <?php endwhile; ?>
    </tbody>
</table>
</div>
<div class="col-md-6">
    <h3>إدارة السيارات</h3>
    <button class="btn btn-success mb-2" data-bs-toggle="modal"
data-bs-target="#addCarModal">إضافة سيارة</button>
    <table class="table table-bordered table-striped">
        <thead>
            <tr>
                <th>اسم السيارة</th>
                <th>معرف السيارة</th>
                <th>المستخدم</th>
                <th>إجراءات</th>
            </tr>
        </thead>
        <tbody>
            <?php $cars->data_seek(0); while($car = $cars-
>fetch_assoc()): ?>
            <tr>
                <td><?= htmlspecialchars($car['car_name']) ?></td>
                <td><?= htmlspecialchars($car['car_identifier']) ?>
                <td><?= htmlspecialchars($car['user_name']) ?></td>
                <td>
                    <!-- عرض الموقع -->
                    <a href="admin_car_locations.php?car_id=<?=
$car['id'] ?>" class="btn btn-info btn-sm">عرض الموقع</a>
                    <!-- تعديل -->
                    <button class="btn btn-warning btn-sm" data-bs-
toggle="modal" data-bs-target="#editCarModal<?= $car['id'] ?>">تعديل</button>
                    <!-- حذف -->
                    <form method="POST" style="display:inline-block"
onsubmit="return confirm('هل أنت متأكد من الحذف؟')">
                        <input type="hidden" name="car_id"
value="<?= $car['id'] ?>">
                        <button type="submit" name="delete_car"
class="btn btn-danger btn-sm">حذف</button>
                    </form>
                </td>
            </tr>
        <!-- Modal تعديل سيارة -->
        <div class="modal fade" id="editCarModal<?= $car['id'] ?>" tabindex="-1" aria-labelledby="editCarModalLabel<?= $car['id'] ?>" aria-
hidden="true">

```

```


<form method="POST">
<div class="modal-header">
<h5 class="modal-title">
تعديل بيانات السيارة</h5>
<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="إغلاق"></button>
</div>
<div class="modal-body">
<input type="hidden" name="car_id" value="<?=$car['id'] ?>">
<div class="mb-3">
<label class="form-label">اسم السيارة</label>
<input type="text" class="form-control" name="car_name" value="<?= htmlspecialchars($car['car_name']) ?>" required>
</div>
<div class="mb-3">
<label class="form-label">معرف السيارة (يجب أن يكون فريد)</label>
<input type="text" class="form-control" name="car_identifier" value="<?= htmlspecialchars($car['car_identifier']) ?>" required>
</div>
<div class="mb-3">
<label class="form-label">المستخدم</label>
<select class="form-select" name="user_id" required>
<?php
$users2 = $mysqli->query("SELECT id, name
FROM users WHERE role = 'customer'");
while($u = $users2->fetch_assoc()): ?>
<option value="<?= $u['id'] ?>" <?= $car['user_id']==$u['id']?'selected':' ?>><?= htmlspecialchars($u['name']) ?></option>
<?php endwhile; ?>
</select>
</div>
</div>
<div class="modal-footer">
<button type="button" class="btn btn-secondary" data-bs-dismiss="modal">إلغاء</button>
<button type="submit" name="edit_car" class="btn btn-primary">حفظ التعديلات</button>
</div>
</form>
</div>
</div>
</div>


```

