Republic of Yemen

Ministry of Education and Scientific Research

Emiratis International University

Information Technology Department

Information Security

الجمهورية اليمنية

وزاره التربية والتعليم والبحث العلمي

الجامعة الاملارتية الدولية

قسم وتكنولوجيا المعلومات

قسم أمن المعلومات

Cyber Pass

فريق المشروع:

حسام طارق زايد

محمد حسن المحويتي

هاشم حسن المداني

عبد الله فتحي العامري

علي علي إسماعيل

مشرف المشروع

أ.م. د/جميل راشد

المشرف المساعد:

أ/ باسل باسرده

تقرير مشروع التخرج المقدم لقسم أمن المعلومات كجزء من متطلبات الحصول على درجة البكالوريوس في أمن المعلومات

1446هـ 2025م

هو تطبيق أو نظام مصمم لمساعدة المستخدمين على إنشاء، وتخزين، وتنظيم كلمات السر الخاصة بهم بطريقة آمنة وسهلة الوصول إليها.

يهدف هذا النوع من المشاريع إلى تزويد المستخدمين بحلول مريحة وآمنة للحفاظ على كلمات السر المختلفة التي يحتاجونها في حياتهم الرقمية اليومية، دون الحاجة لتذكر جميع كلمات المرور لكل حساب.

إقرار المشرف

أشهد أن إعداد هذا المشروع بعنوان "Cyber Pass" تطبيق لحفظ كلمات المرور والذي تم إعداده من قبل مجموعة من طلاب أمن المعلومات من كلية الهندسة- قسم أمن المعلومات كان تحت إشرافي كمتطلب جزئي للحصول على درجة البكالوريوس في أمن المعلومات.

والله ولي التوفيق،،،

اسم المشرف:

التوقيع:

التاريخ:

الـمـشـروع بـعـنـوان"Cyber Pass"

مــشــرف

م	الاسم	الصفة	التوقيع
1	أ.م. د/جميل راشد	رئيس قسم أمن المعلومات	

لجنة المناقشة

م	الاسم	الصفة	التوقيع
1	أ.م. د/جميل راشد	رئيس قسم أمن المعلومات (مشرف المشروع)	
2	د. هشام عقلان	عضو لجنة المناقشة	
3	د. مالك الجبري	عضو لجنة المناقشة	

رئيس القسم/ أ.م. د/جميل راشد

التوقيع/

الاهداء

نهدي هذا المشروع لكل من كان له الفضل بعد الله ببلوغنا هذا المقام، وكل من ضحى وعانى من اجلنا واحترق شوقاً ليعيش لحظات وصولنا لهانية هذا الدرب ... ومن بذل الغالي والرخيص وسعى لتشجعنا وحثنا،،،

عائلاتنا: إليكم نهدي هذا المشروع ولتكونوا فخورين مسرورين بنجحنا.

ولا ننسى كل من بذل دمه فداءً لدين الله وفداءً لهذه الامة، وكل الشهداء من أخواننا في لبنان وفلسطين....

شكر وعرفان

جزيل الشكر وخالص التقدير نقدمه لكل من كانت له بصمه او بذل من الجهد ما تمكنّا عبره لنصل الي هذا النقطة ولنخرج حاملين ما نحمله من المعرفة،

أساتذتنا الكرام من دكاترة ومدرسين أنما تستحقونه من امتنان سنظل نحمله في قلوبنا ولا يمكن أن نستوفيكم حقكم.

نخص بالذكر من كان له الحظ الاوفر في إتمام هذا العمل واخذ على عاتقة مسؤولية الاشراف على هذا المشروع:

أ.م. د/جميل راشد

والمشرف المساعد:

أ. باسل باسردة

ولا ننسى معيدة القسم:

أ. علياء العراسي

كما نتقدم بخالص الشكر والتقدير لرئيس قسم أمن المعلومات:

أ.م. د/جميل راشد

جدول المحتويات

الـــفـــــصـــل الأول

❖ الفصل الأول: المقدمة

1.1 نظرة عامة

في عالمنا الرقمي المتسارع أصبحنا نستخدم العديد من الحسابات عبر الإنترنت، مما يتطلب منا تذكر عدد كبير من كلمات السر المعقدة هذه الحاجة تؤدى إلى تحديات كبيرة، مثل نسيان كلمات السر، وإعادة استخدام نفس كلمة السر في عدة حسابات ومع تزايد التهديدات السيبرانية، أصبح حماية البيانات الشخصية أمرًا بالغ الأهمية حيث تلعب كلمات السر دورا حاسما في هذا السياق، حيث أنها بمثابة مفتاح الوصول إلى حساباتنا المختلفة مما يعرض أمننا الرقمي للخطر.

يهدف هذا المشروع إلى تقديم حل عملي لهذه المشكلة من خلال تطوير نظام إدارة كلمات سر

(Cyber Pass) وتسليط الضوء على أهمية إدارة كلمات السر بشكل صحيح، وتقديم حلول تقنية مبتكرة لتعزيز أمن المعلومات.

1.2 بيان المشكلة

تُعاني إدارة كلمات السر من تحديات عديدة، أبرزها:

- تشفير البيانات: ما هي خوارزميات التشفير التي ستستخدمها لحماية كلمات السر المخزنة؟ وكيف ستضمن أن هذه الخوارزميات قوية وآمنة.
- تخزين البيانات: هل تخزين محلي أم سحابي؟ هل نخزن بيانات المستخدمين بشكل محلي على الجهاز، أم سنستخدم خدمة تخزين سحابية؟ لكل خيار مزاياه وعيوبه من حيث الأمان والأداء.
- التوافق: كيف ستضمن أن النظام متوافق مع التحديثات المستمرة لأنظمة التشغيل والمتصفحات؟
- الثقة: كيف ستقنع المستخدمين بأن نظامك آمن وموثوق به، وأنهم يمكنهم الوثوق به لحماية كلمات السر الخاصة بهم؟
- التعليم: كيف ستعلم المستخدمين أهمية استخدام مدير كلمات السر وكيفية استخدامه بشكل صحيح؟
- التميز: كيف ستتميز عن باقي تطبيقات إدارة كلمات السر الموجودة في السوق.

1.3 أهداف المشروع

يهدف مشروع إدارة كلمات المرور إلى توفير حلول آمنة وفعالة لإدارة وتخزين كلمات المرور الخاصة بالمستخدمين بإضافة الى هذه المشاريع إلى تحقيق الأهداف التالية:

- حماية كلمات المرور من الاختراق والسرقة عن طريق تخزينها مشفرة في قاعدة بيانات آمنة.
- منع إعادة استخدام نفس كلمات المرور في عدة حسابات، مما يقلل من خطر تعرض الحسابات للاختراق.
- توفير مولد آلي لكلمات مرور قوية ومعقدة يصعب تخمينها.
- تقليل مخاطر الاختراقات والاحتيال الإلكتروني.
- نشر ثقافة أمن المعلومات بين المستخدمين وتشجيعهم على اتباع أفضل الممارسات لحماية بياناتهم.

أهداف فرعية:

- دعم خاصية التوليد التلقائي لكلمات المرور القوية.
- تقديم التشفير التام لكلمات المرور.
- توفير واجهة مستخدم سهله وآمنة.
- تحليل لكلمات السر وفحص قوتها

1.4 تعريف المشروع

إدارة كلمات السر (Cyber Pass) هو تطبيق برمجي مصمم بلغة بايثون لحفظ وتنظيم كلمات السر الخاصة بك في مكان واحد آمن مما يوفر عليك عناء تذكر العديد من كلمات السر المعقدة يعمل هذا التطبيق عن طريق تخزين كلمات السر المشفرة خلف كلمة مرور رئيسية واحدة، ويتيح لك الوصول إليها بسهولة عند الحاجة. بالإضافة إلى ذلك، يساعدك (Cyber Pass) على إنشاء كلمات سر قوية وفريدة لكل حساب، وفحص قوتها مما يعزز أمان حساباتك عبر الإنترنت ويقلل من خطر الاختراق.

1.5 نطاق المشروع والقيود

1.5.1 نطاق المشروع

وبما أن المشروع سيكون مفتوح المصدر وذو واجهة رسومية سهلة، فإنه سيكون متاحا على نطاق واسع للجميع، سواء الباحثين أو أي شخص ليس لديه خبرة في هذا المجال.

1.5.2 قيود المشروع

- تعقيد التشفير: تحقيق توازن بين قوة التشفير وسهولة الاستخدام.
- ثقة المستخدم: كسب ثقة المستخدم في أمان البرنامج وحماية بياناته.
- الخصوصية: الامتثال لقوانين حماية البيانات والخصوصية.
- حقوق الملكية الفكرية: حماية حقوق الملكية الفكرية للبرنامج.
- المنافسة: التميز في سوق مليء بالبرامج المنافسة.
- دعم العملاء: توفير دعم فني للعملاء على مدار الساعة.

1.6 منهجية المشروع

يهدف مشروع إدارة كلمات السر إلى تطوير نظام آمن وسهل الاستخدام لحفظ وتنظيم كلمات السر للمستخدمين، مما يعزز أمن البيانات الشخصية ويقلل من الجهد المبذول في تذكرها. يتمثل الهدف الرئيسي في بناء قاعدة بيانات مشفرة لتخزين كلمات السر، وتصميم واجهة مستخدم سهله، وتوفير ميزات إضافية مثل توليد كلمات سر قوية ومراقبة تسرب البيانات.

سيتم ضمان أمان البيانات وحماية خصوصية المستخدمين كما سيتم التركيز على تصميم تجربة مستخدم سلسة وممتعة.

1.7 مراحل تنفيذ المشروع

1.7.1 مرحلة التخطيط

- تحديد المتطلبات التقنية والفنية.
- اختيار التقنيات المناسبة (قواعد البيانات، لغات البرمجة، خوارزميات التشفير).
- تصميم واجهة المستخدم.
- وضع خطة تطوير.

1.7.2 مرحلة التطوير والتحليل

- اختيار اللغة والتقنيات: تحديد لغات البرمجة وأدوات التشفير المناسبة (مثل Python، Cryptography، SQLite).
- بناء قاعدة البيانات لتخزين كلمات السر المشفرة.
- تطوير واجهة المستخدم (تطبيق سطح مكتب).
- تنفيذ خوارزميات التشفير (Fernet - PBKDF2 - SHA-256)
- دمج ميزات إضافية (توليد كلمات سر، تقييم قوة كلمة السر).

1.7.3 مرحلة الاختبار

- اختبار الأداء والأمان.
- اختبار واجهة المستخدم.
- تصحيح الأخطاء.

1.7.4 مرحلة الإطلاق

- إطلاق النسخة الأولى من المنتج.
- التسويق والترويج.

1.7.5 مرحلة الصيانة والتطوير

- تقديم الدعم الفني للمستخدمين.
- إطلاق تحديثات جديدة وإضافة ميزات جديدة.

1.7.6 الارشفة والتوثيق

- ارشفة البيانات والاحتفاظ بها
- توثيق كامل للنظام يشمل التعليمات الخاصة بالاستخدام والصيانة، بالإضافة إلى توضيح جميع جوانب الأمان المعتمدة في النظام.
- تقديم التقرير النهائي الذي يوضح جميع مراحل المشروع، والتحديات التي واجهتها، والحلول المقدمة

1.8 نموذج دورة الحياة

تتضمن المراحل الأساسية للمشروع (التخطيط، التطوير والتحليل، والاختبار، والإطلاق، والصيانة، والارشفة والتوثيق) والتطوير من خلال اختيار التقنيات المناسبة وتطبيق خوارزميات التشفير القوية.

