Ministry of Higher Education
and Scientific Research
Emirates International University
Faculty of Engineering and IT
Department of Information Security

وزارة التعليم العالي
والبحث العلمي
الجامعة الاماراتية الدولية
كليه الهندسة وتكنولوجيا المعلومات
قسم امن المعلومات

منصة برامج اكتشاف الثغرات (مأمون).

عمل الطلاب:

صالح قاسم عبدالله دبوان

محمد خالد احمد الجبلي.

دعاء فاروق احمد الحريبي

ياسين محمد إبراهيم دوم

إشراف:

د جميل راشد

رئيس قسم امن المعلومات الجامعة الإماراتية

تم انجاز هذا المشروع لنيل درجة البكالوريوس لتخصص امن المعلومات

2024/2023م

قال تعالى:-

﴿ٱلَّذِينَ ءَامَنُواْ وَلَمۡ يَلۡبِسُوٓاْ إِيمَٰنَهُم بِظُلۡمٍ أُوْلَٰٓئِكَ لَهُمُ ٱلۡأَمۡنُ وَهُم مُّهۡتَدُونَ﴾

«آل عمران 97»

الخلاصة

في ظل ما نعيشه اليوم من تسارع في التقدم الرقمي والاتصال الدائم بالشبكات الاتصالات أصبح من الضروري إيجاد حل امني رقمي ومن هنا أتت فكرة مأمون وهو عبارة عن منصة تجمع ما بين الشركات و المخترقين الأخلاقيين المحترفين بغرض الحصول على بيئة آمنه حيث تسمح للشركات والمؤسسات بأنشاء البرامج الخاصة بها و إدارتها كما يمكن للمخترقين الأخلاقيين من التقديم لهذه البرامج والحصول على المكافئات المقدمة من الشركات أو المؤسسات حسب نوع البرنامج المقدم والحد من عمليات النصب والاحتيال التي قد تتعرض لها الشركات والمؤسسات.

اهداء

نهدي هذا العمل المتواضع الى وأبأنا وامهاتنا الذين اعتنوا بنا وربونا صغار وتعبوا حتى نصل الى ما وصلنا اليه اليوم من النجاح وكذلك جميع المعلمين والمعلمات الذين كان لهم دور في تأسيس العلم في نفوسنا وجعل العلم محبب الينا وكذلك الى أخوتنا وخواتنا فجزاكم الله عنا خيراً.

شكر وعرفان

قبل كل شيء نود ان نشكر ربنا على ما امنه علينا من نعم وعطايا لا تعد ولا تحصى ومنها نعمه العلم.

كما نود أن نعرب عن شكرنا وتقديرنا للدكتور/ جميل راشد على هذا الامتياز والتوجيه والتشجيع اللطيف والمشورة العلمية والأشراف المفيد والتعاون الصادق الجميل المساد اليه والتمنيات الطيبة التي جعلت من هذا العمل ممكناً.

وكما نشكر كافة أعضاء هيئة التدريس على جهودهم المبذولة في إيصال المعلومة وتبسيطها.

شهادة المشرف

أشهد أن إعداد هذا المشروع بعنوان مأمون أُعدت بواسطة الطلاب والتالية أسماءهم: -

1. محمد خالد أحمد الجبلي.
2. دعاء فاروق احمد الحريبي.
3. ياسين إبراهيم محمد دوم.
4. صالح قاسم دبوان.

تم تحت إشرافي في كلية الهندسة وتكنولوجيا المعلومات قسم أمن المعلومات في الوفاء الجزئي لمتطلبات الحصول على درجة البكالوريوس في الجامعة الإماراتية الدولية.

د جميل راشد

فهرس المحتويات

الخلاصة ---------- II

اهداء ---------- III

شكر وعرفان ---------- IV

شهادة المشرف ---------- V

1.1. **المقدمة** ---------- 02

1.2. **المشاكل** ---------- 03

1.3. **الأهداف** ---------- 03

1.4. **تعريف النظام** ---------- 03

1.5. **صفات المستخدمين** ---------- 03

1.6. **المعوقات** ---------- 04

1.7. **الاقتراحات والفرضيات** ---------- 04

1.8. **المدى** ---------- 04

1.9. **المنهجية المتبعة** ---------- 04

1.10. **اعمال مشابهة** ---------- 05

1.11. **الخطة الزمنية** ---------- 05

1.12. **دراسة الجدوى** ---------- 05

1.12.1. **الجدوى التقنية** ---------- 05

1.12.1.1. ***Hardware*** ---------- 05

1.12.1.2. ***Software*** ---------- 05

1.12.2. **الجدوى الاقتصادية** ---------- 06

1.12.2.1. ***تكاليف التنفيذ*** ---------- 06

1.12.2.2. ***تكاليف التشغيل*** ---------- 06

.1.12.3 **الجدوى الزمنية** ---------- 07

2.1. **الخلفية النظرية** ---------- 09

2.1.1. **مقدمة** ---------- 09

2.1.2. **تاريخ الاختراق** ---------- 09

2.1.3. **مفهوم الـ "Bug Bounty"** ---------- 10

2.1.4. **دور الـBug Bounty في تعزيز الأمان الرقمي** ---------- 10

2.2. **النظام الحالي** ---------- 11

2.3. **الأنظمة المشابهة** ---------- 12

3.1. **الافتتاحية** ---------- 14

3.2. **جمع المعلومات** ---------- 14

3.3. **متطلبات المستخدمين** ---------- 14

3.3.1. **متطلبات فريق امن المعلومات في الشركات والمؤسسات** ---------- 14

3.3.2. متطلبات المخترقين الاخلاقيين ---------- 14
3.3.3. متطلبات المدقق الأمني ---------- 14
3.4. المتطلبات الوظيفية ---------- 14
3.5. المتطلبات غير الوظيفية ---------- 15
3.5.1. قابلية الاستخدام ---------- 15
3.5.2. الاعتمادية ---------- 15
3.5.3. الأداء ---------- 15
3.6. سيناريو المشروع ---------- 16
3.7. نماذج النظام ---------- 16
4.1 مقدمة ---------- 19
4.2 واجهات النظام ---------- 19
4.2.1 الصفحة الرئيسية ---------- 19
4.2.2 واجهة الفرص ---------- 20
4.2.3 واجهة المخترقين الأخلاقيين ---------- 20
4.2.4 واجهات الإحصائيات ---------- 21
4.2.4.1 واجهة احصائيات الشركة ---------- 21
4.2.4.2 واجهة احصائيات المخترق الأخلاقي ---------- 22
4.2.5 واجهات التقارير ---------- 23
4.2.5.1 واجهة التقارير للشركة ---------- 23
4.2.5.2 واجهة التقارير للمخترق الأخلاقي ---------- 23
4.2.6 واجهات خيارات الدفع ---------- 24
4.2.6.1 واجهة خيارات الدفع للشركة ---------- 24
4.2.6.2 واجهة خيارات الدفع للمخترق الأخلاقي ---------- 25
4.2.7 واجهة إدارة البرامج ---------- 25
4.2.8 واجهة انشاء برنامج ---------- 26
4.2.9 واجهة إدارة الأصول ---------- 26
4.2.10 واجهة إضافة أصل ---------- 27
4.2.11 واجهة تسجيل الدخول ---------- 27
4.2.12 واجهة انشاء حساب ---------- 28
4.2.12.1 انشاء حساب كمخترق ---------- 28
4.2.12.2 انشاء حساب كشركة ---------- 29
4.3 الخاتمة ---------- 31
4.4 جوانب القصور ---------- 31
4.5 الأعمال المستقبلية ---------- 31
4.6 المراجع ---------- 33
4.7 الملاحق ---------- 34

فهرس الجداول

جدول 1-1:- صفات المستخدمين 3
جدول 1-2:- الخطة الزمنية 5
جدول 1-3:- تكاليف التنفيذ المادية 6
جدول 1-4:- تكاليف التشغيل المادية 6
جدول 1-5:- الجدوى الزمنية 7

فهرس الرسوم التوضيحية

رسم توضيحي 1-1:- للمنهجية المتبعة 04
رسم توضيحي 3.1:- للكائنات وعلاقاتها داخل قاعدة البيانات 16
رسم توضيحي 3-3:- لعملية الدفع 17
رسم توضيحي3-2:- لعملية تسجيل الدخول داخل النظام 17

فهرس الأشكال

شكل1-1:- رسم توضيحي للمنهجية المتبعة..... 04
شكل 4-1:- الواجهة الرئيسية للنظام 19
شكل 4-2:- واجهة الفرص..... 20
شكل 4-3:- واجهة المخترقين 20
شكل 4-4:- واجهة الأحصائيات للشركة..... 21
شكل 4-5 واجهة الأحصائيات للمخترق الأخلاقي 22
شكل4-6:- واجهة التقارير الخاص بالشركة 23
شكل 4-7 :- واجهة التقارير للمخترق الأخلاقي..... 23
شكل 4-8 :- واجهة خيارات الدفع الخاص بالشركة..... 24
شكل 4-9 :- واجهة خيارات الدفع للمخترق الأخلاقي 25
شكل 4-10 :- واجهة ادارة البرامج..... 25
شكل 4-11:- واجهة انشاء برنامج..... 26
شكل 4-12 :- واجهة ادارة الأصول 26
شكل 4-13:- واجهة اضافة الأصول..... 27
شكل 4-14:- واجهة تسجيل الدخول 27
شكل 4-15:- واجهة انشاء حساب كمخترق 28
شكل 4-16:- واجهة انشاء حساب كشركة..... 29

الفصل الأول:
المقدمة

1.1. المقدمة:

دخلت التكنولوجيا معظم تفاصيل حياتنا، إلى درجة تفوق كل التصورات، وتطورت بتسارع كبير، حيث اصبحت من اهم الوسائل التي تستخدمها الشركات والمؤسسات. ولقد استطاعت التّكنولوجيا بفضل انتشارها أن تُغيّر في أنماط الحياة اليومية للشعوب، سواءً في الجانب الثقافي او الاقتصادي او الاجتماعي.

وتطوّرت البُنى التّحتية لكثيرٍ من المُجتمعات بفعلِ استخدام التّكنولوجيا الحديثة بشكلٍ كبيرٍ في الآونة الأخيرة، وأصبحَ الجانب التّكنولوجيّ من الجوانب الهامّة والأساسية فيها، إذ أحدث تبدُّلاً وتحوُّلاً في مسيرة المُجتمعات على كافة الأصعدة الثقافية والسياسية والاجتماعية ولكن مع كل تطور في عالم التكنولوجيا يوجد الكثير من التهديدات الإلكترونية لهذه التكنولوجيا حيث يشكل الهجوم الالكتروني او الهجوم السيبراني خطرا كبيرا على الشركات او المؤسسات وكافة القطاعات المختلفة حيث يمكن للمهاجم الحصول على البيانات الحساسة لهذه الشركات او المؤسسات او القطاعات في حال نجحت عملية الاختراق مما قد يؤثر سلبا عليها وربما يؤدي الى تدني سمعتها في حال ظهرت هذه المعلومات للعلن.

كما ان الشركات والمؤسسات في حال ارادت ان توظف مخترقا خارجيا لا يمكنها انت تضمن الحصول على تقارير سليمة او صحيحة او في اسواء الحالات عدم الحصول على التقرير وهذا يرجع الى كثرة المحتالين في العالم الرقمي مما يؤدي الى فقدان ثقة الشركات والمؤسسات في أي مخترق خارجي او في حال تواصل معها أحد المخترقين الاخلاقيين حال وجود ثغرة امنية في أنظمة الشركة او المؤسسة.

أيضا لا يمكن للمخترق الأخلاقي ان يضمن الحصول على المقابل المادي في حال قام بتسليم التقرير الى الشركة او المؤسسة مما يؤدي الى بقاء الثغرات منتشرة وموجودة في أنظمة الشركات والمؤسسات في العالم الرقمي.

ولذلك تظل معظم الشركات والمؤسسات معرضة لخطر الثغرات الأمنية ويصعب المخترقون الأخلاقيون من تأمين الشركات والمؤسسات ضد هذه الثغرات.

ومن هنا جاءت فكرة هذا المشروع.

1.2. المشاكل:

1) عدم ضمان الحصول على المقابل المادي من الشركات.
2) عدم ضمان الحصول على تقارير فعليه من المخترقين الاخلاقيين.

1.3. الأهداف:

1) طلب ضمان تجاري للشركة او المؤسسة.
2) عرض التقارير للمراجعة والتدقيق للتأكد من سلامتها.

1.4. تعريف النظام:

عبارة عن منصة ويب تعمل على الربط ما مبين الشركات والمخترقين الاخلاقيين.

1.5. صفات المستخدمين:

جدول 1-1:- صفات المستخدمين

المستخدم	المؤهل	الوظيفة
مدير النظام	متخصص امن معلومات	مراقبه النظام وسير العمليات داخل النظام.
مخترق اخلاقي	محترف في اختبار الاختراق	التسجيل في برامج الشركات واختبار أنظمتها وكتابة التقارير وتسليمها للمنصة.
قسم امن المعلومات في الشركة	متخصص امن المعلومات	عرض برامج المكافآت للمخترقين الاخلاقيين ومراجعة التقارير الواردة منهم.
مدقق أمني	متخصص امن المعلومات	التدقيق في التقارير الواردة من المخترقين الاخلاقيين في حال طلبت الشركة لذلك.