```

        <?php endwhile; ?>
    </tbody>
</table>
</div>
</div>
<!-- إضافة مستخدم -->
<div class="modal fade" id="addUserModal" tabindex="-1" aria-
labelledby="addUserModalLabel" aria-hidden="true">
    <div class="modal-dialog">
        <div class="modal-content">
            <form method="POST">
                <div class="modal-header">
                    <h5 class="modal-title" id="addUserModalLabel">إضافة مستخدم جديد</h5>
                    <button type="button" class="btn-close" data-bs-dismiss="modal"
aria-label="غلق"></button>
                </div>
                <div class="modal-body">
                    <div class="mb-3">
                        <label class="form-label">الاسم</label>
                        <input type="text" class="form-control" name="name" required>
                    </div>
                    <div class="mb-3">
                        <label class="form-label">البريد الإلكتروني</label>
                        <input type="email" class="form-control" name="email" required>
                    </div>
                    <div class="mb-3">
                        <label class="form-label">كلمة المرور</label>
                        <input type="password" class="form-control" name="password"
required minlength="6">
                    </div>
                    <div class="mb-3">
                        <label class="form-label">الدور</label>
                        <select class="form-select" name="role" required>
                            <option value="customer">عميل</option>
                            <option value="admin">مدير</option>
                        </select>
                    </div>
                </div>
                <div class="modal-footer">
                    <button type="button" class="btn btn-secondary" data-bs-
dismiss="modal">إلغاء</button>
                    <button type="submit" name="add_user" class="btn btn-
primary">إضافة</button>
                </div>
            </form>
        </div>
    </div>
</div>

```

```

<!-- Modal -->
<div class="modal fade" id="addCarModal" tabindex="-1" aria-
labelledby="addCarModalLabel" aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <form method="POST">
        <div class="modal-header">
          <h5 class="modal-title" id="addCarModalLabel">إضافة سيارة جديدة</h5>
          <button type="button" class="btn-close" data-bs-dismiss="modal"
aria-label="غلق"></button>
        </div>
        <div class="modal-body">
          <div class="mb-3">
            <label class="form-label">اسم السيارة</label>
            <input type="text" class="form-control" name="car_name"
required>
          </div>
          <div class="mb-3">
            <label class="form-label">معرف السيارة (يجب أن يكون فريداً)</label>
            <input type="text" class="form-control" name="car_identifier"
required>
          </div>
          <div class="mb-3">
            <label class="form-label">المستخدم</label>
            <select class="form-select" name="user_id" required>
              <?php
                $users2 = $mysqli->query("SELECT id, name FROM users WHERE
role = 'customer'");
                while($u = $users2->fetch_assoc()): ?>
                  <option value="<?= $u['id'] ?>"><?=
htmlspecialchars($u['name']) ?></option>
                <?php endwhile; ?>
            </select>
          </div>
        <div class="modal-footer">
          <button type="button" class="btn btn-secondary" data-bs-
dismiss="modal">إلغاء</button>
          <button type="submit" name="add_car" class="btn btn-
primary">إضافة</button>
        </div>
      </form>
    </div>
  </div>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.m
in.js"></script>
</body>

```

```
</html>
```

يتم استرجاع قائمة المستخدمين والسيارات من قاعدة البيانات لعرضها في واجهة HTML.

الواجهة الرسمية:

تتضمن جداول:

لعرض معلومات المستخدمين.

لعرض السيارات.

أزرار: تعديل / حذف / إضافة / عرض المواقع.

3- ملف customer_dashboard.php – لوحة تحكم العميل :

```
<?php
session_start();
if (!isset($_SESSION['user_id']) || $_SESSION['role'] !== 'customer') {
    header('Location: index.php');
    exit;
}
?>
<!DOCTYPE html>
<html lang="ar" dir="rtl">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>لوحة تحكم العميل</title>
    <link
        href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap rtl.min.css"
        rel="stylesheet">
    <script
        src="https://maps.googleapis.com/maps/api/js?key=AIzaSyCrRFrgnU5kENw45gvHaTJQL5yXGUw1zoc"></script>
</head>
<body>
<nav class="navbar navbar-expand-lg navbar-dark bg-primary">
    <div class="container-fluid">
        <a class="navbar-brand" href="#">لوحة تحكم العميل</a>
        <div class="d-flex">
            <a href="logout.php" class="btn btn-outline-light">تسجيل الخروج</a>
        </div>
    </div>
</nav>
<div class="container mt-5">
    <h1>مرحبا بك!</h1>
```