(1.1) مخطط لتنفيذ مراحل المشروع

1.9 الفترة الزمنية للمشروع

المرحلة	المدة الزمنية(بالشهر)
التحديد والاختيار	شهر
التخطيط والبدء بالمشروع	شهر
التحليل	ثلاث أشهر
التصميم	شهر
التنفيذ والاختبار	أربعة أشهر
التوثيق	عشرة أشهر

جدول (1.2) مخطط جانت لتنفيذ مراحل المشروع

المرحلة	الفترة الزمنية بالشهر									
التحديد والاختيار										
التخطيط والبدء بالمشروع										
التحليل										
التصميم										
التنفيذ والاختبار										
التوثيق										

جدول (1.3) مخطط جانت لتنفيذ مراحل المشروع بشكل أكثر تفصيلا

1.10 مستخدمي المشروع

- الأفراد: المستخدمون العاديون الذين يرغبون في حماية كلمات المرور.
- الشركات: المؤسسات التي تحتاج إلى حل آمن لإدارة كلمات سر موظفيها.
- المطورون: المبرمجون الذين يرغبون في دمج حل إدارة كلمات السر في تطبيقاتهم.

1.11 معوقات المشروع

1. الأمان والتشفير:

حماية كلمات السر المخزنة يُعد من أهم التحديات، حيث يجب استخدام خوارزميات تشفير قوية وتطبيق إجراءات أمان متقدمة لحماية البيانات من الاختراق. أي ثغرة أمنية في النظام قد تعرض بيانات المستخدمين للخطر.

2. نسيان كلمة المرور الرئيسية:

يعتمد النظام عادةً على "كلمة مرور رئيسية" للوصول إلى باقي كلمات المرور، وفي حال نسيان المستخدم لكلمة المرور الرئيسية، قد يُفقد الوصول إلى جميع البيانات المخزنة، خاصة إذا كانت البيانات مشفرة بالكامل ولا يمكن استعادتها.

3. هجمات الهندسة الاجتماعية:

يمكن أن يكون المستخدم عرضة لهجمات الهندسة الاجتماعية، التي تعتمد على خداع المستخدم للكشف عن كلمة المرور الرئيسية أو تقديم معلومات حساسة. هذه الهجمات تُعتبر تحدياً كبيراً لأنها تستهدف العنصر البشري وليس النظام نفسه.

4. التكامل مع المتصفحات والتطبيقات:

قد يكون من الصعب ضمان عمل مدير كلمات السر بكفاءة مع جميع المتصفحات والتطبيقات المختلفة، خاصة مع التحديثات المستمرة في البرمجيات والتطبيقات، التي قد تؤثر على التوافق والأداء.

5. مخاوف الخصوصية والثقة:

يُعتبر مدير كلمات السر حلاً حساسًا بسبب المعلومات الشخصية التي يحتفظ بها، ولذلك قد يواجه المستخدمون مشكلة في الثقة بالنظام أو الشركة المطورة له. أي تهاون في معايير الخصوصية قد يؤثر على سمعة النظام وقدرة المستخدمين على الوثوق به.

6. الاستجابة للهجمات المتقدمة:

يجب على النظام أن يكون قادرًا على التعامل مع الهجمات المتقدمة والمتطورة، مثل هجمات القوة الغاشمة (brute force) والهجمات العشوائية لاختبار كلمات المرور، مما يتطلب تحديثات دورية لتعزيز الأمن.

7. استهلاك موارد الجهاز:

قد يستهلك مدير كلمات السر موارد كبيرة من المعالج والذاكرة، خصوصاً في حالة إدارة عدد كبير من الحسابات والمزامنة المستمرة مع السحابة، مما قد يؤثر سلباً على أداء الجهاز.

8. التحديثات المستمرة:

مع تطور تقنيات القرصنة، قد يحتاج النظام إلى تحديثات أمنية متواصلة، ما قد يشكل تحدياً للفرق التطويرية في الحفاظ على مستوى عالٍ من الأمان بشكل دوري.

9. صعوبة تهيئة وتثقيف المستخدم:

قد يواجه المستخدمون صعوبة في فهم كيفية استخدام مدير كلمات السر، خاصة إذا كانت لديهم خلفية تقنية محدودة، مما يتطلب مواد تدريبية وتثقيفية لتوضيح فوائد وكيفية استخدام النظام بأمان.

1.12 تنظيم المشروع

❖ **الفصل الأول الإطار العام**

المقدمة

يهدف مشروع Cyber Pass إلى تطوير تطبيق آمن وسهل الاستخدام لإدارة كلمات السر وتوليد كلمات مرور قوية، مع تخزينها بشكل مشفر. المشروع موجه للأفراد والشركات والمطورين، ويركز على تعزيز أمن المعلومات الشخصية وتقليل الاعتماد على الذاكرة البشرية.

بيان المشكلة:

تواجه إدارة كلمات السر عدة تحديات، أبرزها: التشفير، التخزين، التوافق، كسب ثقة المستخدم، التوعية، والتميز عن المنافسين.

أهداف المشروع:

- تأمين كلمات المرور بالتشفير.
- منع إعادة الاستخدام.
- توليد كلمات مرور قوية.
- توعية المستخدمين بأمن المعلومات.
- تقديم واجهة استخدام بسيطة وآمنة.

تعريف المشروع:

تطبيق مكتوب بلغةPython ، يُخزن كلمات المرور خلف كلمة مرور رئيسية مشفرة، ويدعم إنشاء وفحص قوة كلمات السر.

نطاق المشروع:

تطبيق مفتوح المصدر متاح للجميع بواجهة رسومية سهلة الاستخدام.

القيود:

تحديات في التشفير، الخصوصية، الثقة، دعم المستخدمين، والامتثال للقوانين.

منهجية العمل:

اتباع مراحل تطوير برمجية تشمل التخطيط، التحليل، التصميم، التنفيذ، الاختبار، التوثيق، والتحديث المستمر.

مراحل تنفيذ المشروع:

1. مرحلة التخطيط: تحديد المتطلبات التقنية، اختيار الأدوات المناسبة، تصميم الواجهة، ووضع خطة تطوير.
2. مرحلة التطوير والتحليل: بناء قاعدة البيانات، تطوير الواجهة، تنفيذ التشفير، وإضافة الميزات (توليد كلمات مرور وفحصها).
3. مرحلة الاختبار: اختبار الأمان والأداء، وتجربة واجهة المستخدم.
4. مرحلة الإطلاق: إصدار النسخة الأولى وتسويقها.
5. مرحلة الصيانة والتطوير: توفير الدعم الفني، وتنفيذ التحديثات.
6. مرحلة التوثيق والأرشفة: توثيق النظام وتعليمات الاستخدام، وتقديم تقرير نهائي شامل.

نموذج دورة الحياة:

يعتمد المشروع على دورة حياة تقليدية تشمل المراحل الست السابقة، مع التركيز على الأمان والتقنيات الحديثة للتشفير.

الفترة الزمنية للمشروع:

موزعة على عدة أشهر بحسب الجدول الزمني (كما في مخطط جانت المرفق).

نطاق المشروع (توسيع الاستخدام):

يمكن لأي شخص الاستفادة من التطبيق بفضل بساطته، وهو متاح على نطاق واسع بدون حاجة لخبرة تقنية.

- **الفصل الثاني: الخلفية ومراجعة الادبيات**
- **الفصل الثالث: تحليل متطلبات النظام**
- **الفصل الرابع: النظام المقترح**
- **الفصل الخامس: التنفيذ**
- **الفصل السادس: الاستنتاجات والتوصيات**
- **الفصل السابع: الـخـاتـمـة**

الـــفـــــصـــل الـــــثـــانـــــي

❖ الفصل الثاني: الخلفية ومراجعة الادبيات

2.1 المقدمة

تعد كلمات السر، واحدة من أكثر وسائل الأمان استخدامًا لحماية الحسابات الرقمية والتطبيقات عبر الإنترنت ومع تزايد عدد الحسابات التي يتطلبها المستخدمون في حياتهم اليومية (مثل البريد الإلكتروني، الشبكات الاجتماعية، الحسابات البنكية، وغيرها)، أصبح من الصعب على المستخدمين تذكر كلمات السر المعقدة والطويلة. وهذا أدى إلى ظهور أدوات تُسمى "إدارة كلمات السر" (Cyber Pass) التي توفر وسيلة آمنة وفعالة لتخزين وتنظيم كلمات السر.

2.2 الخلفية النظرية

- أهمية كلمات السر: كلمات السر هي وسيلة أساسية لتحقيق الأمان السيبراني، حيث تقوم بحماية الوصول إلى المعلومات الحساسة.
- التحديات: مع زيادة عدد كلمات السر التي يستخدمها الأفراد، يزداد خطر نسيانها أو تعرضها للخطر.
- الهدف من إدارة كلمات السر: تهدف إدارة كلمات السر إلى توفير أمان عالي لكلمات السر، وتسهيل استخدامها للأفراد.

2.3 نظرة عامه حول إدارة كلمات السر

2.3.1 مفاهيم أساسية

- تخزين كلمات السر: استخدام تقنيات تشفير قوية لحماية كلمات السر.
- توليد كلمات السر: إنشاء كلمات سر قوية ومميزة.
- تحديث كلمات السر: تحديث كلمات السر بانتظام لتعزيز الأمان.
- مصادقة متعددة العوامل: استخدام عدة طرق لتحقق هوية المستخدم.
- تحليل: بمعني يعطيك تقييم لمدى قوة كلمة السر **(قوية-متوسطة-ضعيفة)**

2.3.2 أنواع أنظمة إدارة كلمات السر

- أنظمة إدارة كلمات السر المحلية: تخزن كلمات السر على جهاز المستخدم.
- أنظمة إدارة كلمات السر السحابية: تخزن كلمات السر على السحابة.
- أنظمة إدارة كلمات السر الهجينة: تجمع بين التخزين المحلية والسحابية.

2.3.3 تقنيات تشفير شائعة

- تشفير AES: تقنية تشفير قوية تستخدم في العديد من أنظمة إدارة كلمات السر.
- تشفير SHA: تقنية تشفير تستخدم لتحمي كلمات السر
- تشفير PBKDF2: تقنية تشفير تستخدم لتحمي كلمات السر.

2.3.4 التهديدات والتحديات

1. هجمات القوة الغاشمة: محاولات كسر كلمات السر باستخدام القوة الغاشمة.
2. هجمات الفدية: محاولات سرقة كلمات السر.
3. هجمات الفيروسات: محاولات تثبيت فيروسات لسرقة كلمات السر.
4. هجمات التصيد: محاولات سرقة كلمات السر عن طريق التصيد.

2.4 الأنظمة المشابهة

- **الأنظمة السحابية:**

LastPass

المميزات:

1. تخزين آمن لكلمات السر.
2. توليد كلمات سر قوية.
3. مصادقة متعددة العوامل.
4. تحديث كلمات السر التلقائي.
5. دعم متعددة الأجهزة.
6. واجهة مستخدم سهلة.

العيوب:

1. تكلفة الاشتراك الشهري (2.25 دولار).
2. مشاكل في الأداء أحيانًا.
3. لا يدعم التخزين المحلي.

Password1

المميزات:

1. تخزين آمن لكلمات السر.
2. توليد كلمات سر قوية.
3. مصادقة متعددة العوامل.
4. تحديث كلمات السر التلقائي.
5. دعم متعددة الأجهزة.
6. واجهة مستخدم سهلة.

العيوب:

1. تكلفة الاشتراك الشهري (2.99 دولار).
2. لا يدعم التخزين المحلي.
3. مشاكل في الأداء أحيانًا.

- **الأنظمة المحلية:**

Bitwarden

الميزات:

1. مجاني ومفتوح المصدر.
2. تخزين آمن لكلمات السر.
3. توليد كلمات سر قوية.
4. دعم التخزين المحلي.
5. يدعم مصادقة متعددة العوامل.
6. واجهة مستخدم بسيطة.

العيوب:

1. لا يدعم التكامل مع التطبيقات الشائعة.
2. واجهة مستخدم غير جذابة.
3. مشاكل في الأداء أحيانًا.