1.6 المعوقات:

1) صعوبة اقناع الشركات والمؤسسات بالانضمام الى المنصة.

1.7 الاقتراحات والفرضيات:

1) جعل الانضمام الى المنصة مجانياً مع ميزات محدودة. وعمل حملات إعلانية وترويجية لعرض مميزات الانضمام للمنصة.

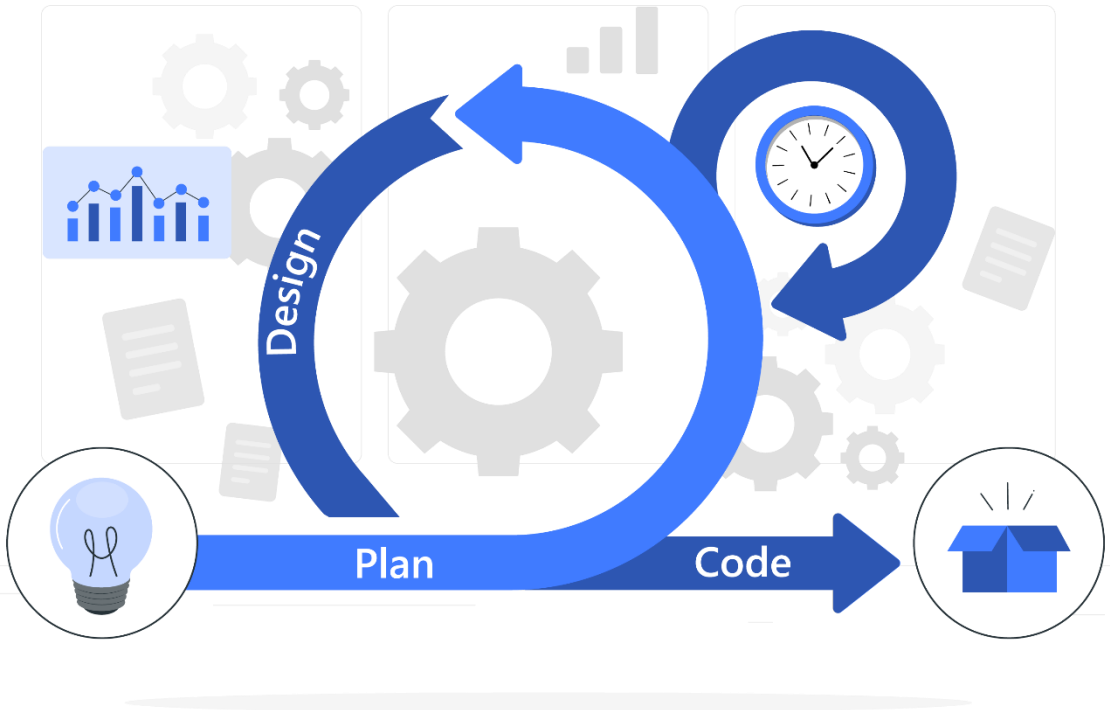
1.8 المدى:

الشركات والمؤسسات الراغبة في تعزيز امنها من خلال مخترقين خارجيين مقابل مكافآت مادية والمخترقين الاخلاقيين الراغبين في المساهمة في تعزيز امان الشركات والمنافسة على المكافآت التي تقدمها الشركات.

1.9 المنهجية المتبعة:

المنهجية التي تم اختيارها في الوقت الحالي هي (Agile) بإطار العمل (Programming Extreme) ولذلك للأسباب التالية:

1) جودة المخرجات من هذه المنهجية.
2) فهم الفريق للمنهجية.
3) وجود المقومات لدى الفريق للعمل بهذه المنهجية.



رسم توضيحي 1-1:- للمنهجية المتبعة

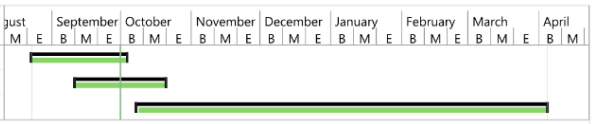
1.10. اعمال مشابهة:

1) منصة HackerOne.
2) منصة Bugcrowd.
3) منصة Synack.
4) منصة Try to hack me.

1.11. الخطة الزمنية:

جدول 2-1:- الخطة الزمنية

ID	Task Mode	Task Name	Duration	Start	Finish
1		التحليل	30 days	Wed 8/23/23	Tue 10/3/23
5		التصميم	30 days	Mon 9/11/23	Fri 10/20/23
8		التنفيذ	130 days	Sun 10/8/23	Thu 4/4/24



1.12. دراسة الجدوى:

1.12.1. الجدوى التقنية:

1.12.1.1. Hardware:

- ستة أجهزة حاسوب لا تقل على ان تكون بمعالج Intel® Core™ i7-8850H و RAM 8GB.
- طابعة.
- Vercel Hosting.
- Domain Nam.

1.12.1.2. Software:

- بيئة تطوير Visual Studio Code.
- Draw.io.
- Figma
- MS Word.
- MS PowerPoint.
- MS Project.
- FastAPI.
- Nextjs.
- Supabase.

1.12.2. الجدوى الاقتصادية:

1.12.2.1. **تكاليف التنفيذ**:

1) Software:

التكاليف البرمجية الخاصة بتنفيذ النظام تم استخدام أدوات وبرامج وتقنيات مجانية لبناء المشروع.

2) Hardware:

جدول 3-1:- تكاليف التنفيذ المادية

المورد	العدد	التكلفة	الاستخدام
PC	6	$700	تطوير المنصة
Printer with Scanner	1	$400	تصوير الوثائق وطباعتها
الإجمالي		$4600	

1.12.2.2. **تكاليف التشغيل**:

1) Hardware:

جدول 4-1:- تكاليف التشغيل المادية

المورد	العدد	التكلفة	الاستخدام	وقت استخدام
Vercel Hosting	1	$0	استضافة المنصة	يومياً
Domain Name	1	$22	عرض الموقع	يومياً
الإجمالي		$22		

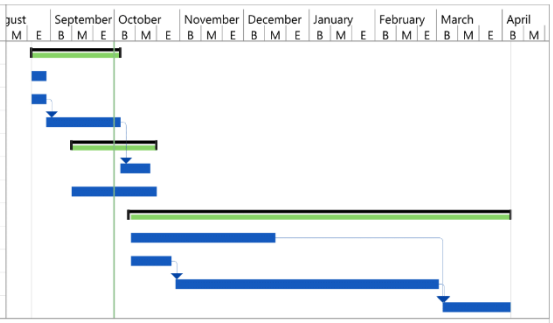
2) Software:

التكاليف البرمجية الخاصة بتشغيل النظام تم استخدام أدوات وبرامج وتقنيات مجانية لبناء المشروع.

1.12.3. الجدوى الزمنية:

جدول 5-1:- الجدوى الزمنية

ID	Task Mod	Task Name	Duration	Start	Finish
1		**التحليل**	**30 days**	**Wed 8/23/23**	**Tue 10/3/23**
2		تحليل بنية الموقع	5 days	Wed 8/23/23	Tue 8/29/23
3		تحليل المتطلبات	5 days	Wed 8/23/23	Tue 8/29/23
4		تحليل قاعدة البيانات	25 days	Wed 8/30/23	Tue 10/3/23
5		**التصميم**	**30 days**	**Mon 9/11/23**	**Fri 10/20/23**
6		تصميم قاعدة البيانات	10 days	Wed 10/4/23	Tue 10/17/23
7		تصميم الواجهات	30 days	Mon 9/11/23	Fri 10/20/23
8		**التنفيذ**	**130 days**	**Sun 10/8/23**	**Thu 4/4/24**
9		تنفيذ الواجهات	50 days	Mon 10/9/23	Fri 12/15/23
10		تنفيذ قاعدة البيانات	15 days	Mon 10/9/23	Fri 10/27/23
11		تنفيذ برمجة عمليات الموقع	90 days	Mon 10/30/23	Fri 3/1/24
12		ربط بين الواجهات والعمليات	24 days	Mon 3/4/24	Thu 4/4/24



2. الفصل الثاني:

الخلفية النظرية والنظام الحالي

2.1. الخلفية النظرية:

2.1.1. مقدمة:

تعيش المجتمعات العصرية اليوم في زمن التكنولوجيا الرقمية حيث تعتمد أنظمة الإعلام والاتصال وغيرها من الأنظمة على شبكة الإنترنت والأنظمة الحاسوبية لتسهيل العمليات وتبادل المعلومات. ومع تزايد تلك التطورات التكنولوجية، تزداد التحديات المتعلقة بأمان المعلومات وخصوصية البيانات. يعد الاختراق أحد تلك التحديات التي تواجهها الأفراد والمؤسسات.

الاختراق غير الأخلاقي يشمل استخدام المهارات الحاسوبية للوصول إلى نظم معلوماتية أو أجهزة حاسوب دون إذن قانوني، سواء بهدف الاستكشاف أو التجربة أو التسبب في الضرر.

يتسبب الاختراق غير الأخلاقي في تلف البيانات وخسائر مالية وتأثيرات سلبية على الخصوصية والأمان السيبراني.

يتطلب التصدي للتحديات المتعلقة بالاختراق التعاون بين الجهات المعنية، بما في ذلك الحكومات والشركات ومقدمي الخدمات الأمنية. يجب تعزيز الوعي بمخاطر الاختراق غير الأخلاقي وتعزيز التدابير الأمنية لحماية الأنظمة الحاسوبية والبيانات ومن ضمن هذه التدابير هو منصات برامج مكافحة الثغرات او ال Bug Bounty.

2.1.2. تاريخ الاختراق:

يعود تاريخ الاختراق إلى أوائل اختراقات الحواسيب والشبكات الإلكترونية في السبعينيات من القرن الماضي. كانت هذه الاختراقات غالباً ما تأتي من هواة التكنولوجيا الذين كانوا يكتشفون ثغرات الأمان والتأمين في أنظمة الحواسيب والشبكات. كانت أهدافهم في العادة للتسلية أو للترفيه الشخصي، ولم تكن تلك الاختراقات تحمل الأهداف الخبيثة التي تصاحب الاختراقات الحديثة.

مع تطور التكنولوجيا وانتشار استخدام الإنترنت والشبكات الرقمية، أصبحت الاختراقات أكثر تعقيدا وخطورة. بدأت الاختراقات في تلك الفترة تطرح تحديات جديدة أمام مجتمع الأمان الرقمي، حيث برزت حاجة ملحة للابتكار وتطوير سبل الحماية والدفاع ضد الهجمات الإلكترونية.

من أوائل الاختراقات الشهيرة كانت اختراقات شبكات الاتصالات وخدمات البريد الإلكتروني في التسعينيات من القرن الماضي، حيث شابت الشبكة العنكبوتية آلاف الاختراقات التي أثرت بشكل كبير على البيئة الرقمية في ذلك الوقت. من هنا برز دور الشركات الكبرى والحكومات في تعزيز الأمان الرقمي والعمل على مكافحة الاختراقات بطرق متقدمة.

على مر السنين، زادت الاختراقات بشكل كبير جداً وأصبحت تشمل جميع القطاعات والمؤسسات، من الشركات الكبيرة إلى المؤسسات الصغيرة والفردية. حيث تبنت الهجمات الإلكترونية أهدافاً مالية واستهدافاً للبيانات الحساسة والمعلومات السرية.

في الوقت الحاضر، تعد مشكلة الاختراقات الرقمية من أكبر التحديات التي تواجه العالم الرقمي. ومع تزايد الاعتماد على التكنولوجيا والتحول الرقمي في مختلف جوانب الحياة، يصبح تعزيز الأمان الرقمي أمراً ضرورياً للحفاظ على خصوصية البيانات وسلامة البنية التحتية الرقمية.

من هنا، يظهر دور ال bug bounty بوضوح كبير في تحسين الأمان الرقمي ومكافحة الاختراقات. بواسطة استخدام ال bug bounty، يمكن للمؤسسات والشركات تشجيع القراصنة الأخلاقيين للاستفادة من مهاراتهم في اكتشاف الثغرات والاختراقات والإبلاغ عنها بشكل أخلاقي وآمن. وبذلك، تساهم ال bug bounty في تعزيز الأمان الرقمي وتقليل فرص التعرض للاختراقات بشكل فعال.

2.1.3. مفهوم الـ "Bug Bounty":

يعد الـ "Bug Bounty" أحد أساليب الإبلاغ عن الاختراقات وتحفيز الباحثين عن الثغرات الأمنية على اكتشاف الثغرات والتقارير عنها بشكل أخلاقي. يتمثل الهدف من الـ "Bug Bounty" في تشجيع الباحثين على العثور على الثغرات الأمنية والإبلاغ عنها للشركات والمؤسسات المعنية مقابل مكافأة مالية أو تقدير عام.