```

<form class="row g-3 mb-4" method="GET" action="customer_dashboard.php">
    <div class="col-md-6">
        <label for="car_id" class="form-label">اختر سيارتك</label>
        <select class="form-select" id="car_id" name="car_id" required>
            <option value="">اختر السيارة...</option>
            <?php
                جلب السيارات المرتبطة بالمستخدم // جلب السيارات المرتبطة بالمستخدم
                $mysqli = new mysqli("localhost", "u222784783_ahmed_car",
                "Qweff112233+", "u222784783_car_system");
                $user_id = $_SESSION['user_id'];
                $result = $mysqli->query("SELECT id, car_name FROM cars
                WHERE user_id = $user_id");
                while ($row = $result->fetch_assoc()) {
                    $selected = (isset($_GET['car_id'])) && $_GET['car_id']
                    == $row['id']) ? 'selected' : '';
                    echo "<option value='" . $row['id'] . "' $selected>" .
                    htmlspecialchars($row['car_name']) . "</option>";
                }
            ?>
        </select>
    </div>
    <div class="col-md-6">
        <label for="date" class="form-label">تاریخ (اختیاری)</label>
        <input type="date" class="form-control" id="date" name="date"
        value="<?php echo htmlspecialchars($_GET['date'] ?? ''); ?>">
    </div>
    <div class="col-12">
        <button type="submit" class="btn btn-success">بحث</button>
        <?php if (isset($_GET['car_id'])) && $_GET['car_id'] != ''): ?>
            <button type="button" class="btn btn-secondary"
            onclick="exportCSV()">تصدیر سجل المواقع</button>
            <?php endif; ?>
    </div>
</form>
<div id="map" style="height: 400px; width: 100%;"></div>
<h3 class="mt-4">سجل المواقع</h3>
<table class="table table-bordered table-striped mb-4"
id="historyTable">
    <thead>
        <tr>
            <th>التاریخ والوقت</th>
            <th>خط العرض</th>
            <th>خط الطول</th>
        </tr>
    </thead>
    <tbody>
        <?php
            $locations = [];
            if (isset($_GET['car_id'])) && $_GET['car_id'] != '' ) {

```

```

        $car_id = intval($_GET['car_id']);
        $date_filter = isset($_GET['date']) && $_GET['date'] != '' ?
$_GET['date'] : null;
        $query = "SELECT latitude, longitude, timestamp FROM locations
WHERE car_id = $car_id";
        if ($date_filter) {
            $query .= " AND DATE(timestamp) = '" . $mysqli-
>real_escape_string($date_filter) . "'";
        }
        $query .= " ORDER BY timestamp ASC LIMIT 100";
        $result = $mysqli->query($query);
        while ($row = $result->fetch_assoc()) {
            $locations[] = $row;
            echo "<tr><td>" . htmlspecialchars($row['timestamp']) .
"</td><td>" . htmlspecialchars($row['latitude']) . "</td><td>" .
htmlspecialchars($row['longitude']) . "</td></tr>";
        }
        $latest = count($locations) > 0 ? $locations[count($locations)-
1] : null;
    } else {
        $latest = null;
    }
?>
</tbody>
</table>
</div>
<script>
function initMap() {
    var map = new google.maps.Map(document.getElementById('map'), {
        center: {lat: 15.3694, lng: 44.1910},
        zoom: 7
    });
    var route = [];
    <?php if (!empty($locations)) {
        foreach ($locations as $loc) {
            echo "route.push({lat: " . floatval($loc['latitude']) . ", lng:
" . floatval($loc['longitude']) . "});\n";
        }
    }
    if (isset($latest) && $latest) { ?>
        var marker = new google.maps.Marker({
            position: {lat: <?php echo $latest['latitude']; ?>, lng: <?php
echo $latest['longitude']; ?>},
            map: map,
            title: 'آخر موقع للسيارة'
        );
        map.setCenter(marker.getPosition());
        map.setZoom(15);
    <?php }
}

```