- **الأنظمة الهجينة:**

Kaspersky Password Manager #

الميزات:

1. تخزين آمن لكلمات السر.
2. توليد كلمات سر قوية.
3. مصادقة متعددة العوامل.
4. تحديث كلمات السر التلقائي.
5. دعم متعددة الأجهزة.
6. التكامل مع التطبيقات الشائعة.
7. واجهة مستخدم سهلة.

العيوب:

1. تكلفة الاشتراك الشهري (2.25 دولار).
2. لا يدعم التخزين المحلي.
3. مشاكل في الأداء أحيانًا.

RoboForm #

الميزات:

1. تخزين آمن لكلمات السر.
2. توليد كلمات سر قوية.
3. مصادقة متعددة العوامل.
4. تحديث كلمات السر التلقائي.
5. دعم متعددة الأجهزة.
6. التكامل مع التطبيقات الشائعة.

العيوب:

1. تكلفة الاشتراك الشهري (1.99 دولار).
2. لا يدعم التخزين المحلي.
3. مشاكل في الأداء أحيانًا.

CyberArk

الميزات:

1. تخزين آمن لكلمات السر.
2. توليد كلمات سر قوية.
3. مصادقة متعددة العوامل.
4. تحديث كلمات السر التلقائي.
5. دعم متعددة الأجهزة.
6. التكامل مع التطبيقات الشائعة.
7. واجهة مستخدم سهلة.
8. حلول أمان متقدمة.

العيوب:

1. تكلفة الاشتراك العالية.
2. معقدة ومتطلبة خبرة فنية.
3. لا تناسب المستخدمين العاديين.

- **الأنظمة المخصصة للأعمال**

Okta #

المميزات:

1. إدارة هوية وهوية.
2. مصادقة متعددة العوامل.
3. تحديث كلمات السر التلقائي.
4. دعم متعددة الأجهزة.
5. التكامل مع التطبيقات الشائعة.
6. واجهة مستخدم سهلة.
7. حلول أمان متقدمة.

العيوب:

1. تكلفة الاشتراك الشهري (2 دولار).
2. معقدة ومتطلبة خبرة فنية.
3. لا تناسب المستخدمين العاديين.

جدول (1.4) جدول مقارنة بين باقي الأنظمة والسابقة والنظام الحالي

اسم التطبيق (Application Name)	النوع/البيئة (Type/Environment)	أهم المميزات (Key Features)	أهم العيوب (Key Disadvantages)
LastPass	سحابي (Cloud)	تخزين آمن، توليد كلمات سر، مصادقة متعددة، دعم متعدد الأجهزة، واجهة سهلة.	تكلفة اشتراك، لا يدعم التخزين المحلي.
Password1	سحابي (Cloud)	تخزين آمن، توليد كلمات سر، مصادقة متعددة، دعم متعدد الأجهزة، واجهة سهلة.	تكلفة اشتراك أعلى قليلاً، لا يدعم التخزين المحلي.
Bitwarden	محلي/سحابي (Local/Cloud)	مجاني ومفتوح المصدر، تخزين محلي، يدعم مصادقة متعددة العوامل.	لا يدعم التكامل، واجهة بسيطة.
Kaspersky Password Manager	هجين (Hybrid)	تخزين آمن، توليد، مصادقة متعددة، دعم متعدد الأجهزة، تكامل وتحديث تلقائي.	تكلفة اشتراك، لا يدعم التخزين المحلي بشكل أساسي.
RoboForm	هجين (Hybrid)	تخزين آمن، توليد، مصادقة متعددة، دعم متعدد الأجهزة، تكامل وتحديث تلقائي.	تكلفة اشتراك منخفضة نسبيًا، لا يدعم التخزين المحلي بشكل أساسي.
CyberArk	هجين/أعمال (Hybrid/Business)	حلول أمان متقدمة للشركات، دعم شامل للميزات الأمنية.	تكلفة عالية جدًا، معقد للاستخدام العادي.
Okta	أعمال (Business)	إدارة هوية متقدمة، تكامل واسع، أمان قوي.	تكلفة، معقد، غير مناسب للمستخدم العادي.
Cyber Pass (النظام المقترح)	محلي (Local)	مجاني ومفتوح المصدر، مصادقة متعددة، توليد كلمات مرور قوية، فحص وتقييم قوه كلمه المرور	لا يدعم التخزين الخارجي لا يدعم التكامل(حاليا)

الـــفـــــصـــل الـــثـــالث

❖ الفصل الثالث: تحليل متطلبات النظام

3.1 المقدمة

تحليل متطلبات **(Cyber Pass):** الأساس لبناء نظام آمن وفعال...

مرحباً بك في عالم إدارة كلمات المرور! مع تزايد الاعتماد على التطبيقات والخدمات الرقمية، أصبحت حماية كلمات المرور أمراً بالغ الأهمية هنا يأتي دور **"Cyber Pass "**، وهو تطبيق مصمم لتخزين وتنظيم كلمات المرور بشكل آمن، مما يسهل على المستخدمين إدارة حساباتهم المتعددة دون الحاجة لتذكر كلمات مرور معقدة.

لماذا نحتاج إلى تحليل متطلبات؟

قبل البدء في تطوير أي تطبيق، من الضروري إجراء تحليل دقيق لجميع المتطلبات. هذا التحليل يضمن أن التطبيق النهائي يلبي احتياجات المستخدمين بشكل كامل، ويعالج التحديات الأمنية، ويقدم تجربة استخدام سلسة

3.2 أهداف مشروع تحليل المتطلبات

- فهم متطلبات المستخدم: تحديد الميزات الأساسية التي يتوقعها المستخدمون من النظام، مثل:
  - تخزين كلمات مرور آمنة.
  - توليد كلمات مرور قوية.
  - ملء النماذج تلقائياً.
  - مشاركة كلمات المرور بشكل آمن.
  - دعم منصات متعددة.
- تحديد المتطلبات التقنية: تحديد التقنيات والأدوات اللازمة لتطوير التطبيق، مثل:
  - قاعدة بيانات آمنة لتخزين كلمات المرور.
  - خوارزميات تشفير قوية.
  - واجهة مستخدم سهلة الاستخدام.
  - تكامل مع المتصفحات.
- تقييم المخاطر الأمنية: تحديد المخاطر الأمنية المحتملة وتطوير استراتيجيات للحد منها، مثل:
  - حماية ضد الهجمات السيبرانية.
  - إدارة الأذونات والوصول.

- وضع خطة تطوير: وضع خطة زمنية وميزانية واضحة لتطوير التطبيق.

3.3 المتطلبات

3.3.1 المتطلبات الوظيفية Functional Requirements

- تسجيل الدخول والأمان: مصادقة المستخدمين وتحقق من الهوية.
- تخزين كلمات السر: تخزين آمن لكلمات السر والبيانات الحساسة.
- توليد كلمات سر: توليد كلمات سر قوية ومتعددة الحروف.
- إدارة كلمات السر: تحديث، حذف، ونسخ كلمات السر.
- بحث وتصفية: بحث عن كلمات سر محددة.
- تكرار كلمات السر: تحذير من تكرار كلمات السر.
- تحليل أمان: تحليل أمان كلمات السر.
- تذكير كلمات السر: تذكير المستخدمين بتغيير كلمات السر.
- دعم متعددة الأجهزة: الوصول إلى كلمات السر من الأجهزة المختلفة.
- التكامل مع التطبيقات: التكامل مع التطبيقات الشائعة.

3.3.2 المتطلبات غير الوظيفية Non-Functional Requirements

- الأمان: تشفير قوي لكلمات السر.
- السرعة: أداء سريع في البحث والوصول إلى كلمات السر.
- الاستقرار: مقاومة للعطل والخطأ.
- الوضوح: واجهة مستخدم سهلة الاستخدام.
- التعاون: دعم التعاون الفوري.
- التكامل: التكامل مع أنظمة التشغيل المختلفة.
- التوافق: التوافق مع المعايير الأمنية الدولية.
- الدعم الفني: دعم فني جيد.
- التحديثات: تحديثات دورية لتحسين الأمان.

3.3.3 المتطلبات الفنية

- لغة البرمجة: Java، Python، أو C++.
- قاعدة البيانات: MySQL، MongoDB، أو PostgreSQL.
- إطار العمل: Spring، Django، أو React.
- نظام التشغيل: Windows، Linux، أو macOS.
- مكتبات الأمان: OpenSSL، SSL/TLS.

3.3.4 المتطلبات القانونية

- الامتثال للقوانين: الامتثال لقوانين الخصوصية والأمان.
- شروط الاستخدام: تحديد شروط الاستخدام.
- سياسة الخصوصية: تحديد سياسة الخصوصية.
- حماية حقوق النشر: حماية حقوق النشر.

3.3.5 المتطلبات المالية

- التكلفة الإجمالية: تقدير التكلفة الإجمالية.
- التكلفة الشهرية: تقدير التكلفة الشهرية.
- ميزانية التطوير: تحديد ميزانية التطوير.
- ميزانية الصيانة: تحديد ميزانية الصيانة.

3.4 الجدوى

قمنا بدراسة الجدوى لمعرفة ما إذا كان النظام مجدي من جميع النواحي التي قد تؤثر على النظام وتقييم مدى جدوى هذا المشروع وقدرته على النجاح وتحقيق الأهداف المرجوة ...

الجدوى هي تقييم القيمة أو الفائدة من مشروع أو استثمار أو فكرة تجارية

؛ الجوانب الذي قمنا بدراستها:

3.4.1 الجدوى التقنية

- الأمان: تشفير قوي (AES، RSA)، مصادقة متعددة العوامل.
- القدرة على التنفيذ: إمكانية تنفيذ المشروع باستخدام تقنيات حديثة (Java، Python، C++).
- الابتكار: توفير حلول مبتكرة لتحليل كلمات السر وتقديم توصيات.
- لجودة: جودة النظام فيما يتعلق بالأداء والاستقرار.
- التكامل: تكامل النظام مع الأنظمة الأخرى (Android، iOS، Windows، macOS)..
القدرة على التوسع: إمكانية توسيع النظام ليشمل ميزات جديدة.
- السرعة: سرعة الوصول إلى كلمات السر والبحث السريع.

3.4.1.1 تجهيزات الهاردوير (Hardware)

1. المعالج (CPU): (Intel Core i7 أو AMD)
2. الذاكرة العشوائية (RAM): 8 جيجابايت على الأقل.
3. التخزين: 256 جيجابايت من مساحة التخزين الصلبة (SSD) أو أكثر.
4. شاشة: شاشة بدقة p1080 أو أعلى.

3.4.1.2 تجهيزات السوفت وير (Software)

1. نظام التشغيل: Windows 10/11، macOS، Linux (Ubuntu).
2. لغة البرمجة: Java، Python، C++، أو JavaScript.
3. إطار العمل: Spring، Django، React، أو Angular.
4. قاعدة البيانات: MySQL، MongoDB، Microsoft SQL Server.
5. مكتبة الأمان: OpenSSL، SSL/TLS، أو AES.

3.4.1.3 تجهيزات الشبكة (Network)

1. السرعة: سرعة اتصال إنترنت 10 ميجابايت/ثانية أو أعلى.
2. البروتوكول: TCP/IP، HTTP/HTTPS، أو SSH.
3. النطاق الترددي: 1 جيجابايت/ثانية أو أعلى.
4. الخوادم: خادم ويب (Apache، Nginx، أو IIS).
5. النظام الأساسي: IPv4 أو IPv6.

3.4.2 الجدوى الاقتصادية

- توفير التكاليف: تقليل تكاليف إدارة كلمات السر اليدوية.
- زيادة الإنتاجية: توفير الوقت والمجهود في إدارة كلمات السر.
- تحسين الأمان: حماية البيانات الحساسة من الاختراقات.
- زيادة الربح: تحقيق إيرادات من اشتراكات المستخدمين.
- توفير فرص العمل: توفير فرص عمل في مجال التكنولوجيا.