2.1.4. دور الـBug Bounty في تعزيز الأمان الرقمي:

يحمل دور الـBug Bounty أهمية كبيرة في تعزيز الأمان الرقمي وحماية الأنظمة والبيانات الحساسة، وذلك على عدة جوانب:

1. اكتشاف الثغرات وسرعة التصحيح:

من خلال مبادرات الـBug Bounty ، يمكن للشركات والمؤسسات اكتشاف الثغرات الأمنية بشكل سريع وفعّال. فالباحثون الأمنيون المشاركون في تلك المبادرات يقومون بتحليل الأنظمة والتطبيقات الرقمية بشكل دقيق والعثور على الثغرات، مما يسمح للفرق الفنية بتصحيح تلك الثغرات على الفور قبل أن يستغلها المهاجمون.

2. توعية المطورين وتحسين الجودة:

تشجيع المطورين ومصممي البرمجيات على تحسين أمان الأنظمة والتطبيقات، حيث يُشجعون على كتابة شيفرات أمنية وإصلاح الثغرات المكتشفة. وهذا بدوره يساهم في تحسين الجودة العامة للبرمجيات وزيادة صلابتها أمام محاولات الاختراق.

3. الحد من التكلفة الناتجة عن الاختراقات:

من خلال اكتشاف الثغرات الأمنية وتصحيحها مسبقًا، يمكن لمبادرات الـBug Bounty تقليل التكاليف الناتجة عن الاختراقات والاستجابة لها. وهذا يساهم في توفير الموارد المالية والبشرية التي قد تستنفذها الشركات في حالة حدوث اختراق غير مخطط له.

4. تشجيع ثقافة العمل الجماعي والتعاون:

تعزيز ثقافة العمل الجماعي والتعاون بين الباحثين الأمنيين والشركات، مما يساهم في زيادة الوعي الأمني وتبادل المعرفة والتقنيات بين الأطراف المهتمة.

في الختام، يُظهر دور مبادرات الـBug Bounty أهمية كبيرة في تحسين الأمان الرقمي وحماية الأنظمة الرقمية. حيث إنها تساهم في تحفيز الباحثين الأمنيين وتوجيه قدراتهم نحو إيجاد حلول أمان مبتكرة وذكية. وبما أن التهديدات السيبرانية تتطور باستمرار، يعد دعم مبادرات الـBug Bounty أمرًا حيويًا للمساهمة في بناء بيئة رقمية آمنة ومستقرة للمستخدمين والشركات على حد سواء.

2.2. النظام الحالي:

في النظام الحالي تقوم الشركات بعرض وظيفة مخترق أخلاقي ليقوم بعد ذلك المخترقون بالتقديم على الوظيفة ثم بعد اجراء المقابلات الشخصية يتم اخيار مجموعة من المخترقين ثم تعرض عليهم أصول الشركة التي تريد عمل اختبار اختراق لها ثم يقوم المخترقون بعمل اختبار اختراق لها وفي هذه الحالة يجب على الشركة إعطاء مقابل مادي للمخترقين سواء وجدوا ثغرات امنية ام لا وهذا يكلف الشركة كثيرا.

في بعض الحالات يقوم المخترقون باختراق أصول الشركة عن بعد بدون اخذ أي اذن منها وفي حالة تم اكتشاف ثغرات امنية لا يتم التبليغ عنها وانما يتم بيعها في السوق السوداء للثغرات مما يعرض الشركة للخطر وفي حالة قام المخترق بعرض الثغرة للشركة فانه لا يمكن له ان يضمن حقة المادي جراء هذه الثغرة مما يضطرهم الى اللجوء الى عمل أشياء غير أخلاقية كما ان الشركة لا يمكنها ان تضمن ان هذه الثغرة موجودة بالفعل في نظامها وذلك بسبب قلة المعلومات التي يكشفها المخترق للشركة.

2.3. الأنظمة المشابهة:

- **HackerOne**:
  - المميزات:
    - سهولة الاستخدام.
    - يمكن تقييم الثغرة بأكثر من طريقة.
    - سرعة الاستجابة لمراجعة الثغرات الحساسة.
  - العيوب:
    - لا يوفر طبقة مجانية للاستخدام.
    - خدمة عملاء غير خبيرة في مجال امن المعلومات.
- **Bugcrowd**:
  - المميزات:
    - إدارة البرامج ممتازة.
    - خدمة API ممتازة.
  - العيوب:
    - يجب عليك اختيار المخترقين لكل برنامج.
    - تعقيدات استخدام المنصة.
- **Synack**:
  - المميزات:
    - سهولة الاستخدام.
    - يمكن عرض تقارير مخصصة حول المخاطر التي تم اكتشافها واصلاحها.
  - العيوب:
    - لا يوجد معيار لتقييم المخاطر المكتشفة.
    - أسعار مرتفعة نسبياً.

3. الفصل الثالث:

تحليل النظام

3.1. الافتتاحية:

في هذا الفصل سيتم وصف عمل النظام وطريقة جمع المعلومات وطريقة التحليل المستخدمة وتوضيح متطلبات المستخدم سواء كانوا شركات ومؤسسات أو مخترقين اخلاقيين وسيتم توضيح المتطلبات الوظيفية والغير الوظيفية التي سيوفرها النظام، وسيتم توضيح الـ UML والتي تحتوي على تفصيل لعمل النظام، والتي ستساعد لاحقا في عملية التصميم والتنفيذ.

3.2. جمع المعلومات:

- البحث:

تم عمل عدة بحوثات عبر الانترنت لفهم كيفية عمل الشركات مع مواقع اختبار الاختراق الموجودة حالياً.

- الملاحظة:

من خلال الملاحظة في الواقع المُعاش لاحظنا كيف تعمل الشركات لاكتشاف الثغرات في أصولها وكيف يعمل المخترقون على استغلال الثغرات المكتشفة.

3.3. متطلبات المستخدمين:

3.3.1. متطلبات فريق امن المعلومات في الشركات والمؤسسات:

- عرض التقارير الأمنية وادارتها.
- التواصل مع المخترقين الاخلاقيين.
- انشاء برامج المكافئات.
- منح المكافئات.

3.3.2. متطلبات المخترقين الاخلاقيين:

- التواصل مع الشركات.
- عرض البرامج والانضمام اليها.
- تقديم التقارير الأمنية.
- متابعة عمليات دفع المكافئات.

3.3.3. متطلبات المدقق الأمني:

- عرض التقارير الأمنية.
- مراقبة حالات التقارير.
- سجل وصول التقارير.
- التواصل مع المخترقين الاخلاقيين والشركات.

3.4. المتطلبات الوظيفية:

❖ **خدمة التسجيل:**

- تتيح هذه الخدمة للشركات والمؤسسات والمخترقين الاخلاقيين بالتسجيل في المنصة.

❖ **خدمة إدارة البرامج:**

- تتيح هذه الخدمة للشركات والمؤسسات بإنشاء وإدارة ومراقبة البرامج داخل المنصة.

❖ **خدمة إدارة الأصول:**

- تتيح هذه الخدمة للشركات والمؤسسات بإدارة أصولها والسماح للمخترقين بالوصول لها ضمن البرامج او منعهم من ذلك.

❖ **خدمة الاشتراكات:**

- تتيح هذه الخدمة للشركات والمؤسسات الاشتراك في المميزات الإضافية التي تقدمها المنصة.

❖ **خدمة كتابة التقارير:**

- تتيح هذه الخدمة للمخترق الأخلاقي بكتابة التقرير الخاص به وتسليمه الى الشركة.

❖ **خدمة عرض التقارير:**

- تتيح هذه الخدمة للشركات والمؤسسات والمدقق الأمني باستعراض التقارير وقراءتها والاطلاع عليها.

❖ **خدمة التسجيل في البرامج:**

- تتيح هذه الخدمة للمخترقين الاخلاقيين بالتسجيل ضمن البرامج التي تقدمها الشركات والمؤسسات والمساهمة فيها برفع التقارير الأمنية.

❖ **خدمة منح واستلام المكافئات:**

- تتيح هذه الخدمة للشركات والمؤسسات بدفع المكافئات للمخترقين الاخلاقيين الذين قدموا تقارير امنية مقبولة وصحيحة.

❖ **خدمة إدارة التقارير:**

- تتيح هذه الخدمة للشركات والمؤسسات والمدققين الأمنيين بمراجعة واستعراض التقارير المقدمة وادارتها وقبولها او رفضها او طلب الإضافة عليها.

3.5. المتطلبات غير الوظيفية:

3.5.1. قابلية الاستخدام:

- يجب على النظام ان يقدم تجربة ممتازة للمستخدم خالية من أي تعقيدات وسهلة التعلم والممارسة بدون بذل أي جهد لتحليل الأشكال والرسوميات من المستخدمين.

3.5.2. الاعتمادية:

- يجب على النظام ان يسجل جميع التحركات في البيانات اللازمة لمنع المراوغة والاحتيال.

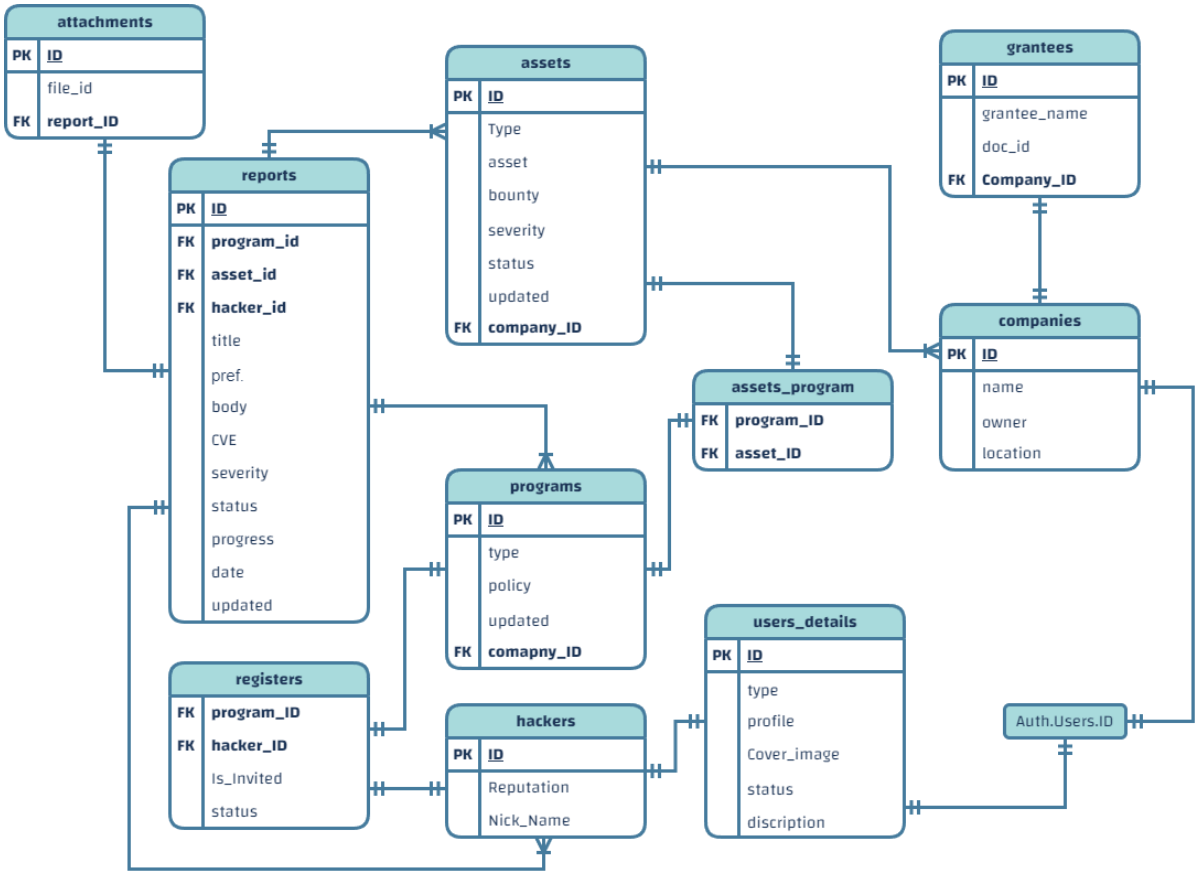
3.5.3. الأداء:

- يجب ان يعمل النظام بسرعة وكفأه عالية وان يكون متوفر على مدار الساعة للمستخدمين دون توقف مفاجئ في الخدمات.