```

?>
if (route.length > 1) {
    var polyline = new google.maps.Polyline({
        path: route,
        geodesic: true,
        strokeColor: '#FF0000',
        strokeOpacity: 1.0,
        strokeWeight: 3
    });
    polyline.setMap(map);
}
}
google.maps.event.addDomListener(window, 'load', initMap);

function exportCSV() {
    var table = document.getElementById('historyTable');
    var rows = Array.from(table.querySelectorAll('tr'));
    var csv = [];
    for (var i = 0; i < rows.length; i++) {
        var cols = rows[i].querySelectorAll('th,td');
        var row = [];
        for (var j = 0; j < cols.length; j++) {
            row.push('"' + cols[j].innerText.replace(/\//g, '') + '"');
        }
        csv.push(row.join(','));
    }
    var csvFile = new Blob([csv.join('\n')], {type: 'text/csv'});
    var downloadLink = document.createElement('a');
    downloadLink.download = 'location_history.csv';
    downloadLink.href = window.URL.createObjectURL(csvFile);
    downloadLink.style.display = 'none';
    document.body.appendChild(downloadLink);
    downloadLink.click();
    document.body.removeChild(downloadLink);
}
</script>
</body>
</html>

```

يتيح هذا الملف للعميل تسجيل الدخول إلى حسابه الشخصي ومتابعة موقع سيارته على الخريطة اللحظية، مع إمكانية تصفح سجل الإحداثيات السابقة حسب التاريخ وعدد النقاط، بالإضافة إلى تصدير البيانات إلى ملف .CSV.

1. حماية الصفحة والتحقق من جلسة العميل

التأكد من أن المستخدم قام بتسجيل الدخول.

يتم حصر الوصول على العملاء فقط، وليس المديرين.

2. الاتصال بقاعدة البيانات واسترجاع السيارات الخاصة بالعميل

يقوم بجلب كل السيارات المرتبطة بالمستخدم الحالي من جدول cars.

3. اختيار السيارة والتاريخ وعدد النقاط

يتم التقاط مدخلات المستخدم من واجهة الموقع (اختيار السيارة، التاريخ، وعدد الإحداثيات).

4. استعلام قاعدة البيانات لجلب الموقع

استعلام ديناميكي يسمح بتصفية البيانات حسب التاريخ وعدد النقاط.

5. عرض الخريطة باستخدام Google Maps

يتم رسم المسار الكامل للمركبة بناءً على بيانات الموقع.

تُضاف علامات البداية والنهاية بلون مختلف.

6. تصدير البيانات إلى ملف CSV

توفر الواجهة زرًا لتصدير سجل الموقع للعميل لاستخدامه لاحقًا.

7. واجهة HTML (مختصرة)

جدول يعرض:

الوقت (timestamp)

خط العرض (latitude)

خط الطول (longitude)

زر "عرض" على Google Maps

وواجهة اختيار التاريخ وعدد النقاط.

زر تصدير إلى CSV.

4- login.php – التحقق من تسجيل الدخول

```
<?php

session_start();
// Database connection
$mysqli = new mysqli("localhost", "u222784783_ahmed_car", "Qweff112233+",
"u222784783_car_system");
if ($mysqli->connect_errno) {
    die("<div class='alert alert-danger'>فشل الاتصال بقاعدة البيانات</div>");
}