3.4.3 الجدوى التشغيلية

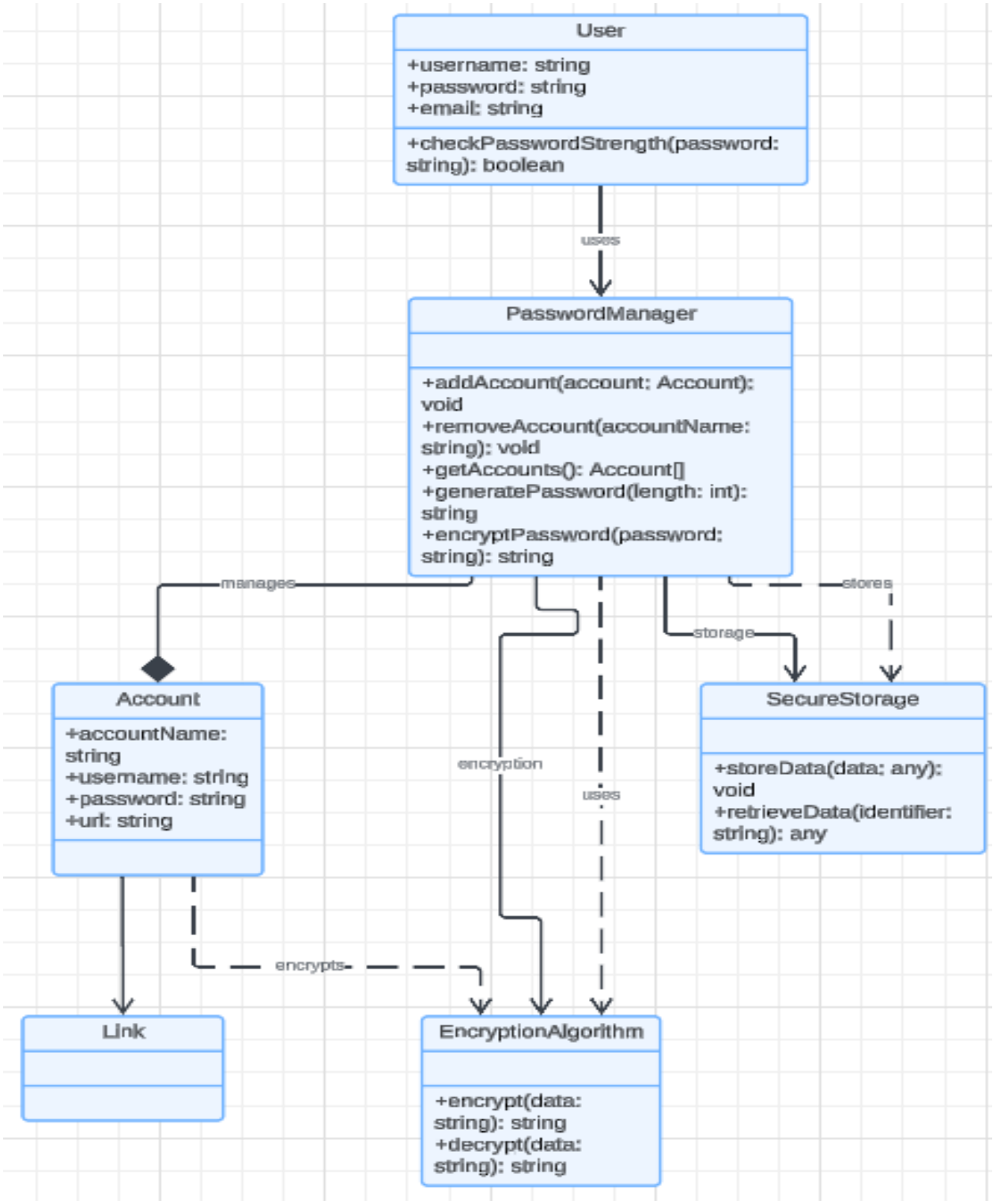
- السرعة والكفاءة: سرعة الوصول إلى كلمات السر والعمليات.
- الاستقرار والموثوقية: مقاومة للعطل والخطأ.
- الأمان والخصوصية: حماية البيانات الحساسة.
- السهولة في الاستخدام: واجهة مستخدم سهلة الاستخدام.
- التكامل مع الأنظمة الأخرى: التكامل مع التطبيقات والأنظمة الشائعة.
- الدعم الفني: دعم فني جيد ومستمر.
- التحديثات والصيانة: تحديثات دورية وصيانة النظام.

3.5 السيناريو:

في عالم رقمي مليء بالحسابات والمواقع الإلكترونية، أصبح تذكر عشرات كلمات المرور المعقدة مهمة شبه مستحيلة. هنا يأتي دور تطبيق إدارة كلمات المرور لينقذنا من هذا العناء يوجد اشخاص يعانون من مشاكل نسيان كلمات المرور وحساباتهم باستمرار ف يضطرون إلى إعادة تعيينها مرارًا وتكرارً وهذا يضيع وقتهم ويسبب لهم إزعاجًا كبيرً بعد تحميل التطبيق، قرر شخص بإنشاء حساب جديد وقام بتعيين كلمة مرور رئيسية قوية، وهي المفتاح الذي سيفتح له عالمه الجديد الآمن. وبدأ بإضافة كلمات المرور الخاصة به واحدة تلو الأخرى، من حساب البريد الإلكتروني إلى حسابات وسائل التواصل الاجتماعي، وحتى حسابات البنوك مع مرور الوقت، أصبح يعتمد بشكل كبير على تطبيق إدارة كلمات المرور. لم يعد يقلق بشأن نسيان كلمات المرور، وأصبح يشعر بالأمان التام. حتى أنه بدأ في مشاركة التطبيق مع أصدقائه وعائلته، لينقذهم أيضًا من مشكلة كلمات المرور.

ف تطبيق إدارة كلمات المرور هو أداة قوية وضرورية في عالمنا الرقمي فهو يساعدنا على حماية خصوصيتنا وأمان بياناتنا، ويجعل حياتنا أسهل وأكثر تنظيمًا

(ER Diagram) 3.7



(ER Diagram) :(3.1) شكل

الكيانات (الفئات) الرئيسية:

- **User (المستخدم):**

لديه خصائص (attributes): username (اسم المستخدم)، password (كلمة المرور)، email (البريد الإلكتروني) - كلها من نوع string (نص).

لديه عملية (operation): checkPasswordStrength (password: string): Boolean - تتحقق من قوة كلمة المرور، وتُرجع true إذا كانت قوية وfalse بخلاف ذلك.

- لديه عمليات:
  - add Account (account: Account): void: لإضافة حساب جديد.
  - remove Account (account Name: string): void: لإزالة حساب بناءً على اسمه.
  - get Accounts (): Account []: لاسترجاع قائمة بجميع الحسابات (مصفوفة من كائنات Account).
  - generate Password (length: int): string: لتوليد كلمة مرور عشوائية بطول محدد.
  - encrypt Password (password: string): string: لتشفير كلمة مرور معينة.

- **Account (الحساب):**

يمثل حسابًا فرديًا يتم إدارته بواسطة Password Manager.

لديه خصائص: account Name (اسم الحساب), username (اسم المستخدم لهذا الحساب), password (كلمة المرور لهذا الحساب), URL (عنوان URL المرتبط بالحساب) - كلها من نوع string.

Secure Storage (التخزين الآمن)

- لديه عمليات:
  - store Data(data: any): void: لتخزين أي نوع من البيانات.
  - retrieve Data(identifier: string): any: لاسترجاع البيانات باستخدام معرف.
  - Encryption Algorithm (خوارزمية التشفير).

يمثل فئة لتوفير وظائف التشفير وفك التشفير.

- لديه عمليات:
- encrypt(data: string): string: لتشفير البيانات.
- decrypt(data: string): string: لفك تشفير البيانات.

Link (الرابط):

يمثل رابطًا خارجيًا أو معلومة إضافية للحساب.

- العلاقات بين الكيانات:

User Uses Password Manager (المستخدم يستخدم مدير كلمات المرور):

خط صلب مع سهم يشير من User إلى Password Manager. هذا يعني أن User يتفاعل مع Password Manager لأداء مهام إدارة كلمات المرور.

Password Manager Manages Account (مدير كلمات المرور يدير الحساب):

خط صلب من Password Manager إلى Account مع كلمة "manages" بجانب العلاقة. هذا يشير إلى أن Password Manager مسؤول عن إنشاء، تحديث، وحذف كائنات Account.

Stores Secure Storage-Password Manager (مدير كلمات المرور يخزن في التخزين الآمن):

خط صلب من Password Manager إلى Secure Storage مع كلمة "stores" بجانب العلاقة. هذا يعني أن Password Manager يستخدم Secure Storage لحفظ البيانات.

Secure Storage Stores Encryption Algorithm (التخزين الآمن يستخدم خوارزمية التشفير):

خط صلب من Secure Storage إلى Encryption Algorithm مع كلمة "stores" بجانب العلاقة. هذا يوحي بأن Secure Storage قد يعتمد على Encryption Algorithm لتأمين البيانات المخزنة.

Password Manager Uses Encryption Algorithm (مدير كلمات المرور يستخدم خوارزمية التشفير):

خط متقطع (dashed line) من Password Manager إلى Encryption Algorithm مع كلمة "uses" بجانب العلاقة. هذا يشير إلى علاقة اعتماد (dependency)، حيث يعتمد Password Manager على Encryption Algorithm لأداء وظائف التشفير (مثل encrypt Password).

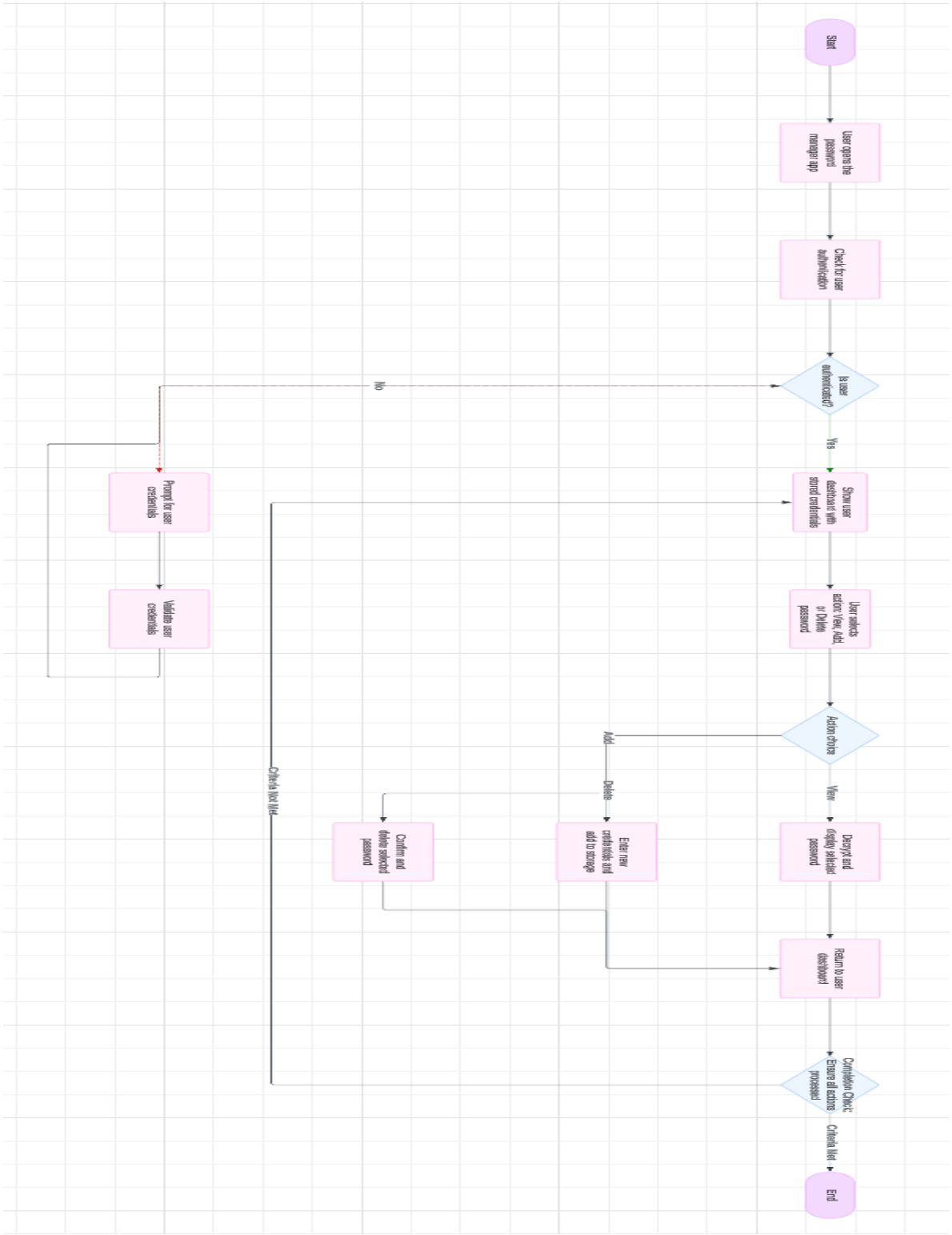
Account Encrypts with Encryption Algorithm (الحساب يُشفر باستخدام خوارزمية التشفير):

خط متقطع (dashed line) من Account إلى Encryption Algorithm مع كلمة "encrypts" بجانب العلاقة. هذا يعني أن Account قد يستخدم (أو يتطلب) Encryption Algorithm لتشفير بعض بياناته (خاصة كلمة المرور).