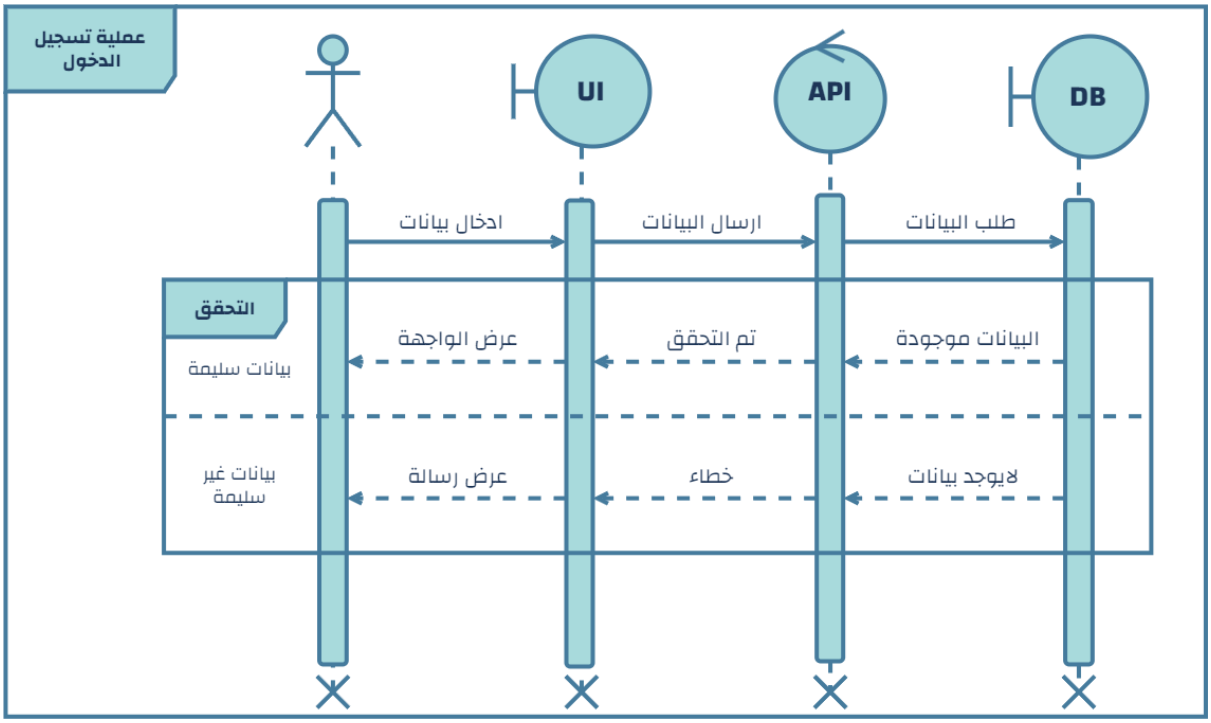
3.6. سيناريو المشروع:

المشروع عبارة عن منصة ويب تسمح للشركات والمؤسسات والقطاعات المختلفة بإنشاء حساباتها داخل المنصة ومن ثم عرضها على مدير النظام للموافقة وفي حالة الموافقة يمكن للشركة استخدام النظام بميزات محدودة او الاشتراك في الخدمات المدفوعة كما يتيح النظام للشركات من انشاء البرامج العامة والخاصة (في حالة الاشتراك) ويتيح لها إدارة البرامج ويتيح أيضا للمخترقين الاخلاقيين ان يسجلوا في المنصة بدون التصريح عن هوياتهم الحقيقية ويتيح لهم أيضا الانضمام الى البرامج العامة وتقديم التقارير عن الثغرات الموجودة في أصول البرنامج المحدد ثم يعرض التقرير على المدقق الأمني (في حالة الاشتراك) او الفريق الأمني في الشركة المالكة للبرنامج ثم التأكد من صحة التقرير وإعطاء المكافأة للمخترق صاحب التقرير وتحويلها الى حسابه في المنصة ليتمكن من الحصول عليها في احدى خدمات السحب التي تدعمها المنصة.

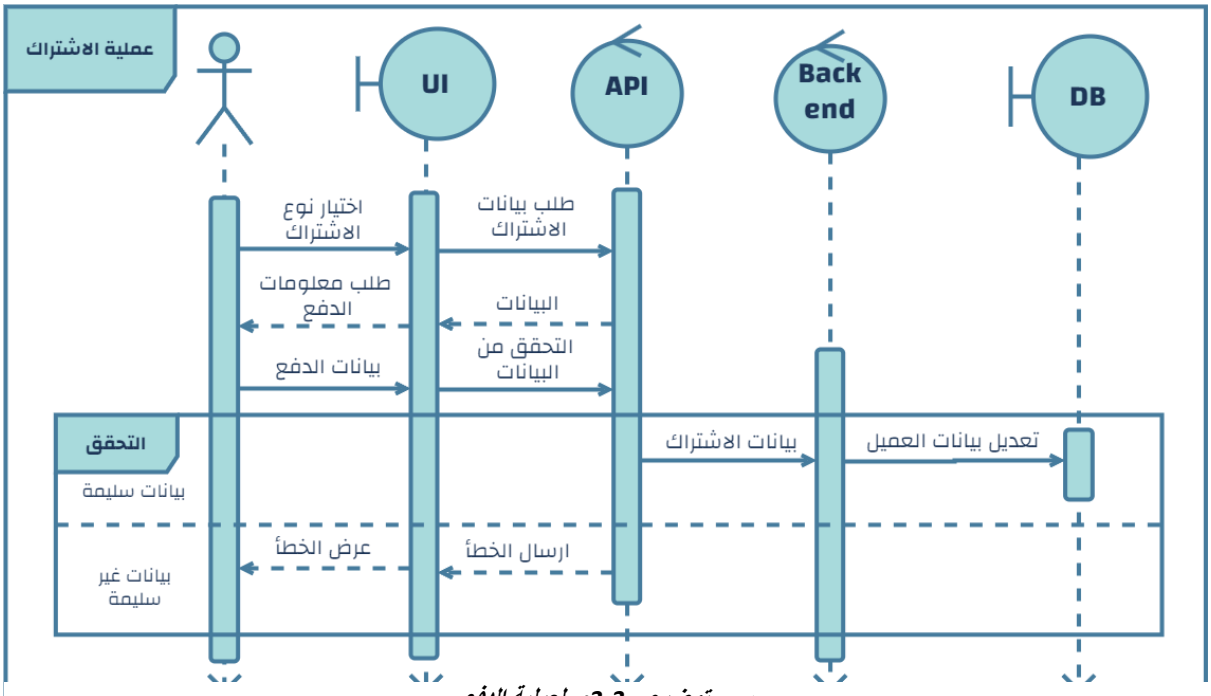
3.7. نماذج النظام:



رسم توضيحي 3.1:- للكائنات وعلاقاتها داخل قاعدة البيانات



رسم توضيحي2-3:- لعملية تسجيل الدخول داخل النظام



رسم توضيحي 3-3:- لعملية الدفع

4.الفصل الرابع

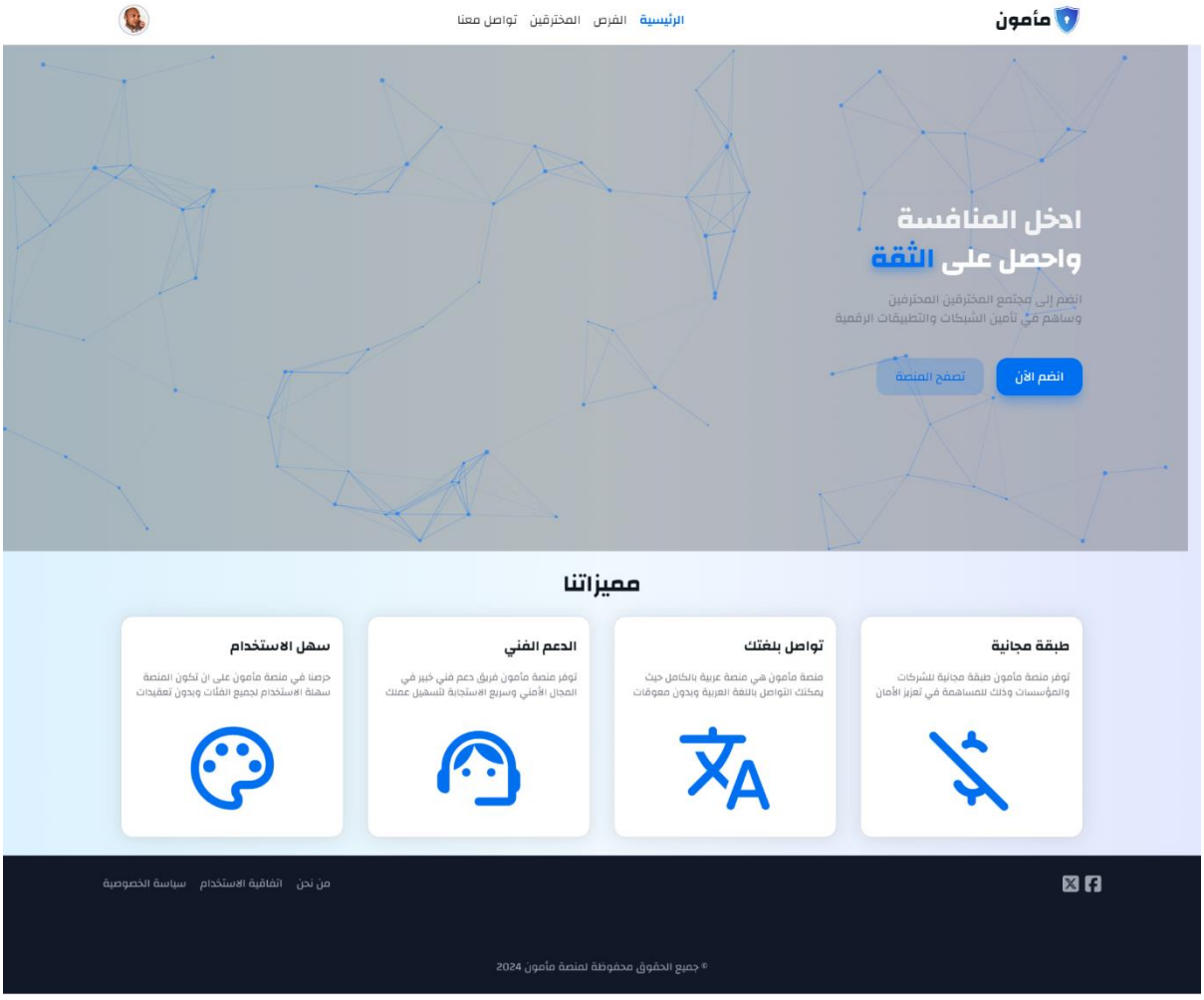
تصميم النظام

4.1 مقدمة:

تم في هذا الفصل التطرق الى مخطط قاعدة البيانات والواجهات الرئيسية للنظام.

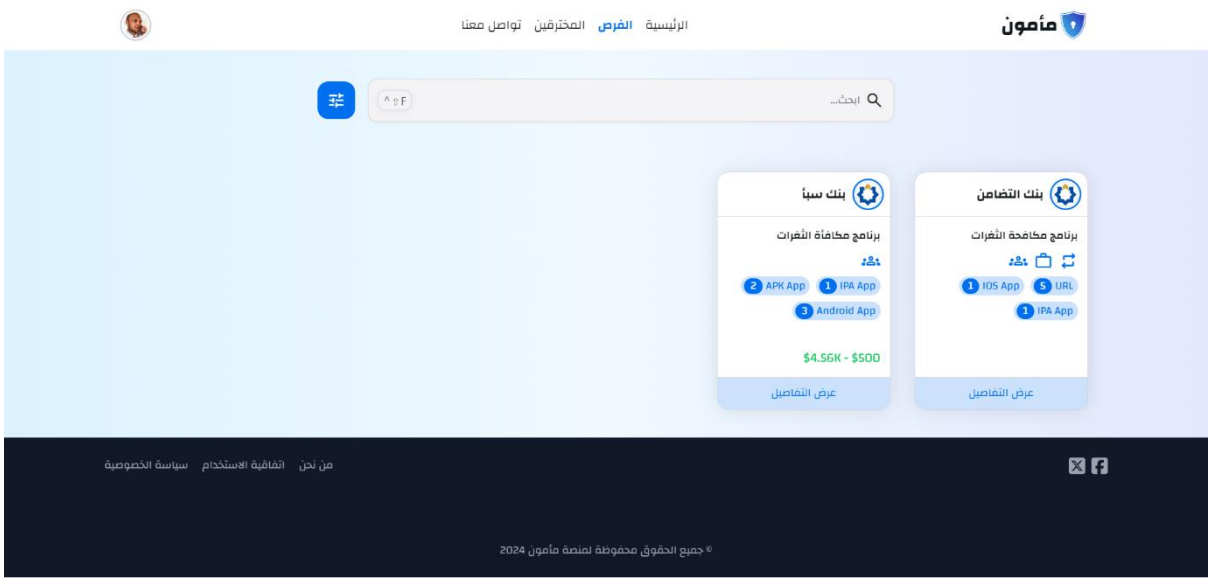
4.2 واجهات النظام:

4.2.1 الصفحة الرئيسية:



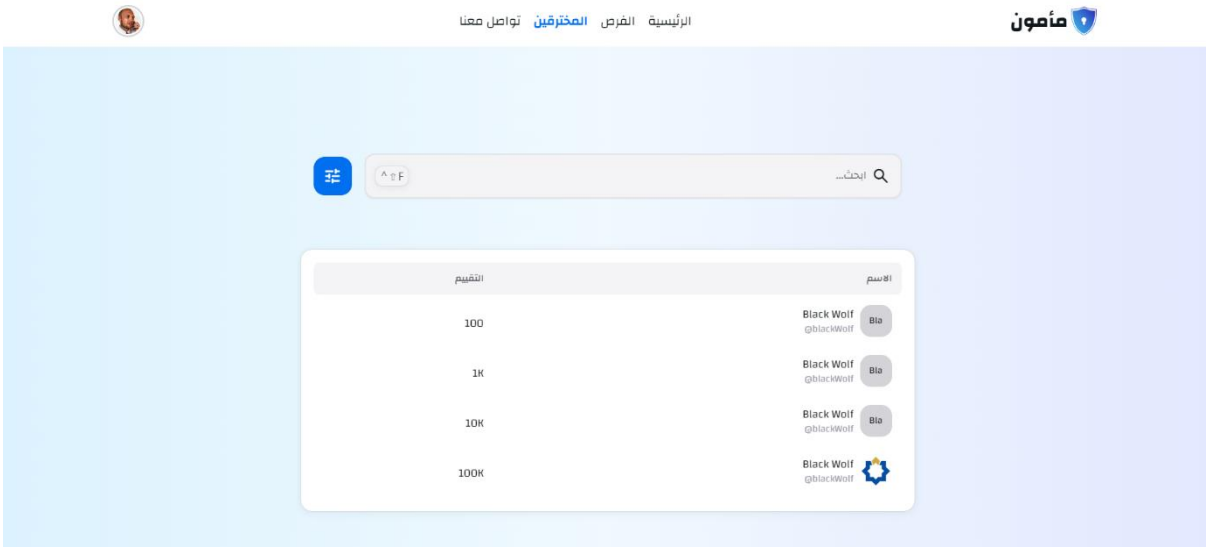
شكل 4-1:- الواجهة الرئيسية للنظام

4.2.2 واجهة الفرص:



شكل 2-4:- واجهة الفرص

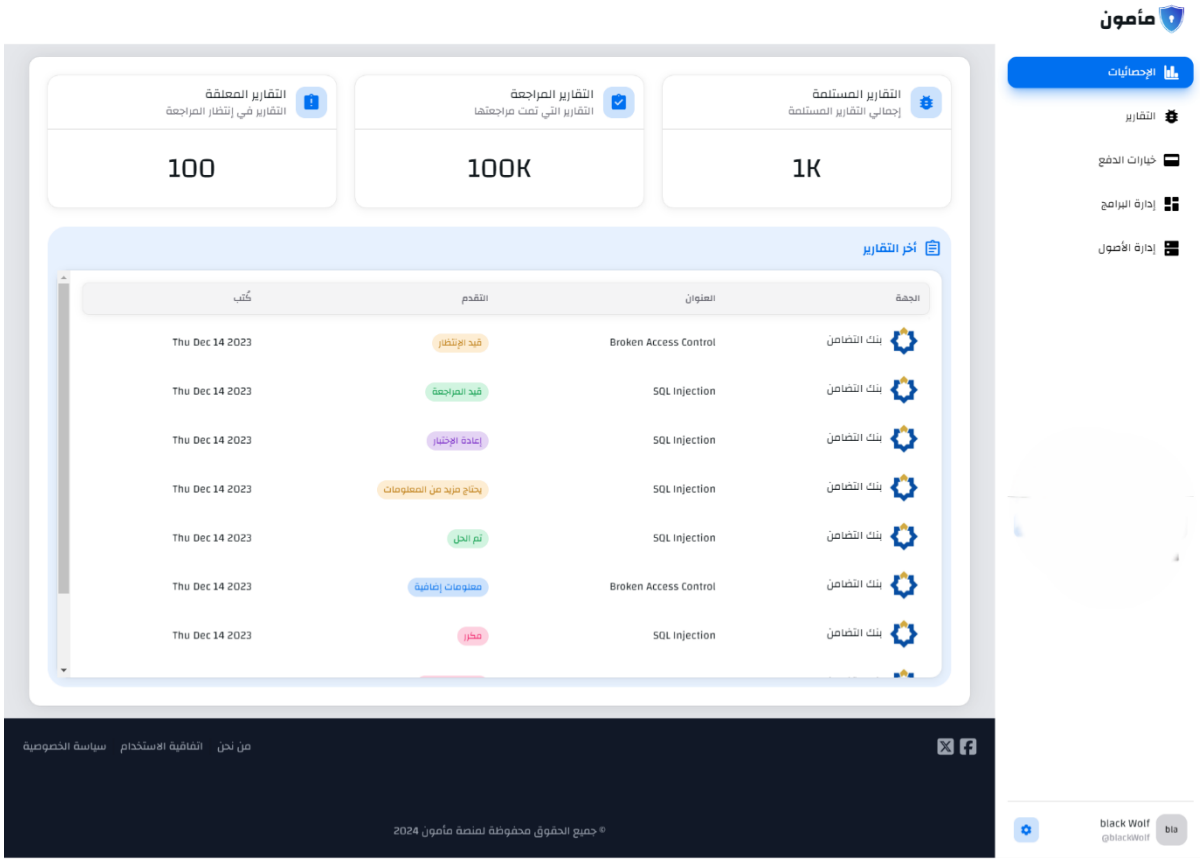
4.2.3 واجهة المخترقين الأخلاقيين:



شكل 3-4:- واجهة المخترقين

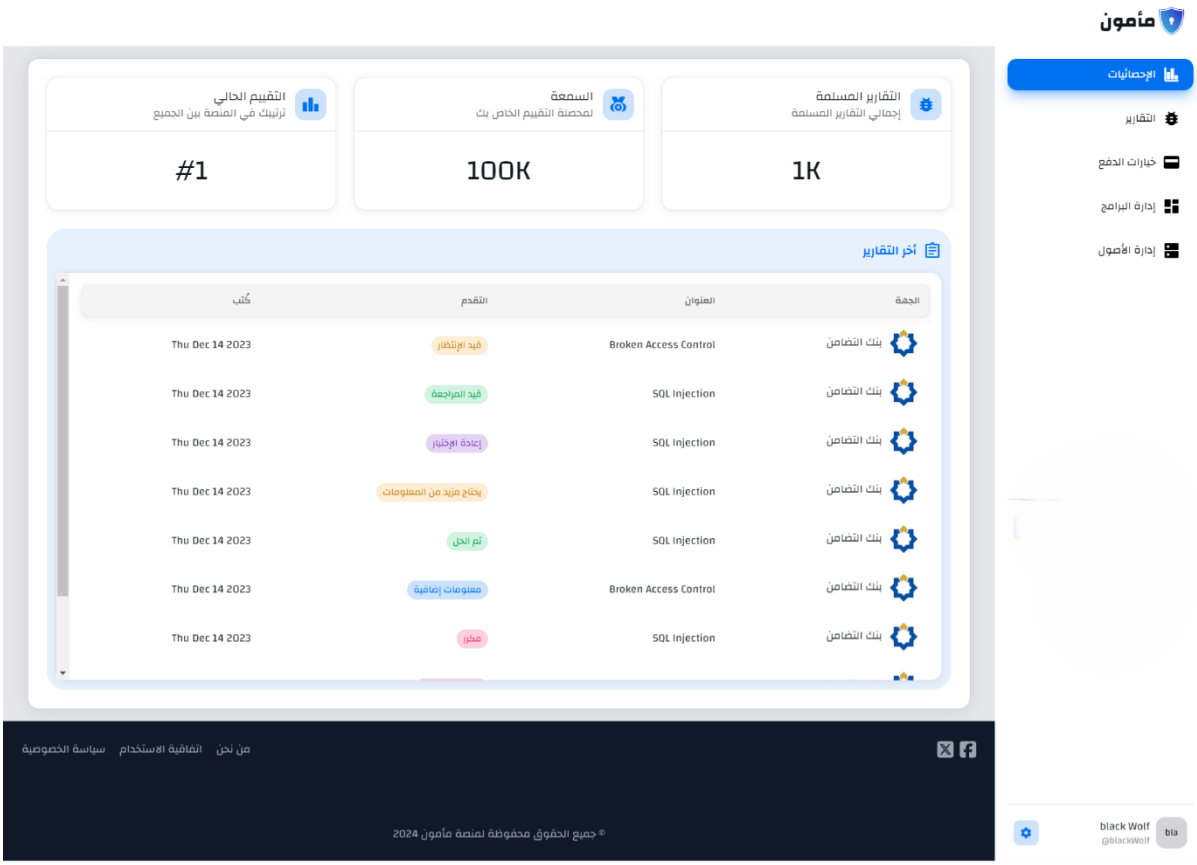
4.2.4 واجهات الإحصائيات:

4.2.4.1 واجهة احصائيات الشركة:



شكل 4-4:- واجهة الأحصائيات للشركة

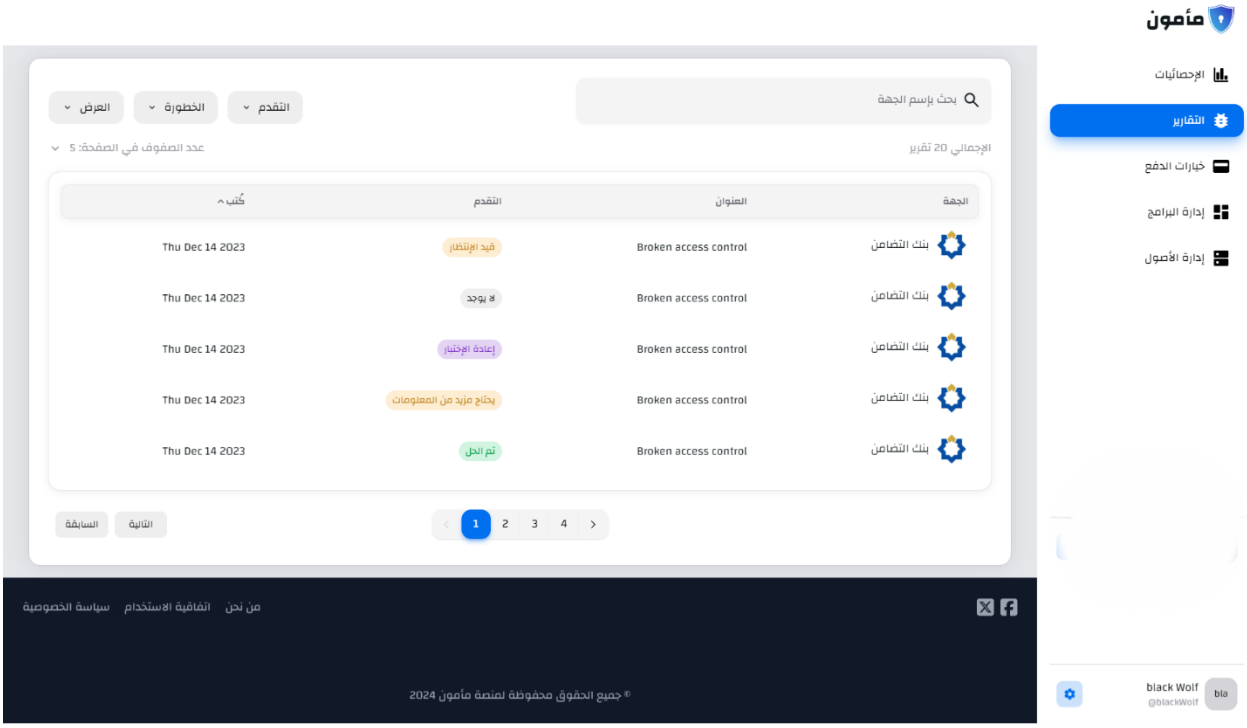
4.2.4.2 واجهة احصائيات المخترق الأخلاقي:



شكل 5-4 واجهة الأحصائيات للمخترق الأخلاقي

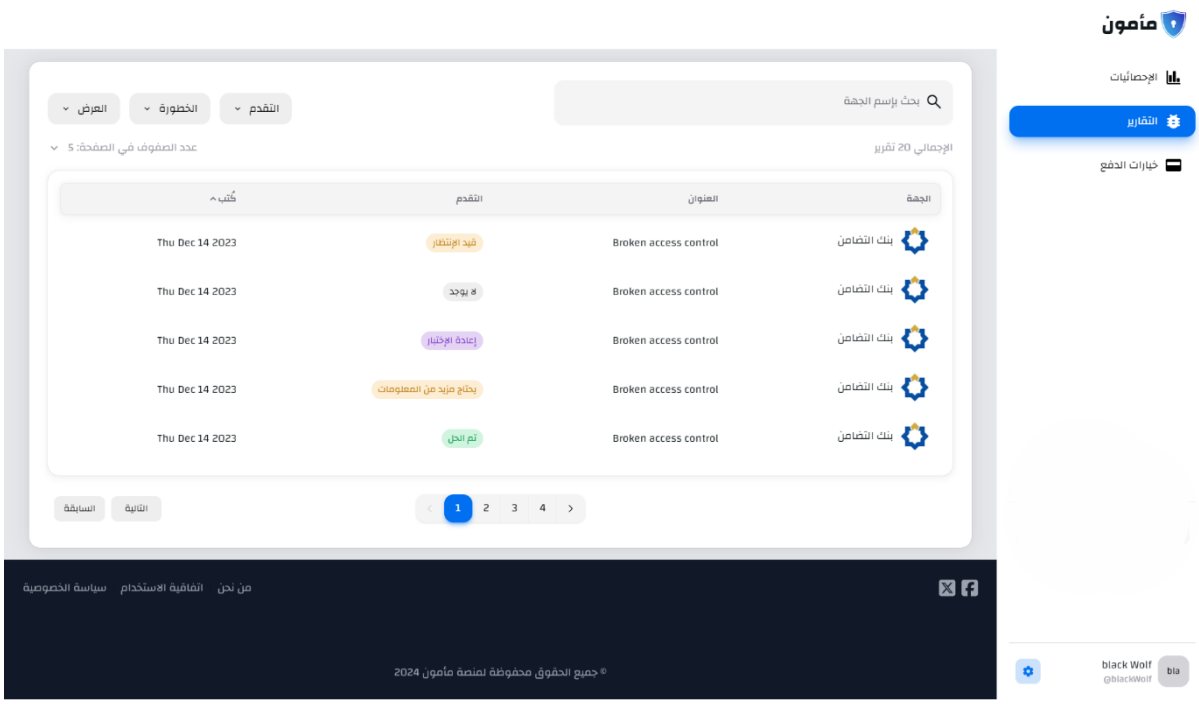
4.2.5 واجهات التقارير:

4.2.5.1 واجهة التقارير للشركة:



شكل4-6:- واجهة التقارير الخاص بالشركة

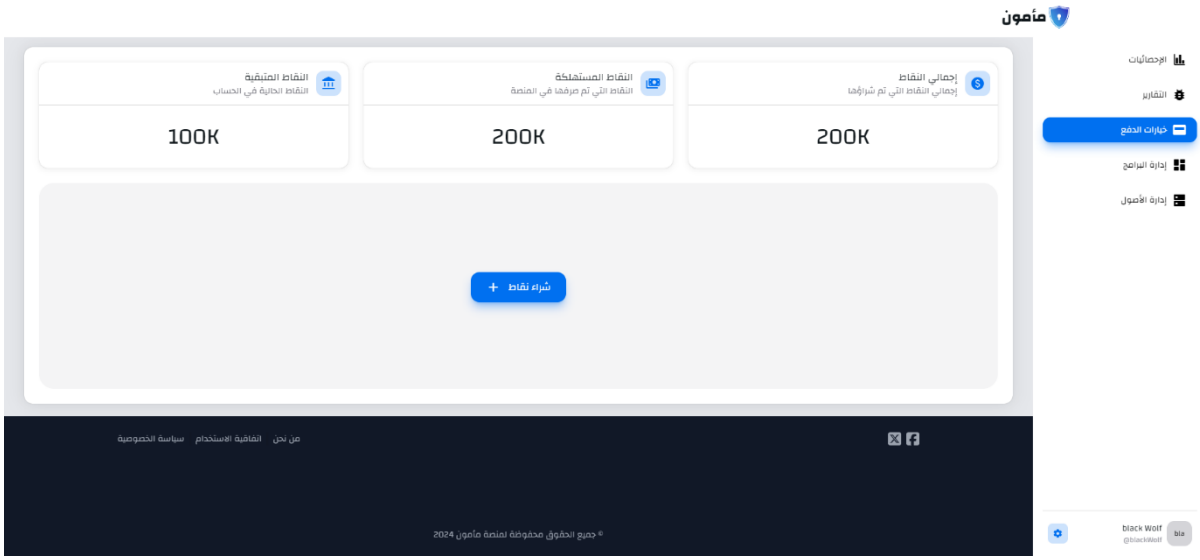
4.2.5.2 واجهة التقارير للمخترق الأخلاقي:



شكل 4-7 :- واجهة التقارير للمخترق الأخلاقي

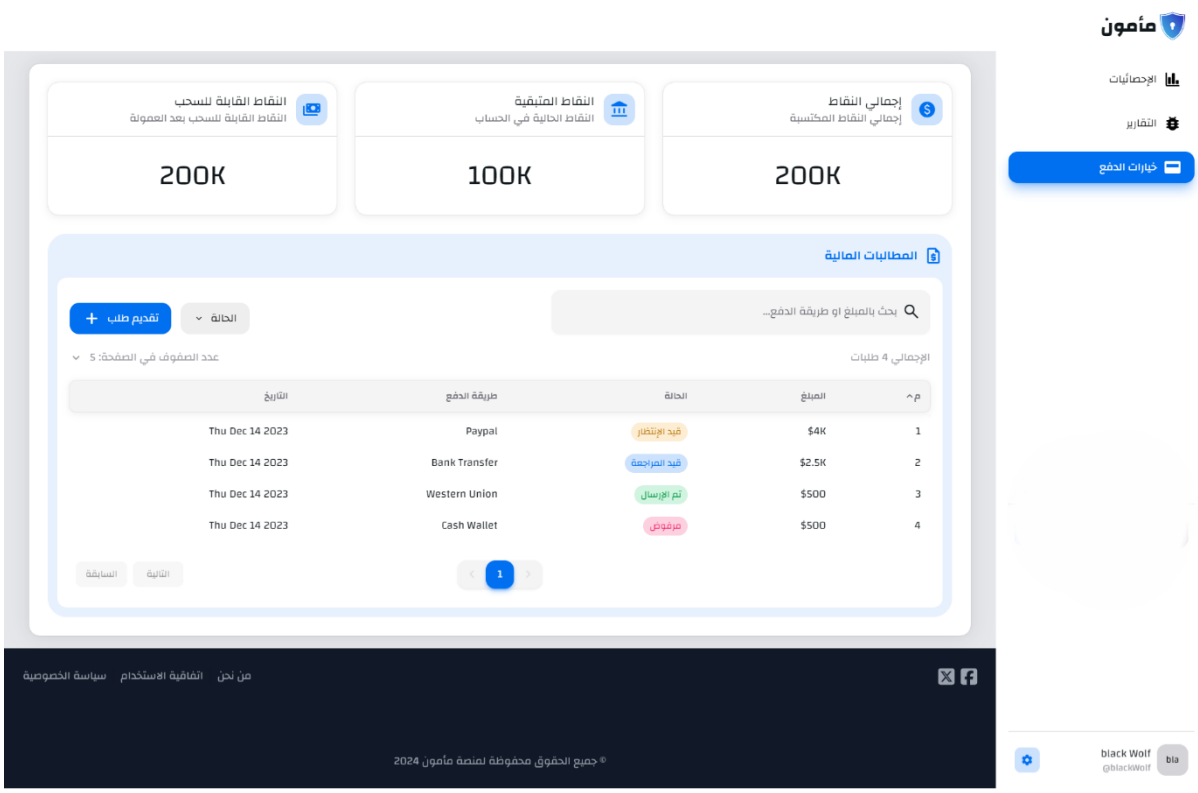
4.2.6 واجهات خيارات الدفع:

4.2.6.1 واجهة خيارات الدفع للشركة:



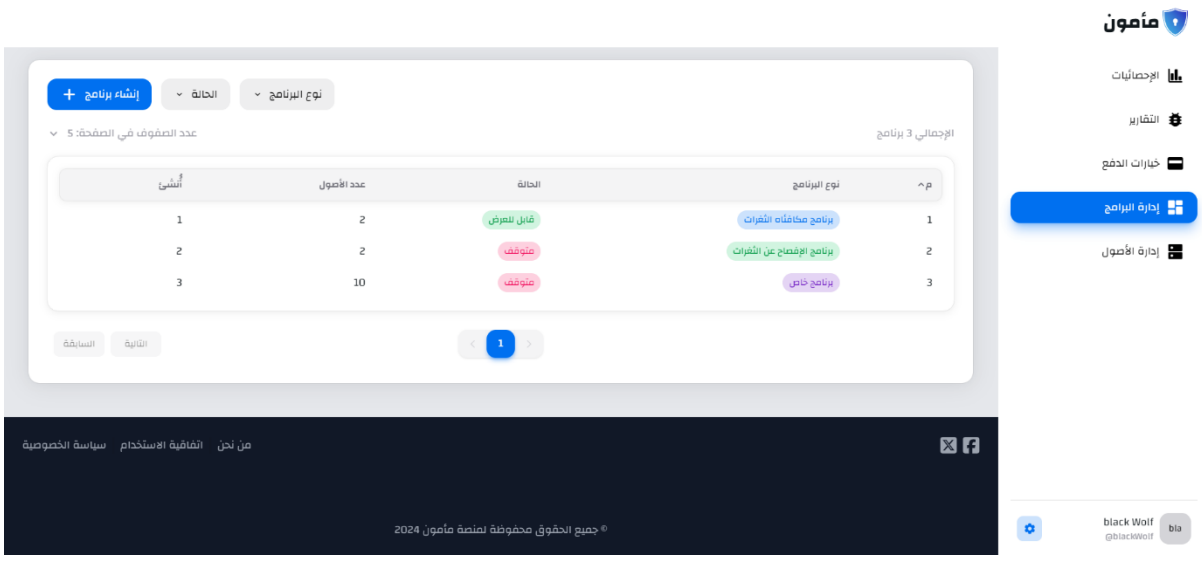
شكل 8-4 :- واجهة خيارات الدفع الخاص بالشركة

4.2.6.2 واجهة خيارات الدفع للمخترق الأخلاقي:



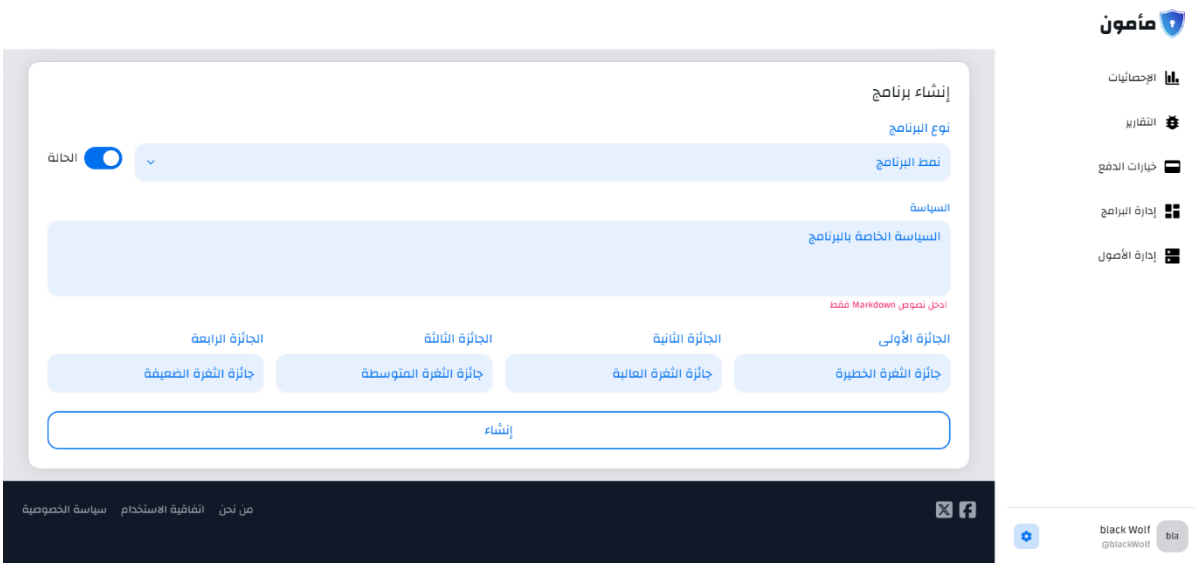
شكل 4-9 :- واجهة خيارات الدفع للمخترق الأخلاقي

4.2.7 واجهة إدارة البرامج:



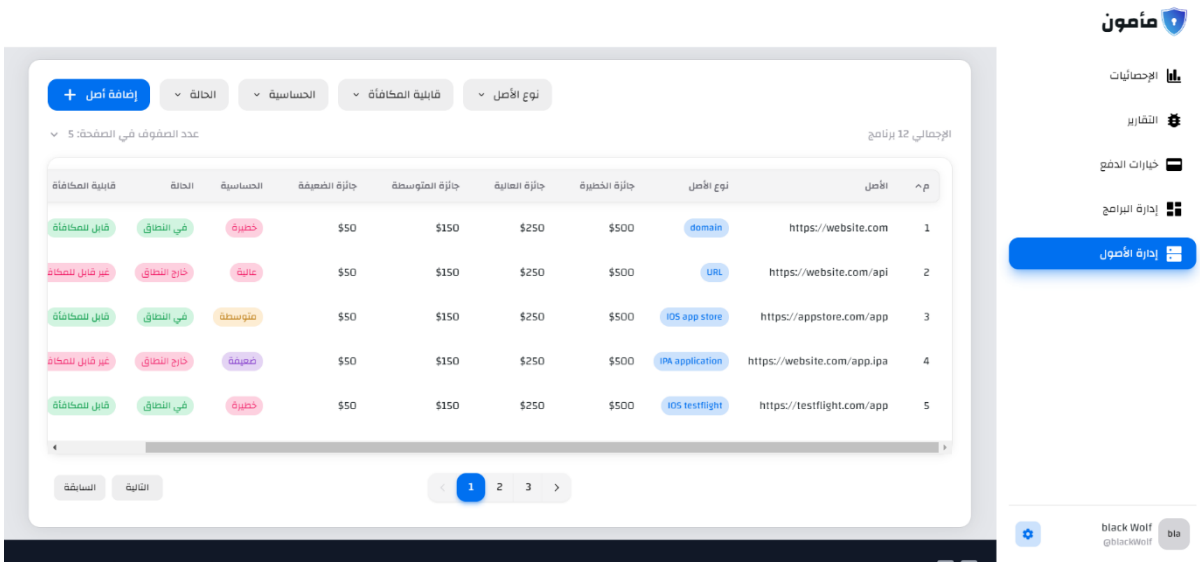
شكل 4-10 :- واجهة ادارة البرامج

4.2.8 واجهة انشاء برنامج:



شكل 4-11:- واجهة انشاء برنامج

4.2.9 واجهة إدارة الأصول:



شكل 4-12 :- واجهة ادارة الأصول

4.2.10 واجهة إضافة أصل:

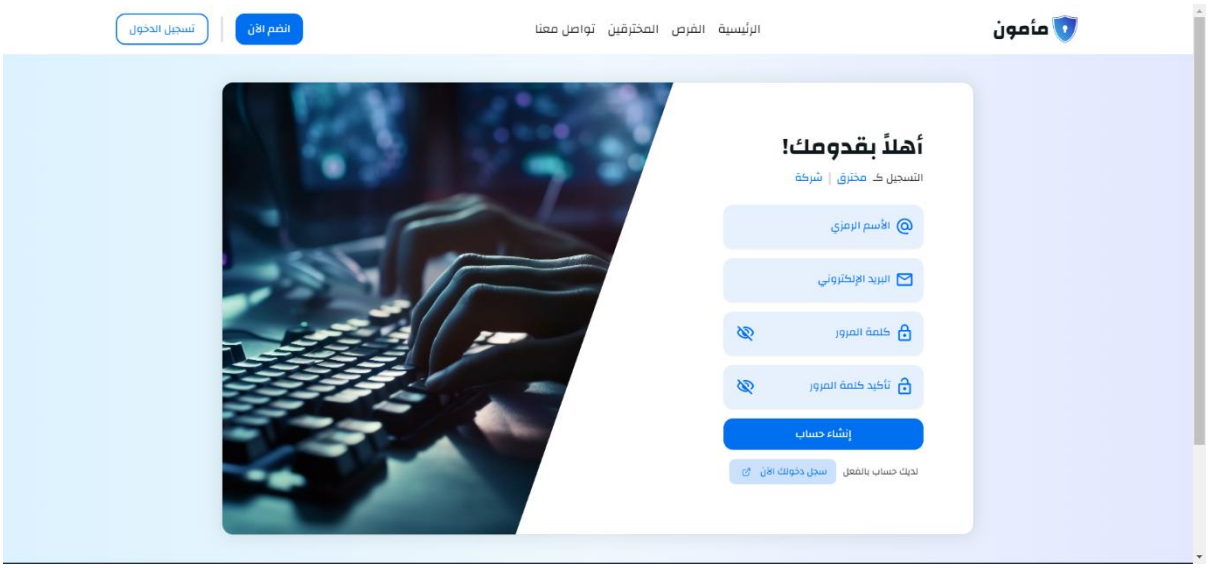
شكل 13-4:- واجهة اضافة الأصول

4.2.11 واجهة تسجيل الدخول:

شكل 14-4:- واجهة تسجيل الدخول

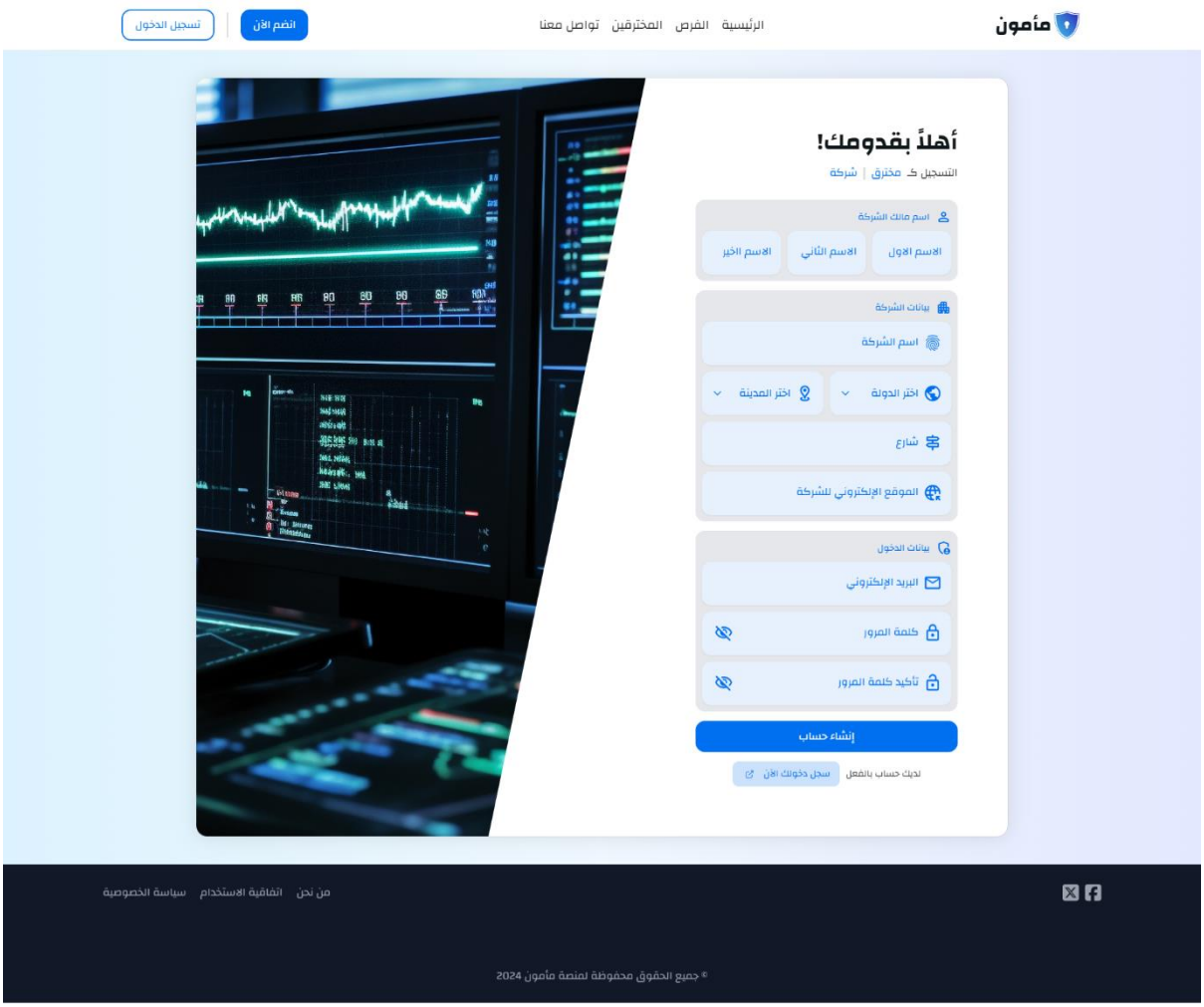
4.2.12 واجهة انشاء حساب:

4.2.12.1 انشاء حساب كمخترق:



شكل 15-4:- واجهة انشاء حساب كمخترق

4.2.12.2 انشاء حساب كشركة:



شكل16-4:- واجهة انشاء حساب كشركة

الخاتمة

4.3 الخاتمة:

تم انجاز هذا المشروع بحيث يكون سهل الأستخدام والوصول للبرامج وأن يكون باللغة العربية بحيث يسهل للجميع المشاركة فيه والحد من الثغرات الأمنية الموجودة والمنتشرة، وقد تم استخدام البرامج الحديثة في بناء الموقع مثل الـ Fast.api وعمل ربط بين الـ Front-End و Back-End بحيث يجعل التصفح للموقع ذو كفاءة العالية وتم بناء الموقع عن طريق مكتبة NextJS الخاصة ببناء الـ Front-End والذي ساعدت في بناء شكل الموقع وجعله أكثر تفاعلاً وأسهل أستخداماً وكذلك تم أستخدام Supabase والذي ساعدت في انشاء قاعدة البيانات الخاصة بالموقع.

4.4 جوانب القصور:

1. **الاقتصار على التعليم المجاني وعدم وجود المراجع الكافية.**
2. **لم نستطيع ربط الموقع فعلي على بوابات الدفع الإلكتروني.**

4.5 الأعمال المستقبلية:

1. **نشر المنصة على الأنترنت.**
2. **ربط المنصة بطرق الدفع المحلية والدولية لتصبح متاحه للجميع.**

المراجع والملاحق

4.6 المراجع:-

[1] Shelly Cashman Series, Systems Analysis and Design, ELEVENTH EDITION,2017.

[2] Mark Lutz, Learning Python, FOURTH EDITION,2009.

[3] Fullstack.io, 30 Day of React an introduction to react in 30 bite size morsels.

[4] Boris Cherny, Programming TypeScript Making Your JavaScript Applications Scale,2019.

- https://nextui.org/
- https://stackoverflow.com/
- https://fastapi.tiangolo.com/
- https://nextjs.org/
- https://supabase.com
- https://www.python.org/
- https://www.w3schools.com/python/python_reference.asp
- https://docs.python.org/3/download.html
- https://www.postgresql.org/
- https://react.dev/
- https://www.typescriptlang.org/
- https://www.javascript.com/
- https://www.w3schools.com/js/
- https://www.w3schools.com/typescript/typescript_intro.php
- https://www.w3schools.com/postgresql/index.php