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $email = trim($_POST['email'] ?? '');
    $password = $_POST['password'] ?? '';
    $error = '';
    if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        $error = 'البريد الإلكتروني غير صحيح';
    } elseif (strlen($password) < 6) {
        $error = ' يجب أن تكون كلمة المرور 6 أحرف على الأقل';
    } else {
        $stmt = $mysqli->prepare("SELECT id, password, role FROM users WHERE
email = ?");
        $stmt->bind_param("s", $email);
        $stmt->execute();
        $stmt->store_result();
        if ($stmt->num_rows === 1) {
            $stmt->bind_result($user_id, $hash, $role);
            $stmt->fetch();
            if (password_verify($password, $hash)) {
                $_SESSION['user_id'] = $user_id;
                $_SESSION['role'] = $role;
                if ($role === 'admin') {
                    header('Location: admin_dashboard.php');
                } else {
                    header('Location: customer_dashboard.php');
                }
                exit;
            } else {
                $error = 'كلمة المرور غير صحيحة';
            }
        } else {
            $error = ' المستخدم غير موجود';
        }
        $stmt->close();
    }
    if ($error) {
        echo "<div class='alert alert-danger text-center'>$error</div>";
    }
}
```

```

        echo "<a href='index.php' class='btn btn-secondary mt-2'>العودة إلى تسجيل الدخول</a>";
    }
}
$mysqli->close();
?>

```

يُستخدم هذا الملف لمعالجة بيانات تسجيل الدخول التي يُدخلها المستخدم (مدير أو عميل) في نموذج صفحة index.php. يتم التحقق من البريد وكلمة المرور، وإنشاء جلسة Session وتوجيهه المستخدم بناءً على دوره.

5. ملف logout.php – تسجيل الخروج من النظام

```

<?php
session_start();
session_destroy();
header('Location: index.php');
exit;
?>

```

هذا الملف يُستخدم لإنهاء الجلسة النشطة لأي مستخدم (مدير أو عميل) عند الضغط على زر "تسجيل الخروج"، ثم يعيد توجيهه إلى صفحة الدخول index.php.

admin _ car _ location.php.7

```

<?php
session_start();
if (!isset($_SESSION['user_id']) || $_SESSION['role'] !== 'admin') {
    header('Location: index.php');
    exit;
}

// $mysqli = new mysqli("localhost", "root", "", "car_system");
$mysqli = new mysqli("localhost", "u222784783_ahmed_car", "Qweff112233+", "u222784783_car_system");

// التحقق من وجود car_id
if (!isset($_GET['car_id']) || empty($_GET['car_id'])) {
    header('Location: admin_dashboard.php');
    exit;
}

$car_id = intval($_GET['car_id']);

```

```

// جلب معلومات السيارة
$car_stmt = $mysqli->prepare("SELECT cars.*, users.name as user_name FROM
cars JOIN users ON cars.user_id = users.id WHERE cars.id = ?");
$car_stmt->bind_param("i", $car_id);
$car_stmt->execute();
$car_result = $car_stmt->get_result();
$car = $car_result->fetch_assoc();

if (!$car) {
    header('Location: admin_dashboard.php');
    exit;
}
$car_stmt->close();
?>
<!DOCTYPE html>
<html lang="ar" dir="rtl">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>سجل موقع السيارة - لوحة تحكم المدير</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap rtl.mi
n.css" rel="stylesheet">
    <script
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyCrRFgnU5kENw45gvHaTJ
QL5yXGUw1zoc"></script>
</head>
<body>
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
    <div class="container-fluid">
        <a class="navbar-brand" href="admin_dashboard.php">لوحة تحكم المدير</a>
        <div class="d-flex">
            <a href="logout.php" class="btn btn-outline-light">تسجيل الخروج</a>
        </div>
    </div>
</nav>
<div class="container mt-5">
    <div class="d-flex justify-content-between align-items-center mb-4">
        <h1>سجل موقع السيارة</h1>
        <a href="admin_dashboard.php" class="btn btn-secondary">العودة للوحة
التحكم</a>
    </div>

    <div class="card mb-4">
        <div class="card-body">
            <h5 class="card-title">معلومات السيارة</h5>
            <div class="row">
                <div class="col-md-4">

```