Account Links to Link (الحساب يرتبط بـ Link):

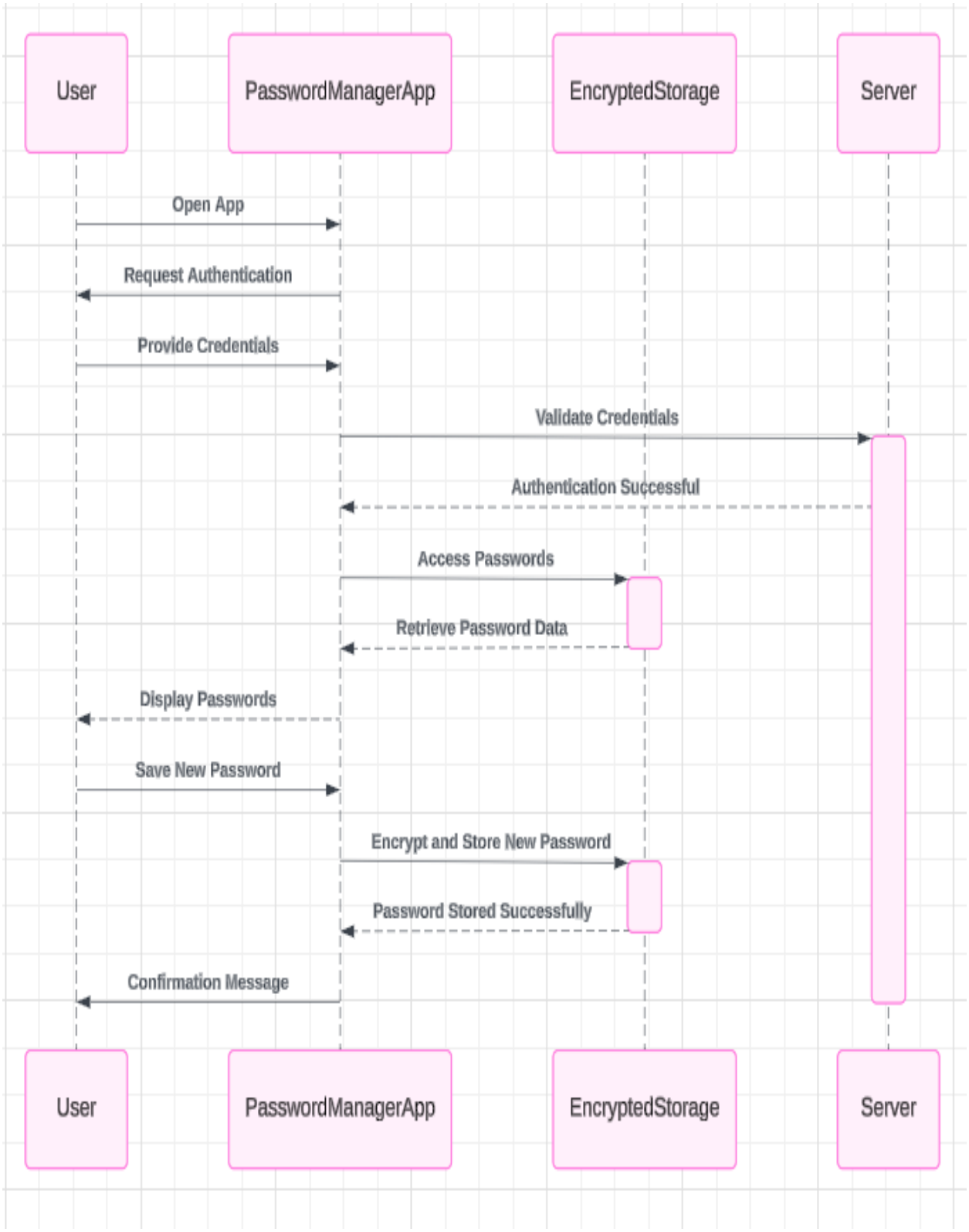
خط صلب من Account إلى Link. هذا يشير إلى أن Account يمكن أن يكون له علاقة مع Link (ربما يحتوي على رابط أو عدة روابط).

Data Flow Diagram (DFD) 3.8



Data Flow Diagram (DFD) :(3.2) شكل

Sequence Diagram 3.9



Sequence Diagram :(3.3) شكل

الصوره اعلاة هي رسم بياني لتسلسل الأحداث (Sequence Diagram) فيUML ، والذي يوضح التفاعل الزمني بين الكائنات أو المكونات في نظام معين. هذا الرسم يصف كيفية تفاعل المستخدم مع تطبيق إدارة كلمات المرور.

خطوات التسلسل:

1. **المستخدم (User) يفتح التطبيق:(PasswordManagerApp)**
   - يبدأ التفاعل بإرسال المستخدم رسالة "Open App" (فتح التطبيق) إلى تطبيق إدارة كلمات المرور.
2. **طلب المصادقة:(Request Authentication)**
   - بعد فتح التطبيق، يرسل تطبيق إدارة كلمات المرور رسالة "Request Authentication"(طلب المصادقة) إلى المستخدم، مما يعني أنه يطلب بيانات الاعتماد للمتابعة.
3. **المستخدم يوفر بيانات الاعتماد:(Provide Credentials)**
   - يرسل المستخدم رسالة "Provide Credentials" (توفير بيانات الاعتماد) إلى تطبيق إدارة كلمات المرور، والتي تتضمن اسم المستخدم وكلمة المرور الخاصة به.
4. **التحقق من صحة بيانات الاعتماد:(Validate Credentials)**
   - يقوم تطبيق إدارة كلمات المرور بإرسال رسالة "Validate Credentials" (التحقق من صحة بيانات الاعتماد) إلى **الخادم .(Server)** هذا يشير إلى أن عملية المصادقة تتم على الخادم (ربما للتحقق من قاعدة بيانات المستخدمين).
5. **المصادقة ناجحة:(Authentication Successful)**
   - يرسل الخادم رسالة "Authentication Successful" (المصادقة ناجحة) إلى تطبيق إدارة كلمات المرور، مؤكداً أن بيانات الاعتماد صحيحة. يلاحظ هنا أن الخادم لديه خط حياة طويل يغطي معظم التفاعل، مما يشير إلى دوره المستمر في المصادقة والعمليات الأخرى.
6. **الوصول إلى كلمات المرور:(Access Passwords)**

- بعد المصادقة الناجحة، يرسل تطبيق إدارة كلمات المرور رسالة Access" "Passwords(الوصول إلى كلمات المرور) إلى **التخزين المشفر (EncryptedStorage)**.هذا يعني أنه يطلب الوصول إلى البيانات المخزنة.

7. **استرجاع بيانات كلمات المرور:(Retrieve Password Data)**

- يقوم التخزين المشفر بإرسال رسالة "Retrieve Password Data" (استرجاع بيانات كلمات المرور) إلى تطبيق إدارة كلمات المرور. هذا يتضمن جلب كلمات المرور المشفرة.

8. **عرض كلمات المرور:(Display Passwords)**

- يرسل تطبيق إدارة كلمات المرور رسالة "Display Passwords" (عرض كلمات المرور) إلى المستخدم، مما يسمح للمستخدم برؤية كلمات المرور الخاصة به. (يفترض هنا أن التطبيق قام بفك تشفيرها قبل العرض).

9. **حفظ كلمة مرور جديدة:(Save New Password)**

- يرسل المستخدم رسالة "Save New Password" (حفظ كلمة مرور جديدة) إلى تطبيق إدارة كلمات المرور، مما يشير إلى رغبته في إضافة أو تحديث كلمة مرور.

10. **تشفير وتخزين كلمة المرور الجديدة:(Encrypt and Store New Password)**

- يقوم تطبيق إدارة كلمات المرور بإرسال رسالة Encrypt and Store New" "Password(تشفير وتخزين كلمة المرور الجديدة) إلى التخزين المشفر. هذا يعني أن التطبيق يقوم بتشفير كلمة المرور الجديدة ثم يطلب من التخزين المشفر حفظها.

11. **تم تخزين كلمة المرور بنجاح:(Password Stored Successfully)**

- يرسل التخزين المشفر رسالة "Password Stored Successfully" (تم تخزين كلمة المرور بنجاح) إلى تطبيق إدارة كلمات المرور، مؤكداً إتمام عملية الحفظ.

12. **رسالة تأكيد:(Confirmation Message)**

- أخيراً، يرسل تطبيق إدارة كلمات المرور رسالة "Confirmation Message" (رسالة تأكيد) إلى المستخدم، لإعلامه بأن العملية قد تمت بنجاح.

3.10 النظام المقترح

هو تطبيق سطح مكتب (Desktop Application) لإدارة كلمات المرور تم تطويره باستخدام لغة بايثون.

الهدف الرئيسي للتطبيق وهو توفير وسيلة آمنة ومحلية للمستخدمين لتخزين وإدارة بيانات اعتماداتهم المختلفة.

3.11 مميزات النظام

خدمة اداره الحسابات

يوفر النظام واجهة شاملة لإضافة، تعديل، وحذف سجلات كلمات المرور الخاصة بالمواقع والتطبيقات المختلفة، مع إمكانية تخزين معلومات إضافية مثل أسماء المستخدمين والملاحظات والتصنيفات .

خدمة انشاء وتوليد كلمات السر القوية

يتضمن النظام أداة مدمجة لتوليد كلمات مرور عشوائية ذات قوة عالية، مما يساعد المستخدمين على تجنب استخدام كلمات مرور ضعيفة أو يسهل تخمينها.

خدمة فحص وتحليل قوة كلمة المرور

يقدم النظام تحليلاً فورياً لقوة كلمة المرور التي يتم إدخالها أو توليدها، مع مؤشر مرئي يوضح مدى أمان كلمة المرور المقترحة.

مصادقة ثنائية (2FA)

يمكن للمستخدمين تفعيل المصادقة الثنائية كطبقة أمان إضافية عند تسجيل الدخول، مما يعزز حماية الحساب بشكل كبير.

إمكانية النسخ الاحتياطي والاستعادة

يتيح النظام للمستخدمين تصدير نسخة احتياطية من قاعدة بيانات كلمات المرور المشفرة، بالإضافة إلى إمكانية استيرادها لاحقاً لاستعادة البيانات.