4.7 الملاحق:

```
import React from "react";
import {
  Table,
  TableHeader,
  TableColumn,
  TableBody,
  TableRow,
  TableCell,
  Input,
  Button,
  DropdownTrigger,
  Dropdown,
  DropdownMenu,
  DropdownItem,
  Chip,
  User,
  Pagination,
  Selection,
  ChipProps,
  SortDescriptor
} from "@nextui-org/react";
import {PlusIcon} from "./PlusIcon";
```

```
import {VerticalDotsIcon} from "./VerticalDotsIcon";
import {ChevronDownIcon} from "./ChevronDownIcon";
import {SearchIcon} from "./SearchIcon";
import {columns, users, statusOptions} from "./data";
import {capitalize} from "./utils";

const statusColorMap: Record<string, ChipProps["color"]> = {
  active: "success",
  paused: "danger",
  vacation: "warning",
};

const INITIAL_VISIBLE_COLUMNS = ["name", "role", "status",
"actions"];

type User = typeof users[0];

export default function App() {
  const [filterValue, setFilterValue] = React.useState("");
  const [selectedKeys, setSelectedKeys] =
React.useState<Selection>(new Set([]));
  const [visibleColumns, setVisibleColumns] =
React.useState<Selection>(new Set(INITIAL_VISIBLE_COLUMNS));
```

```
  const [statusFilter, setStatusFilter] =
React.useState<Selection>("all");
  const [rowsPerPage, setRowsPerPage] = React.useState(5);
  const [sortDescriptor, setSortDescriptor] =
React.useState<SortDescriptor>({
    column: "age",
    direction: "ascending",
  });

  const [page, setPage] = React.useState(1);

  const hasSearchFilter = Boolean(filterValue);

  const headerColumns = React.useMemo(() => {
    if (visibleColumns === "all") return columns;

    return columns.filter((column) =>
Array.from(visibleColumns).includes(column.uid));
  }, [visibleColumns]);

  const filteredItems = React.useMemo(() => {
    let filteredUsers = [...users];

    if (hasSearchFilter) {
```

```
      filteredUsers = filteredUsers.filter((user) =>
        user.name.toLowerCase().includes(filterValue.toLowerCase()),
      );
    }
    if (statusFilter !== "all" && Array.from(statusFilter).length !==
statusOptions.length) {
      filteredUsers = filteredUsers.filter((user) =>
        Array.from(statusFilter).includes(user.status),
      );
    }

    return filteredUsers;
  }, [users, filterValue, statusFilter]);

  const pages = Math.ceil(filteredItems.length / rowsPerPage);

  const items = React.useMemo(() => {
    const start = (page - 1) * rowsPerPage;
    const end = start + rowsPerPage;

    return filteredItems.slice(start, end);
  }, [page, filteredItems, rowsPerPage]);
```

```
  const sortedItems = React.useMemo(() => {
    return [...items].sort((a: User, b: User) => {
      const first = a[sortDescriptor.column as keyof User] as number;
      const second = b[sortDescriptor.column as keyof User] as
number;
      const cmp = first < second ? -1 : first > second ? 1 : 0;


      return sortDescriptor.direction === "descending" ? -cmp : cmp;
    });
  }, [sortDescriptor, items]);


  const renderCell = React.useCallback((user: User, columnKey:
React.Key) => {
    const cellValue = user[columnKey as keyof User];


    switch (columnKey) {
      case "name":
        return (
          <User
            avatarProps={{radius: "lg", src: user.avatar}}
            description={user.email}
            name={cellValue}
          >
            {user.email}
```

```
        </User>
      );
    case "role":
      return (
        <div className="flex flex-col">
          <p className="text-bold text-small
capitalize">{cellValue}</p>
          <p className="text-bold text-tiny capitalize text-default-
400">{user.team}</p>
        </div>
      );
    case "status":
      return (
        <Chip className="capitalize"
color={statusColorMap[user.status]} size="sm" variant="flat">
          {cellValue}
        </Chip>
      );
    case "actions":
      return (
        <div className="relative flex justify-end items-center gap-
2">
          <Dropdown>
            <DropdownTrigger>
```

```
            <Button isIconOnly size="sm" variant="light">
              <VerticalDotsIcon className="text-default-300" />
            </Button>
          </DropdownTrigger>
          <DropdownMenu>
            <DropdownItem>View</DropdownItem>
            <DropdownItem>Edit</DropdownItem>
            <DropdownItem>Delete</DropdownItem>
          </DropdownMenu>
        </Dropdown>
      </div>
    );
  default:
    return cellValue;
 }
}, []);

const onNextPage = React.useCallback(() => {
 if (page < pages) {
  setPage(page + 1);
 }
}, [page, pages]);
```

```
  const onPreviousPage = React.useCallback(() => {
    if (page > 1) {
      setPage(page - 1);
    }
  }, [page]);

  const onRowsPerPageChange = React.useCallback((e:
React.ChangeEvent<HTMLSelectElement>) => {
    setRowsPerPage(Number(e.target.value));
    setPage(1);
  }, []);

  const onSearchChange = React.useCallback((value?: string) => {
    if (value) {
      setFilterValue(value);
      setPage(1);
    } else {
      setFilterValue("");
    }
  }, []);

  const onClear = React.useCallback(()=>{
    setFilterValue("")
```

```
    setPage(1)
  },[])

  const topContent = React.useMemo(() => {
    return (
      <div className="flex flex-col gap-4">
        <div className="flex justify-between gap-3 items-end">
          <Input
            isClearable
            className="w-full sm:max-w-[44%]"
            placeholder="Search by name..."
            startContent={<SearchIcon />}
            value={filterValue}
            onClear={() => onClear()}
            onValueChange={onSearchChange}
          />
          <div className="flex gap-3">
            <Dropdown>
              <DropdownTrigger className="hidden sm:flex">
                <Button endContent={<ChevronDownIcon
className="text-small" />} variant="flat">
                  Status
                </Button>
```

```
            </DropdownTrigger>
            <DropdownMenu
              disallowEmptySelection
              aria-label="Table Columns"
              closeOnSelect={false}
              selectedKeys={statusFilter}
              selectionMode="multiple"
              onSelectionChange={setStatusFilter}
            >
              {statusOptions.map((status) => (
                <DropdownItem key={status.uid}
className="capitalize">
                  {capitalize(status.name)}
                </DropdownItem>
              ))}
            </DropdownMenu>
          </Dropdown>
          <Dropdown>
            <DropdownTrigger classNamе="hidden sm:flex">
              <Button endContent={<ChevronDownIcon
className="text-small" />} variant="flat">
                Columns
              </Button>
            </DropdownTrigger>
```

```
            <DropdownMenu
              disallowEmptySelection
              aria-label="Table Columns"
              closeOnSelect={false}
              selectedKeys={visibleColumns}
              selectionMode="multiple"
              onSelectionChange={setVisibleColumns}
            >
              {columns.map((column) => (
                <DropdownItem key={column.uid}
className="capitalize">
                  {capitalize(column.name)}
                </DropdownItem>
              ))}
            </DropdownMenu>
          </Dropdown>
          <Button color="primary" endContent={<PlusIcon />}>
            Add New
          </Button>
        </div>
      </div>
      <div className="flex justify-between items-center">
        <span className="text-default-400 text-small">Total
{users.length} users</span>
```

```
        <label className="flex items-center text-default-400 text-small">
          Rows per page:
          <select
            className="bg-transparent outline-none text-default-400 text-small"
            onChange={onRowsPerPageChange}
          >
            <option value="5">5</option>
            <option value="10">10</option>
            <option value="15">15</option>
          </select>
        </label>
      </div>
    </div>
  );
}, [
  filterValue,
  statusFilter,
  visibleColumns,
  onSearchChange,
  onRowsPerPageChange,
  users.length,
  hasSearchFilter,
```

```
  ]);


  const bottomContent = React.useMemo(() => {
    return (
      <div className="py-2 px-2 flex justify-between items-center">
        <span className="w-[30%] text-small text-default-400">
          {selectedKeys === "all"
            ? "All items selected"
            : `${selectedKeys.size} of ${filteredItems.length} selected`}
        </span>
        <Pagination
          isCompact
          showControls
          showShadow
          color="primary"
          page={page}
          total={pages}
          onChange={setPage}
        />
        <div className="hidden sm:flex w-[30%] justify-end gap-2">
          <Button isDisabled={pages === 1} size="sm" variant="flat"
onPress={onPreviousPage}>
            Previous
```

```
        </Button>
        <Button isDisabled={pages === 1} size="sm" variant="flat"
onPress={onNextPage}>
          Next
        </Button>
      </div>
    </div>
  );
 }, [selectedKeys, items.length, page, pages, hasSearchFilter]);

 return (
  <Table
    aria-label="Example table with custom cells, pagination and
sorting"
    isHeaderSticky
    bottomContent={bottomContent}
    bottomContentPlacement="outside"
    classNames={{
      wrapper: "max-h-[382px]",
    }}
    selectedKeys={selectedKeys}
    selectionMode="multiple"
    sortDescriptor={sortDescriptor}
    topContent={topContent}
```

```
    topContentPlacement="outside"
    onSelectionChange={setSelectedKeys}
    onSortChange={setSortDescriptor}
  >
    <TableHeader columns={headerColumns}>
      {(column) => (
        <TableColumn
          key={column.uid}
          align={column.uid === "actions" ? "center" : "start"}
          allowsSorting={column.sortable}
        >
          {column.name}
        </TableColumn>
      )}
    </TableHeader>
    <TableBody emptyContent={"No users found"}
items={sortedItems}>
      {(item) => (
        <TableRow key={item.id}>
          {(columnKey) => <TableCell>{renderCell(item,
columnKey)}</TableCell>}
        </TableRow>
      )}
    </TableBody>
```

```
    </Table>
  );
}
  import { createClient } from '@/utils/supabase/server';

  export default async function Notes() {
    const supabase = createClient();
    const { data: notes } = await supabase.from("notes").select();

    return <pre>{JSON.stringify(notes, null, 2)}</pre>
  }
  import { createServerClient, type CookieOptions } from
'@supabase/ssr'
  import { cookies } from 'next/headers'

  export function createClient() {
    const cookieStore = cookies()

    return createServerClient(
      process.env.NEXT_PUBLIC_SUPABASE_URL!,
      process.env.NEXT_PUBLIC_SUPABASE_ANON_KEY!,
      {
        cookies: {
```

```
        get(name: string) {
          return cookieStore.get(name)?.value
        },
        set(name: string, value: string, options: CookieOptions) {
          try {
            cookieStore.set({ name, value, ...options })
          } catch (error) {
            // The `set` method was called from a Server Component.
            // This can be ignored if you have middleware refreshing
            // user sessions.
          }
        },
        remove(name: string, options: CookieOptions) {
          try {
            cookieStore.set({ name, value: '', ...options })
          } catch (error) {
            // The `delete` method was called from a Server
Component.
            // This can be ignored if you have middleware refreshing
            // user sessions.
          }
        },
      },
```

```
    }
  )
 }
import { createServerClient, type CookieOptions } from
'@supabase/ssr'
import { NextResponse, type NextRequest } from 'next/server'

export async function updateSession(request: NextRequest) {
 let response = NextResponse.next({
  request: {
   headers: request.headers,
  },
 })

 const supabase = createServerClient(
  process.env.NEXT_PUBLIC_SUPABASE_URL!,
  process.env.NEXT_PUBLIC_SUPABASE_ANON_KEY!,
  {
   cookies: {
    get(name: string) {
     return request.cookies.get(name)?.value
    },
    set(name: string, value: string, options: CookieOptions) {
```

```
        request.cookies.set({
          name,
          value,
          ...options,
        })
        response = NextResponse.next({
          request: {
            headers: request.headers,
          },
        })
        response.cookies.set({
          name,
          value,
          ...options,
        })
      },
      remove(name: string, options: CookieOptions) {
        request.cookies.set({
          name,
          value: '',
          ...options,
        })
        response = NextResponse.next({
```

```
          request: {
            headers: request.headers,
          },
        })
        response.cookies.set({
          name,
          value: '',
          ...options,
        })
      },
    },
  }
)

  await supabase.auth.getUser()

  return response
}

import React from "react";
import {Navbar, NavbarBrand, NavbarContent, NavbarItem, Link,
Button} from "@nextui-org/react";
import {AcmeLogo} from "./AcmeLogo.jsx";
```

```
export default function App() {
  return (
    <Navbar isBordered>
      <NavbarBrand>
        <AcmeLogo />
        <p className="font-bold text-inherit">ACME</p>
      </NavbarBrand>
      <NavbarContent className="hidden sm:flex gap-4"
justify="center">
        <NavbarItem>
          <Link color="foreground" href="#">
            Features
          </Link>
        </NavbarItem>
        <NavbarItem isActive>
          <Link href="#" aria-current="page">
            Customers
          </Link>
        </NavbarItem>
        <NavbarItem>
          <Link color="foreground" href="#">
            Integrations
```

```
        </Link>
      </NavbarItem>
    </NavbarContent>
    <NavbarContent justify="end">
      <NavbarItem className="hidden lg:flex">
        <Link href="#">Login</Link>
      </NavbarItem>
      <NavbarItem>
        <Button as={Link} color="primary" href="#" variant="flat">
          Sign Up
        </Button>
      </NavbarItem>
    </NavbarContent>
  </Navbar>
 );
}

from typing import Annotated

from fastapi import Body, FastAPI
from pydantic import BaseModel

app = FastAPI()
```

```
class Item(BaseModel):
    name: str
    description: str | None = None
    price: float
    tax: float | None = None


class User(BaseModel):
    username: str
    full_name: str | None = None


@app.put("/items/{item_id}")
async def update_item(
    *,
    item_id: int,
    item: Item,
    user: User,
    importance: Annotated[int, Body(gt=0)],
    q: str | None = None,
):
```

```
    results = {"item_id": item_id, "item": item, "user": user,
"importance": importance}
    if q:
        results.update({"q": q})
    return results
from fastapi import FastAPI
from fastapi.middleware.cors import CORSMiddleware


app = FastAPI()


origins = [
    "http://localhost.tiangolo.com",
    "https://localhost.tiangolo.com",
    "http://localhost",
    "http://localhost:8080",
]


app.add_middleware(
    CORSMiddleware,
    allow_origins=origins,
    allow_credentials=True,
    allow_methods=["*"],
    allow_headers=["*"],
```

```
)


@app.get("/")
async def main():
    return {"message": "Hello World"}
import cookies from 'next/headers'
import { db } from '@/app/lib/db'
import { encrypt } from '@/app/lib/session'


export async function createSession(id: number) {
  const expiresAt = new Date(Date.now() + 7 * 24 * 60 * 60 * 1000)


  // 1. Create a session in the database
  const data = await db
    .insert(sessions)
    .values({
      userId: id,
      expiresAt,
    })
    // Return the session ID
    .returning({ id: sessions.id })
```

```
  const sessionId = data[0].id

  // 2. Encrypt the session ID
  const session = await encrypt({ sessionId, expiresAt })

  // 3. Store the session in cookies for optimistic auth checks
  cookies().set('session', session, {
    httpOnly: true,
    secure: true,
    expires: expiresAt,
    sameSite: 'lax',
    path: '/',
  })
}

from typing import Annotated

from fastapi import FastAPI, Path
from pydantic import BaseModel

app = FastAPI()
```

```
class Item(BaseModel):
    name: str
    description: str | None = None
    price: float
    tax: float | None = None



@app.put("/items/{item_id}")
async def update_item(
    item_id: Annotated[int, Path(title="The ID of the item to get",
ge=0, le=1000)],
    q: str | None = None,
    item: Item | None = None,
):
    results = {"item_id": item_id}
    if q:
        results.update({"q": q})
    if item:
        results.update({"item": item})
    return results
{
    "name": "Foo",
    "description": "The pretender",
```

```
    "price": 42.0,
    "tax": 3.2
}
import { cookies } from 'next/headers'

export default function Page() {
  const cookieStore = cookies()
  const theme = cookieStore.get('theme')
  return '...'
}
import { cookies } from 'next/headers'

export default function Page() {
  const cookieStore = cookies()
  return cookieStore.getAll().map((cookie) => (
    <div key={cookie.name}>
      <p>Name: {cookie.name}</p>
      <p>Value: {cookie.value}</p>
    </div>
  ))
}
import { cookies } from 'next/headers'
```

```
export default function Page() {
  const cookieStore = cookies()
  const hasCookie = cookieStore.has('theme')
  return '...'
}
'use server'

import { cookies } from 'next/headers'

async function create(data) {
  cookies().set('name', 'lee')
  // or
  cookies().set('name', 'lee', { secure: true })
  // or
  cookies().set({
    name: 'name',
    value: 'lee',
    httpOnly: true,
    path: '/',
  })
}'use server'

import { cookies } from 'next/headers'
```

```
async function delete(data) {
  cookies().delete('name')
}
'use server'

import { cookies } from 'next/headers'

async function delete(data) {
  const oneDay = 24 * 60 * 60 * 1000
  cookies().set('name', 'value', { expires: Date.now() - oneDay })
                                                                .{
```