```

        <strong>اسم السيارة</strong> <?=
htmlspecialchars($car['car_name']) ?>
    </div>
    <div class="col-md-4">
        <strong>معرف السيارة</strong> <?=
htmlspecialchars($car['car_identifier']) ?>
    </div>
    <div class="col-md-4">
        <strong>المالك</strong> <?=
htmlspecialchars($car['user_name']) ?>
    </div>
</div>

<form class="row g-3 mb-4" method="GET">
    <input type="hidden" name="car_id" value="<?= $car_id ?>">
    <div class="col-md-6">
        <label for="date" class="form-label">تاريخ (اختياري)</label>
        <input type="date" class="form-control" id="date" name="date"
value="<?php echo htmlspecialchars($_GET['date'] ?? ''); ?>">
    </div>
    <div class="col-md-6">
        <label for="limit" class="form-label">عدد النقاط</label>
        <select class="form-select" id="limit" name="limit">
            <option value="50" <?= ($_GET['limit'] ?? '50') == '50' ? 'selected' : '' ?>>50</option>
            <option value="100" <?= ($_GET['limit'] ?? '50') == '100' ? 'selected' : '' ?>>100</option>
            <option value="200" <?= ($_GET['limit'] ?? '50') == '200' ? 'selected' : '' ?>>200</option>
            <option value="500" <?= ($_GET['limit'] ?? '50') == '500' ? 'selected' : '' ?>>500</option>
        </select>
    </div>
    <div class="col-12">
        <button type="submit" class="btn btn-primary">بحث</button>
        <button type="button" class="btn btn-success"
onclick="exportCSV()">تصدير البيانات</button>
    </div>
</form>

<div id="map" style="height: 400px; width: 100%; margin-bottom:
20px;"></div>

<h3>سجل المواقع</h3>
<table class="table table-bordered table-striped" id="historyTable">
    <thead>
        <tr>

```

```

<th>التاريخ والوقت</th>
<th>خط العرض</th>
<th>خط الطول</th>
<th>رابط الخريطة</th>
</tr>
</thead>
<tbody>
<?php
$locations = [];
$date_filter = isset($_GET['date']) && $_GET['date'] != '' ?
$_GET['date'] : null;
$limit = intval($_GET['limit']) ?? 50;

$query = "SELECT latitude, longitude, timestamp FROM locations WHERE
car_id = ?";
$params = [$car_id];
$types = "i";

if ($date_filter) {
    $query .= " AND DATE(timestamp) = ?";
    $params[] = $date_filter;
    $types .= "s";
}

$query .= " ORDER BY timestamp DESC LIMIT ?";
$params[] = $limit;
$types .= "i";

$stmt = $mysqli->prepare($query);
$stmt->bind_param($types, ...$params);
$stmt->execute();
$result = $stmt->get_result();

while ($row = $result->fetch_assoc()) {
    $locations[] = $row;
    $google_maps_url = "https://maps.google.com/maps?q=" .
$row['latitude'] . "," . $row['longitude'];
    echo "<tr>";
    echo "<td>" . htmlspecialchars($row['timestamp']) . "</td>";
    echo "<td>" . htmlspecialchars($row['latitude']) . "</td>";
    echo "<td>" . htmlspecialchars($row['longitude']) . "</td>";
    echo "<td><a href='".$google_maps_url' target='_blank' class='btn
btn-sm btn-outline-primary'>عرض على الخريطة</a></td>";
    echo "</tr>";
}
$stmt->close();

// عكس المصفوفة لعرض المسار بالترتيب الصحيح على الخريطة
$locations_for_map = array_reverse($locations);

```

```

$latest = count($locations) > 0 ? $locations[0] : null;
?>
</tbody>
</table>

<?php if (empty($locations)): ?>
<div class="alert alert-info">
    لا توجد بيانات موقع لهذه السيارة في التاريخ المحدد.
</div>
<?php endif; ?>
</div>