التخزين المشفر والآمن للبيانات

يتم تشفير جميع كلمات المرور الحساسة باستخدام خوارزمية Fernet القوية، ويتم اشتقاق مفتاح التشفير بشكل آمن من كلمة المرور الرئيسية للمستخدم وحفظها في قاعدة بيانات SQLite محلية، مما يضمن حماية البيانات حتى في حالة الوصول غير المصرح به لملف قاعدة البيانات.

الـــفـــــصـــل الـــرابـــع

❖ الفصل الرابع: التنفيذ

4.1 المقدمة

يُعدّ تنفيذ هذا النظام إحدى المراحل النهائية في بناء المشروع. يهدف هذا النظام إلى تحسين تجربة المستخدم وتسهيل خدمات ادارة كلمات المرور بالاعتماد على تقنيات التشفير الحديثة اضافة الى استخدام ادوات برمجية حديثة.

4.2 سيناريو التنفيذ

سيناريو التنفيذ تطبيق إدارة كلمات المرور (Cyber Pass):

1. **تشغيل التطبيق:**
   - يُفتح التطبيق بشاشة تسجيل الدخول الأساسية.
   - يتم تحميل قواعد البيانات (`users. dB`, `passwords. dB`, `history. dB`) تلقائياً.
   - إذا كانت الملفات غير موجودة، تُنشَأ تلقائياً مع الجداول والأعمدة المطلوبة.

2. **تسجيل حساب جديد:**
   - يُنقر على زر **"إنشاء حساب"** في شاشة تسجيل الدخول.
   - تُملأ الحقول التالية:
     - اسم المستخدم (مطلوب، فريد).
     - البريد الإلكتروني (اختياري).
     - معلومات شخصية (اختياري).
     - كلمة المرور الرئيسية (يجب أن تكون قوية: 8 أحرف على الأقل، تحتوي على أحرف كبيرة وصغيرة وأرقام ورموز).
     - تأكيد كلمة المرور.
     - عند النقر على **"إنشاء الحساب"**
   - عند تسجيل مستخدم جديد، يتم توليد ملح (Salt) فريد لكل مستخدم .

- يتم استخدام كلمة المرور الرئيسية التي يُدخلها المستخدم والملح مع خوارزمية PBKDF2HMACمع SHA256 وعدد كبير من الدورات (لاشتقاق مفتاح. يتم تهشئة هذا المفتاح المشتق وتخزينه مع الملح في قاعدة بيانات المستخدمين .(users. dB) هذا يضمن عدم تخزين كلمات المرور الرئيسية بصيغة مقروءة ويجعل عملية كسرها أكثر صعوبة. تُخزَّن بيانات المستخدم في `users. dB`.

3. تسجيل الدخول:

- يُدخل المستخدم اسم المستخدم وكلمة المرور الرئيسية.
- يُتحقق من صحتها عبر مقارنة الهاش المُخزَّن مع الهاش المُنشَأ من كلمة المرور المدخلة والملح.
- إذا نجح التحقق:
- يُنشَأ مفتاح تشفير Fernet لفك تشفير كلمات المرور المحفوظة.
- إذا كان المصادقة الثنائية (FA2) مُمكَّنة:
- يُطلب من المستخدم إدخال الرمز من تطبيق المصادقة (مثل Google Authenticator).
- تنتقل الواجهة إلى الشاشة الرئيسية.

- عند تسجيل الدخول، يتم جلب الهاش والملح المخزنين للمستخدم من .users. dbتُستخدم كلمة المرور التي أدخلها المستخدم والملح المخزن لإعادة اشتقاق المفتاح ثم تهشئته بنفس الطريقة. إذا تطابق الهاش المشتق حديثًا مع الهاش المخزن، يتم التحقق من كلمة المرور .

4. الواجهة الرئيسية:

4.1 الإحصائيات (الشريط العلوي):

- ع تُعرض بطاقات إحصائية:
- عدد الحسابات المُخزَّنة.
- عدد كلمات المرور الضعيفة.
- عدد الإضافات الحديثة (خلال 7 أيام).
- عدد التصنيفات (Tags).

4.2 إدارة كلمات المرور

- إضافة سجل جديد:

- يُدخل الموقع، اسم المستخدم، كلمة المرور، ملاحظات، وتصنيف (Tag).
- يُنقر على " 💾 حفظ" لتشفير كلمة المرور وحفظها في `passwords. dB`.
- تُحلَّل قوة كلمة المرور تلقائياً.

- عرض/تعديل/حذف السجلات:

- تُعرض كلمات المرور كبطاقات قابلة للتمرير.
- لكل بطاقة أزرار:
- نسخ: نسخ كلمة المرور إلى الحافظة (بعد فك التشفير).
- عرض: إظهار كلمة المرور بشكل مؤقت.
- تعديل: تحميل بيانات السجل في الحقول العلوية للتعديل.
- حذف: حذف السجل بعد التأكيد.

4.3 البحث:

- يُدخل مصطلح بحث في شريط البحث للعثور على سجلات مطابقة (حسب الموقع، اسم المستخدم، الملاحظات، التصنيف).

5. الميزات المتقدمة:

5.1 المصادقة الثنائية (FA2):

- في الإعدادات ← المصادقة الثنائية:
- تمكين: يُولَّد مفتاح سري وعرضه كـ QR Code للمستخدم لمسحه.
- تعطيل: يتطلب تأكيد كلمة المرور الرئيسية.

➢ يتم تطبيق المصادقة الثنائية (2FA) باستخدام مكتبة .pyotpعند تمكينها، يتم توليد مفتاح سري (Secret Key)وتخزينه للمستخدم. عند تسجيل الدخول، بعد التحقق من كلمة المرور الرئيسية، يُطلب من المستخدم إدخال رمز TOTP يتم التحقق من صحته باستخدام المفتاح السري المخزن. يمكن عرض رمز QR للمساعدة في إعداد 2 FAعلى تطبيقات المصادقة الخارجية.

5.2 النسخ الاحتياطي:

- تصدير: حفظ نسخة من `passwords. dB` في مسار مُحدد.
- يمكن للمستخدم تصدير قاعدة بيانات كلمات المرور (passwords. dB) إلى ملف خارجي.
- استيراد: استبدال قاعدة البيانات الحالية بملف نسخة احتياطية.

5.3 إعدادات الجلسة:

- ضبط مدة انتهاء الجلسة (بالثواني) بعد عدم النشاط.

5.4 تغيير كلمة المرور الرئيسية:

- في الإعدادات ← تغيير كلمة المرور:
- يُطلب إدخال كلمة المرور القديمة والجديدة.
- تُعاد تشفير جميع كلمات المرور باستخدام المفتاح الجديد.

➢ عند تغيير كلمة المرور الرئيسية، يتم توليد ملح جديد ومفتاح تشفير Fernet جديد. ثم يتم فك تشفير جميع كلمات المرور المخزنة باستخدام مفتاح Fernet القديم وإعادة تشفيرها باستخدام مفتاح Fernetالجديد قبل تحديث كلمة المرور الرئيسية والملح في قاعدة بيانات المستخدمين.

6. سجل النشاطات (History):

- يتم تسجيل جميع الإجراءات الهامة للمستخدم في جدول history بـ history. dB مع الطابع الزمني ونوع النشاط والتفاصيل ذات الصلة .
- يُعرض سجل تفصيلي لكل الأنشطة (تسجيل الدخول، إضافة/تعديل/حذف كلمات المرور، تغيير الإعدادات).
- يمكن للمستخدم عرض سجل النشاطات الخاص به داخل التطبيق .
- يتم دعم إنشاء تقرير PDF لسجل النشاطات باستخدام مكتبة report lab.

7. المظهر:

- يمكن التبديل بين الوضعين **الغامق** و**الفاتح** من الإعدادات.

8. تسجيل الخروج:

- يُنقر على زر "تسجيل الخروج" في الشريط الجانبي.
- تُغلَق الجلسة، وتُحذَف مفاتيح التشفير من الذاكرة.

9. إغلاق التطبيق:

- تُغلَق اتصالات قواعد البيانات بأمان.
- تُحفَظ الإعدادات (مثل الوضع اللوني).

4.3 خوارزميات التشفير والمكتبات والتقنيات الأمنية الرئيسية

4.3.1 الخوارزميات والتقنيات الأمنية الرئيسية

- PBKDF2 (Password-Based Key Derivation Function 2): تُستخدم لاشتقاق مفتاح تشفير قوي وآمن من كلمة المرور الرئيسية التي يُدخلها المستخدم. هذه الخوارزمية مُصممة لتكون بطيئة computationally، مما يجعل هجمات القوة العمياء (brute-force attacks) أكثر صعوبة بكثير.

```
kdf = PBKDF2HMAC(
    algorithm=hashes.SHA256(),
    length=32,
    salt=salt,
    iterations=100000,
    backend=default_backend()
)
```

- SHA256: هي دالة تجزئة (Hashing Function) تُستخدم كجزء من عملية اشتقاق المفتاح في PBKDF2. تُستخدم أيضاً في بعض أجزاء الكود (وإن كان يبدو أن الاستخدام الرئيسي للهاش هو عبر PBKDF2 للمصادقة الرئيسية).

- Fernet: هي مواصفات تشفير متماثل (Symmetric Encryption) تضمن أن الرسالة المشفرة لا يمكن تعديلها أو قراءتها بدون المفتاح. هي مبنية على عدة بدائيات تشفير، وتستخدم بشكل أساسي خوارزمية التشفير AES (Advanced Encryption Standard) في وضع CBC وتتضمن توقيع رسالة (message signature) لضمان سلامة البيانات. تُستخدم في المشروع لتشفير وفك تشفير كلمات المرور المخزنة.

```
fernet = Fernet(encryption_key)
encrypted_password = fernet.encrypt(password.encode())
```

- TOTP (Time-based One-Time Password): تُستخدم في تنفيذ المصادقة الثنائية (FA2). تولد رموزاً سرية تتغير بناءً على الوقت الحالي والمفتاح السري المشترك (Secret Key) وتوليد رموز مؤقتة للمصادقة الثنائية (FA2) باستخدام مكتبة pyotp.

```
secret = pyotp.random_base32()
totp = pyotp.TOTP(secret)
```

4.3.2 المكتبات البرمجية الرئيسية

- **customtkinter: المكتبة الأساسية لبناء واجهة المستخدم الرسومية (GUI) الحديثة والجذابة للتطبيق.**

```
# إنشاء إطار تسجيل الدخول
login_frame = ctk.CTkFrame(self.root, width=400, height=500)
ctk.CTkLabel(login_frame, text="تسجيل الدخول", font=("Arial", 30)).pack(pady=20)

# زر مع أيقونة
ctk.CTkButton(login_frame, text="💾 حفظ", command=self.save_password)
```

- **sqlite3: مكتبة بايثون القياسية للتفاعل مع قواعد بيانات SQLite، والتي تُستخدم لتخزين بيانات المستخدمين، كلمات المرور، وسجل النشاطات في ملفات منفصلة (users. dB, passwords. dB, history. dB).**

```
# إنشاء جدول المستخدمين
user_cursor.execute("""
    CREATE TABLE IF NOT EXISTS users (
        id INTEGER PRIMARY KEY,
        username TEXT UNIQUE,
        password TEXT
    )""")

# إضافة مستخدم جديد
user_cursor.execute("INSERT INTO users (username, password) VALUES (?, ?)", ("user1", "hash12
3"))
```

- **cryptography: مكتبة قوية للتعامل مع العمليات التشفيرية. يُستخدم منها تحديداً:**
  - **cryptography. Fernet: لتنفيذ التشفير المتماثل لكلمات المرور.**
  - **cryptography.hazmat.primitives.kdf. pbkdf2: لتنفيذ خوارزمية اشتقاق المفتاح PBKDF2.**
  - **cryptography. hazmat. primitives. hashes (مستخدم مع PBKDF2): لتحديد دالة التجزئة (مثل SHA256).**

```
from cryptography.fernet import Fernet
from cryptography.hazmat.primitives.kdf.pbkdf2 import PBKDF2HMAC

# توليد مفتاح تشفير
kdf = PBKDF2HMAC(algorithm=hashes.SHA256(), length=32, salt=salt, iterations=100000)
key = base64.urlsafe_b64encode(kdf.derive(password.encode()))

# تشفير كلمة المرور
fernet = Fernet(key)
encrypted = fernet.encrypt(b"my_password")
```