<script>
function initMap() {
    var map = new google.maps.Map(document.getElementById('map'), {
        center: {lat: 15.3694, lng: 44.1910},
        zoom: 7
    });

    var route = [];
    <?php if (!empty($locations_for_map)) {
        foreach ($locations_for_map as $loc) {
            echo "route.push({lat: " . floatval($loc['latitude']) . ", lng:
" . floatval($loc['longitude']) . "});\n";
        }
    }
    if (isset($latest) && $latest) { ?>
        var marker = new google.maps.Marker({
            position: {lat: <?php echo $latest['latitude']; ?>, lng: <?php
echo $latest['longitude']; ?>},
            map: map,
            title: 'آخر موقع للسيارة',
            icon: {
                url: 'https://maps.google.com/mapfiles/ms/icons/red-dot.png'
            }
        );
        map.setCenter(marker.getPosition());
        map.setZoom(15);
    <?php }
    ?>

    if (route.length > 1) {
        var polyline = new google.maps.Polyline({
            path: route,
            geodesic: true,
            strokeColor: '#FF0000',
            strokeOpacity: 1.0,
            strokeWeight: 3
        });
    }
}

```

```

polyline.setMap(map);

// إضافة مؤشرات للنقط المهمة
if (route.length > 0) {
    var startMarker = new google.maps.Marker({
        position: route[0],
        map: map,
        title: 'نقطة البداية',
        icon: {
            url: 'https://maps.google.com/mapfiles/ms/icons/green-
dot.png'
        }
    });
}

google.maps.event.addDomListener(window, 'load', initMap);

function exportCSV() {
    var table = document.getElementById('historyTable');
    var rows = Array.from(table.querySelectorAll('tr'));
    var csv = [];

    for (var i = 0; i < rows.length; i++) {
        var cols = rows[i].querySelectorAll('th,td');
        var row = [];
        for (var j = 0; j < cols.length - 1; j++) { // استبعاد عمود رابط الخريطة
            row.push('"' + cols[j].innerText.replace(/\"/g, '\"') + '"');
        }
        csv.push(row.join(','));
    }

    var csvFile = new Blob([csv.join('\n')], {type: 'text/csv'});
    var downloadLink = document.createElement('a');
    downloadLink.download = 'car_locations_<?= $car['car_identifier'] ?>_<?=
date('Y-m-d') ?>.csv';
    downloadLink.href = window.URL.createObjectURL(csvFile);
    downloadLink.style.display = 'none';
    document.body.appendChild(downloadLink);
    downloadLink.click();
    document.body.removeChild(downloadLink);
}
</script>
</body>
</html>

```

هذا الملف يقوم بإظهار موقع السيارة عند الدخول على لوحة تحكم المدير (admin) وإختيار سيارة عند النقر على عرض الموقع يقوم بفتح صفحة العميل التي تحتوي على الخريطة وجدول بالموقع اللي انحفظت لهذه السيارة .

9-هيكل قاعدة البيانات :car_system.sql

يحتوي هذا الملحق على كود SQL المستخدم لإنشاء قاعدة بيانات المشروع، التي تشمل جداول المستخدمين، السيارات، والموقع الجغرافية التي يتم إرسالها من وحدة ESP32. تم تصميم العلاقات بين الجداول بطريقة تحقق التكامل والربط بين المركبة والمستخدم وموقعها الجغرافي.

```
-- phpMyAdmin SQL Dump
-- version 5.2.1
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1:3306
-- Generation Time: Jul 05, 2025 at 05:22 PM
-- Server version: 10.11.10-MariaDB
-- PHP Version: 7.2.34

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Database: `u222784783_car_system`
--


-----
-- Table structure for table `cars`


CREATE TABLE `cars` (
  `id` int(11) NOT NULL,
  `user_id` int(11) NOT NULL,
  `car_name` varchar(100) NOT NULL,
  `car_identifier` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```

-- 
-- Dumping data for table `cars` 

-- 

INSERT INTO `cars` (`id`, `user_id`, `car_name`, `car_identifier`) VALUES
(1, 2, 'مرسيدس', '1');

-- 
-- Table structure for table `locations` 

-- 

CREATE TABLE `locations` (
  `id` int(11) NOT NULL,
  `car_id` int(11) NOT NULL,
  `latitude` double NOT NULL,
  `longitude` double NOT NULL,
  `timestamp` datetime NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

-- 
-- Dumping data for table `locations` 

-- 

INSERT INTO `locations` (`id`, `car_id`, `latitude`, `longitude`,
`timestamp`) VALUES
(228, 1, 15.44887, 45.306451, '2025-06-30 19:45:15'),
(229, 1, 15.448868, 45.306446, '2025-06-30 19:51:28'),
(230, 1, 15.448891, 45.306437, '2025-07-01 17:52:30'),
(231, 1, 15.448885, 45.306437, '2025-07-01 18:00:00'),
(232, 1, 15.448889, 45.306445, '2025-07-01 18:07:37');

-- 
-- Table structure for table `users` 

-- 

CREATE TABLE `users` (
  `id` int(11) NOT NULL,
  `name` varchar(100) NOT NULL,
  `email` varchar(100) NOT NULL,
  `password` varchar(255) NOT NULL,
  `role` enum('admin','customer') DEFAULT 'customer'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

-- 
-- Dumping data for table `users` 

```

```

-- 

INSERT INTO `users` (`id`, `name`, `email`, `password`, `role`) VALUES
(2, 'أحمد خالد', 'ahmed@gmail.com',
'$2y$10$dppbVCagEaDg7oFwrKG.uelviXD3nhXuq9e2cmPpu0i4ODOnYefNu', 'customer'),
(6, 'عبد الحكيم', 'Hakeem@gmail.com',
'$2y$10$LK03nZwxe78kmyeXB2Gb8.3swisbV5IaBsBmSQH3FyXigSuQ7SagW', 'admin'),
(13, 'صالح', 'Saleh@gmail.com',
'$2y$10$ucvl5zBxqtLvUt5FMhk4tOajrvjG64CRoFAXGwdS9k.cnEWo12fkC', 'customer');

-- 
-- Indexes for dumped tables
-- 

-- 
-- Indexes for table `cars`
-- 

ALTER TABLE `cars`
ADD PRIMARY KEY (`id`),
ADD UNIQUE KEY `car_identifier` (`car_identifier`),
ADD KEY `user_id` (`user_id`);

-- 
-- Indexes for table `locations`
-- 

ALTER TABLE `locations`
ADD PRIMARY KEY (`id`),
ADD KEY `car_id` (`car_id`),
ADD KEY `timestamp` (`timestamp`);

-- 
-- Indexes for table `users`
-- 

ALTER TABLE `users`
ADD PRIMARY KEY (`id`),
ADD UNIQUE KEY `email` (`email`);

-- 
-- AUTO_INCREMENT for dumped tables
-- 

-- 
-- AUTO_INCREMENT for table `cars`
-- 

ALTER TABLE `cars`
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=7;

-- 
-- AUTO_INCREMENT for table `locations`
-- 
```

```

-- ALTER TABLE `locations`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=233;

-- -- AUTO_INCREMENT for table `users`

-- ALTER TABLE `users`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=14;

-- -- Constraints for dumped tables
-- 

-- -- Constraints for table `cars`
-- 

ALTER TABLE `cars`
  ADD CONSTRAINT `cars_ibfk_1` FOREIGN KEY (`user_id`) REFERENCES `users`(`id`);

-- -- Constraints for table `locations`
-- 

ALTER TABLE `locations`
  ADD CONSTRAINT `locations_ibfk_1` FOREIGN KEY (`car_id`) REFERENCES `cars`(`id`);
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