- **pyotp: مكتبة لتنفيذ خوارزميات كلمات المرور لمرة واحدة بناءً على الوقت (TOTP) والعداد (HOTP)، وتُستخدم هنا لتنفيذ المصادقة الثنائية (FA2).**

```
import pyotp

# 2 توليد مفتاح سري لـ FA
secret = pyotp.random_base32()
totp = pyotp.TOTP(secret)
code = totp.now()  # الحصول على الرمز الحالي
```

- **qrcode: مكتبة لتوليد رموز QR، تُستخدم في المشروع لإنشاء رمز QR لإعداد المصادقة الثنائية بسهولة عبر تطبيقات مثل Google Authenticator.**

```
import qrcode

# توليد رمز QR للمصادقة الثنائية
qr = qrcode.QRCode(version=1, box_size=10, border=4)
qr.add_data("otpauth://totp/Example:user1?secret=SECRET123")
img = qr.make_image(fill_color="black", back_color="white")
img.save("2fa_qr.png")
```

- **report lab: مكتبة لإنشاء مستندات PDF بشكل برمجي، تُستخدم هنا لإنشاء تقرير PDF لسجل النشاطات.**

```
from reportlab.lib.pagesizes import letter
from reportlab.platypus import SimpleDocTemplate, Paragraph

# إنشاء تقرير PDF
doc = SimpleDocTemplate("report.pdf", pagesize=letter)
story = []
story.append(Paragraph("سجل النشاطات", getSampleStyleSheet()['Title']))
doc.build(story)
```

- **hashlib: مكتبة بايثون القياسية لتوليد التجزئات (hashes). تُستخدم في بعض أجزاء الكود (رغم أن الهاش الرئيسي لكلمة المرور يعتمد على PBKDF2).**

```
import hashlib

# تجزئة كلمة مرور (استخدام احتياطي)
hash = hashlib.sha256("password123".encode()).hexdigest()
```

- **pyperclip: مكتبة للتعامل مع الحافظة (clipboard)، تُستخدم لنسخ كلمات المرور المولدة أو المحفوظة.**

```
import pyperclip

# نسخ كلمة المرور إلى الحافظة
pyperclip.copy("decrypted_password")
```

- **مكتبات مساعدة: مثل random, string (لتوليد كلمات المرور)، os, shutil (لعمليات الملفات مثل النسخ في النسخ الاحتياطي)، time, sys, Json (لحفظ واستعادة الإعدادات)، datetime, io, PIL (لتحميل ومعالجة الصور للأيقونات ورمز QR).**

```
import random
import string

# توليد كلمة مرور عشوائية
password = ''.join(random.choice(string.ascii_letters + string.digits) for _ in range(12))
```

```
import os
import shutil

# نسخ قاعدة البيانات للنسخ الاحتياطي
shutil.copy("passwords.db", "backup.db")

# حذف ملف مؤقت
os.remove("temp_file.txt")
```

```
from PIL import Image
import customtkinter as ctk

# تحميل صورة للأيقونة
icon = ctk.CTkImage(light_image=Image.open("icon.png"), dark_image=Image.open("icon_dark.png"))
```

- **تحليل قوة كلمة المرور:**

```
def analyze_password_strength(self, password):
    score = 0
    if len(password) >= 8: score += 1
    if any(c.isupper() for c in password): score += 1
    return ["قوي" ,"متوسط" ,"ضعيف"][score]
```

- **توليد كلمة المرور العشوائية:**

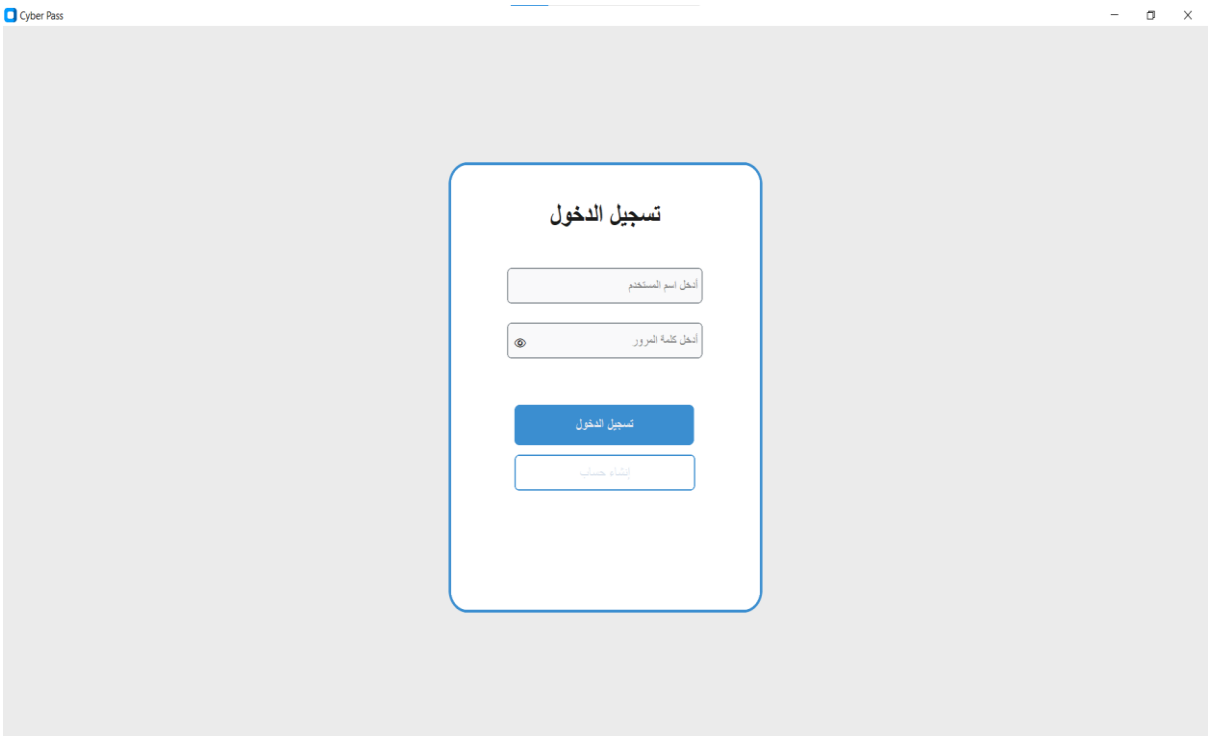
```
def generate_password(self):
    self.check_session_expired()

    # توليد كلمة مرور عشوائية تحتوي على أحرف وأرقام ورموز
    chars = string.ascii_letters + string.digits + string.punctuation
    generated = ''.join(random.choice(chars) for _ in range(12))

    # إدخال كلمة المرور في الحقل وحذف المحتوى السابق
    self.pass_entry.delete(0, "end")
    self.pass_entry.insert(0, generated)

    # نسخ كلمة المرور تلقائياً للحافظة
    pyperclip.copy(generated)

    # تحديث مؤشر قوة كلمة المرور
    self.update_strength()
```

4.4 واجهات النظام

هي الوسيلة التي يتفاعل المستخدم النهائي من خلالها مع النظام.
تسمح للمستخدم بإدخال البيانات، إصدار الأوامر، وتلقي المخرجات والمعلومات من النظام .

4.4.1 واجهه تسجيل الدخول

تعرض حقلي إدخال لاسم المستخدم وكلمة المرور مع عنوان يوضح أنها صفحة تسجيل الدخول. توجد أزرار لتأكيد الدخول (تسجيل الدخول) أو للانتقال إلى صفحة إنشاء حساب جديد (إنشاء حساب)



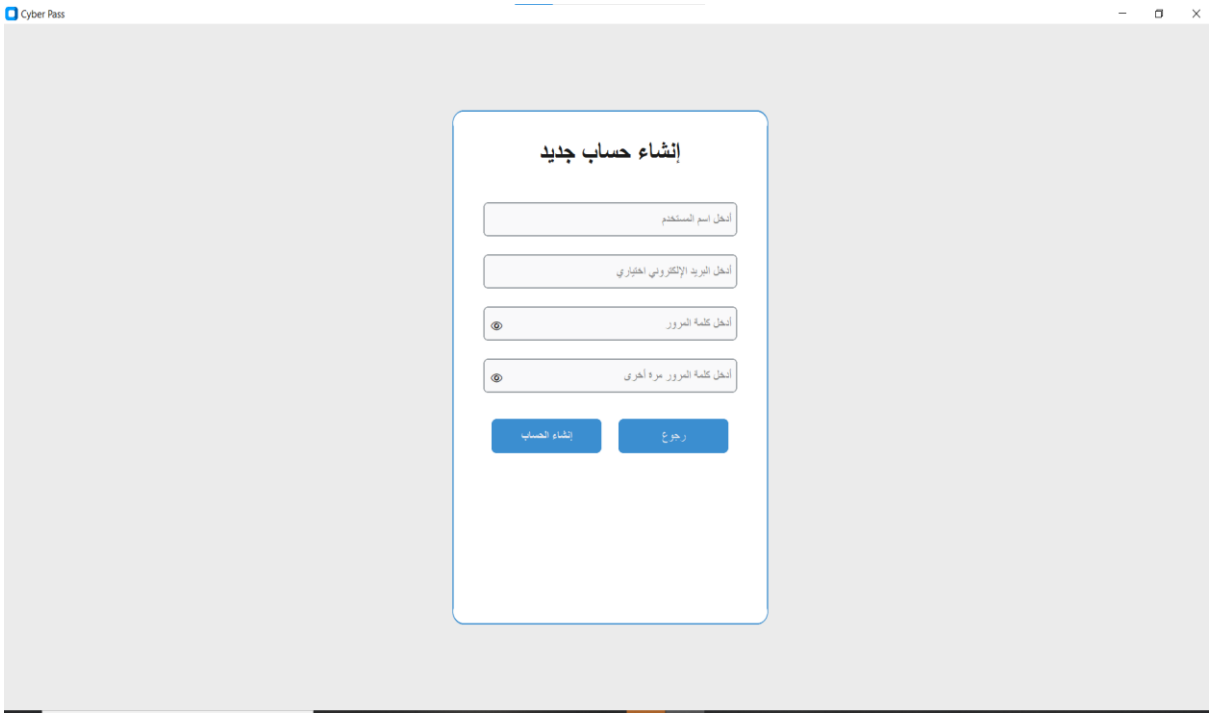
تستخدم CTkFrame لحاويات الواجهة، وCTkLabel لعنوان الصفحة ("تسجيل الدخول")، وCTkEntry لحقلي اسم المستخدم وكلمة المرور (مع خاصية إخفاء النص في كلمة المرور)، بالإضافة إلى CTkButton للأزرار "تسجيل الدخول" و"إنشاء حساب"

4.4.1.1 واجهة تسجيل الدخول (show_login) (الـكـود)

```
def show_login(self):
    self.clear_root()
    # إعداد الإطارات والأزرار
    main_frame = ctk.CTkFrame(self.root, fg_color="transparent")
    login_frame = ctk.CTkFrame(main_frame, width=400, height=500, corner_radius=25)

    # حقول الإدخال
    self._add_field(content, "اسم المستخدم:", 'login_user_entry')
    self._add_field(content, "كلمة المرور:", 'login_pass_entry', show="*")

    # الأزرار
    ctk.CTkButton(btns, text="تسجيل الدخول", command=self.login_user)
    ctk.CTkButton(btns, text="إنشاء حساب", command=self.show_register)
```

4.4.2 واجهه انشاء الحساب

تعرض حقولًا لإدخال بيانات المستخدم الجديد: اسم المستخدم، البريد الإلكتروني، معلومات شخصية، وكلمة المرور مع تأكيدها. تحتوي هذه الواجهة على زر "إنشاء الحساب" لتنفيذ عملية التسجيل، وزر "رجوع" للعودة إلى شاشة تسجيل الدخول.



تستخدم CTkFrame لحاوية الصفحة ومناطق المحتوى، وCTkLabel لعنوان الصفحة ("إنشاء حساب جديد") وبعض التسميات، وCTkEntry لحقول الإدخال المختلفة، وCTkButton لزر إنشاء الحساب وزر الرجوع.

4.4.2.1 واجهة التسجيل:(show_register) (الكود)

```
def show_register(self):
    self.clear_root()
    # حقول الإدخال
    self._add_field(content, "اسم المستخدم:", 'reg_user_entry')
    self._add_field(content, "كلمة المرور:", 'reg_pass_entry', show="*")

    # أزرار
    ctk.CTkButton(btns, text="إنشاء الحساب", command=self.register_user)
    ctk.CTkButton(btns, text="رجوع", command=self.show_login)
```

4.4.3 الصفحة الرئيسية

تضم هذه الواجهة عدة أقسام ضمن النافذة الرئيسية: بطاقات إحصائية في الأعلى (مثل إجمالي الحسابات كلمات ضعيفة، إلخ)، تليها حقول إدخال لإضافة أو تعديل سجل جديد (موقع، اسم مستخدم، كلمة المرور، ملاحظة، تصنيف)، ثم شريط إجراءات يحتوي على أزرار الحفظ والتعديل والتحديث والاستيراد/التصدير، وقسم بحث، وأخيرًا منطقة عرض كلمات المرور المحفوظة (تظهر كبطاقات داخل إطار قابل للتمرير)

حقول الإدخال: CTkLabel مع CTkEntry لكل من: الموقع، اسم المستخدم، الملاحظة، التصنيف، بالإضافة إلى إطار فرعي خاص بكلمة المرور (CTkFrame) يحتوي على CTkEntry مشفرة (show="*")، مع CTkCheckBox لإظهار/إخفاء كلمة المرور، CTkButton لتوليد كلمة مرور تلقائيًا، وCTkLabel + CTkProgressBar لعرض قوة كلمة المرور

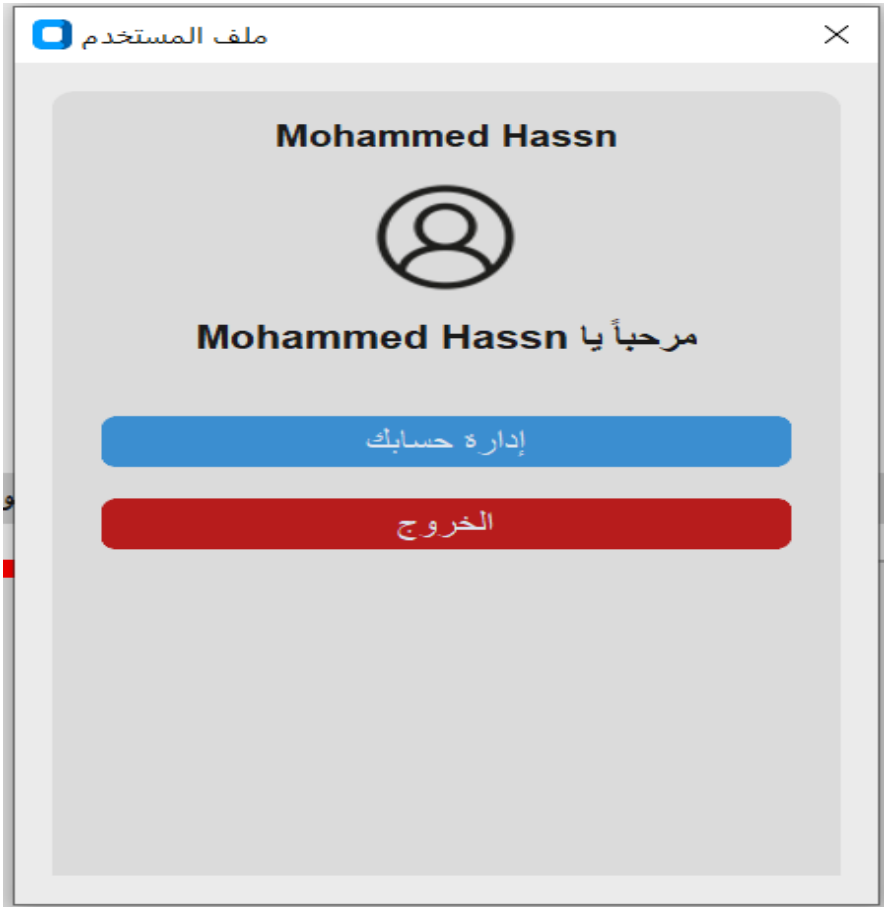
4.4.3.1 الواجهة الرئيسية (show_main_app) (الـكـود)

```
def show_main_app(self):
    self.clear_root()
    self.root.grid_columnconfigure(0, weight=1)
    self.root.grid_columnconfigure(1, weight=0)  # Sidebar column
    self.root.grid_rowconfigure(0, weight=1)

    # ========== Sidebar ==========
    sidebar_bg_color = ctk.ThemeManager.theme["CTkFrame"]["fg_color"]
    self.sidebar_frame = ctk.CTkFrame(
        self.root,
        width=self.sidebar_width_collapsed,
        corner_radius=0,
        fg_color=sidebar_bg_color,
        border_width=0
    )
    self.sidebar_frame.grid(row=0, column=1, sticky="ns", padx=0, pady=0)

    # Sidebar grid configuration
    self.sidebar_frame.grid_rowconfigure(0, weight=0)  # Menu
    self.sidebar_frame.grid_rowconfigure(1, weight=0)  # User
    self.sidebar_frame.grid_rowconfigure(2, weight=1)  # Spacer
    self.sidebar_frame.grid_rowconfigure(3, weight=0)  # History
    self.sidebar_frame.grid_rowconfigure(4, weight=0)  # Help
    self.sidebar_frame.grid_rowconfigure(5, weight=0)  # Settings
    self.sidebar_frame.grid_rowconfigure(6, weight=0)  # Logout
    self.sidebar_frame.grid_columnconfigure(0, weight=1)

    # Sidebar buttons (Menu, User, History, etc.)
    # ...[كود الأزرار الجانبية كما في الملف الأصلي] ...
```

```
# ------ Action Buttons ------
actions = ctk.CTkFrame(self.main_content_frame)
actions.grid(row=2, column=0, pady=10, padx=10, sticky='ew')

buttons = [
    ("💾 حفظ", self.save_password, "green"),
    ("✏️ تعديل", self.update_password, "orange"),
    ("🔄 تحديث الجدول", self.refresh_data, "#0dcaf0"),
    ("📤 تصدير", self.export_backup, "#0d6efd"),
    ("📥 استيراد", self.import_backup, "gray"),
]
```

4.4.4 الملف الشخصي

تظهر كنافذة صغيرة (CTkToplevel) فوق الواجهة الرئيسية بعد الضغط على زر الملف الشخصي. تعرض اسم المستخدم الحالي أو عنوان البريد (مع صورة أيقونية للمستخدم)، وترحيب بسيط. تتضمن الواجهة أزرارًا لإدارة الحساب (تنتقل إلى صفحة الإعدادات) وزر "الخروج" الذي يقوم بتسجيل خروج المستخدم والعودة إلى شاشة تسجيل الدخول.



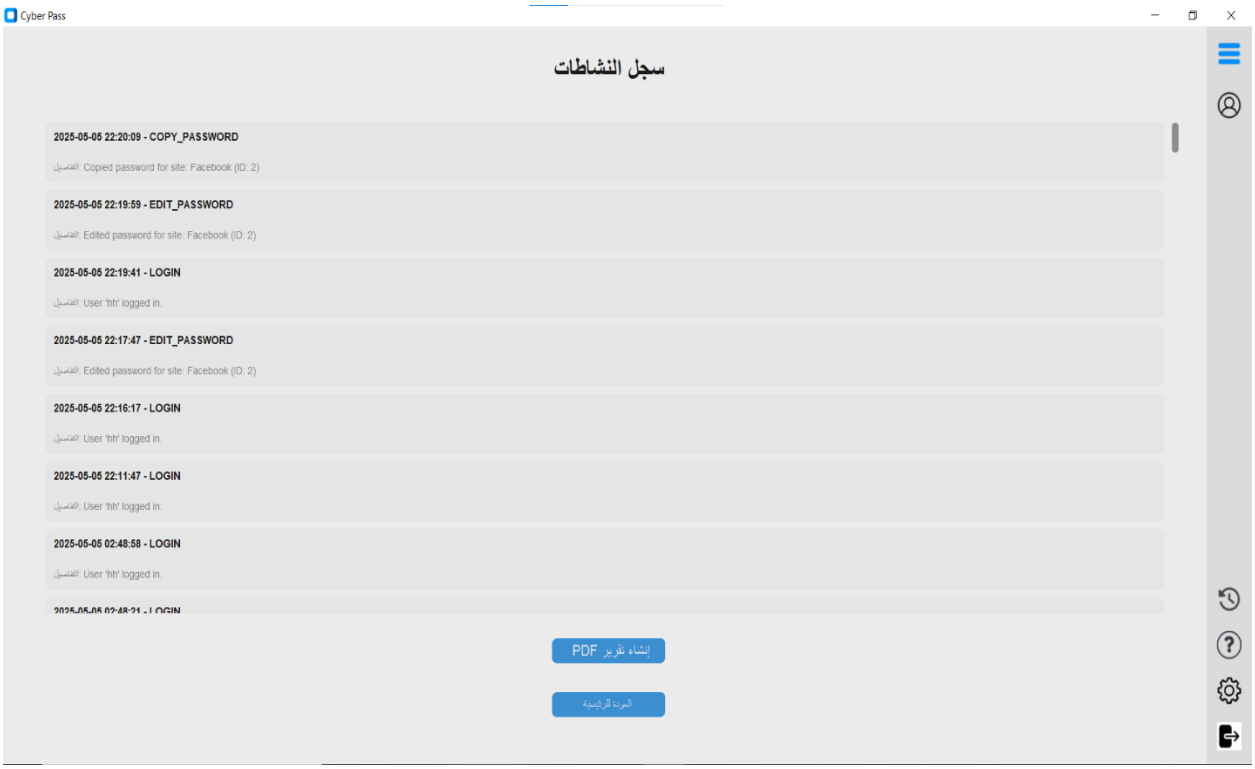
تستخدم CTkToplevel لإنشاء النافذة المنبثقة، وCTkFrame داخلي لتنسيق المحتويات، وCTkLabel لعرض اسم المستخدم وصورة المستخدم الافتراضية أو أيقونته، بالإضافة إلى CTkButton لزر "إدارة حسابك" وCTkButton باللون الأحمر لزر "الخروج"

4.4.4.1 واجهة الملف الشخصي (show_user_profile) (الكود):

```
def show_user_profile(self):
    # معلومات المستخدم
    ctk.CTkLabel(popup_frame, text=self.current_username, font=("Arial", 16))

    # الأزرار
    ctk.CTkButton(..., text="إدارة حسابك", command=self.go_to_settings_from_popup)
    ctk.CTkButton(..., text="الخروج", command=self.logout_from_popup)
```

4.4.5 السجلات او التقارير

تعرض سجلًّا تاريخيًا لعمليات المستخدم (مثل تسجيل الدخول، إضافة/عرض كلمات المرور، تغيير الإعدادات، إلخ) في إطار قابل للتمرير. تبدأ بملصق عنوان **"سجل النشاطات"**، وتحتوي على CTkScrollableFrameيحوي قائمة من السجلات، حيث يتم إنشاء كل سجل ضمن CTkFrame فرعي لعرض طابع الزمن ونوع النشاط والتفاصيل.



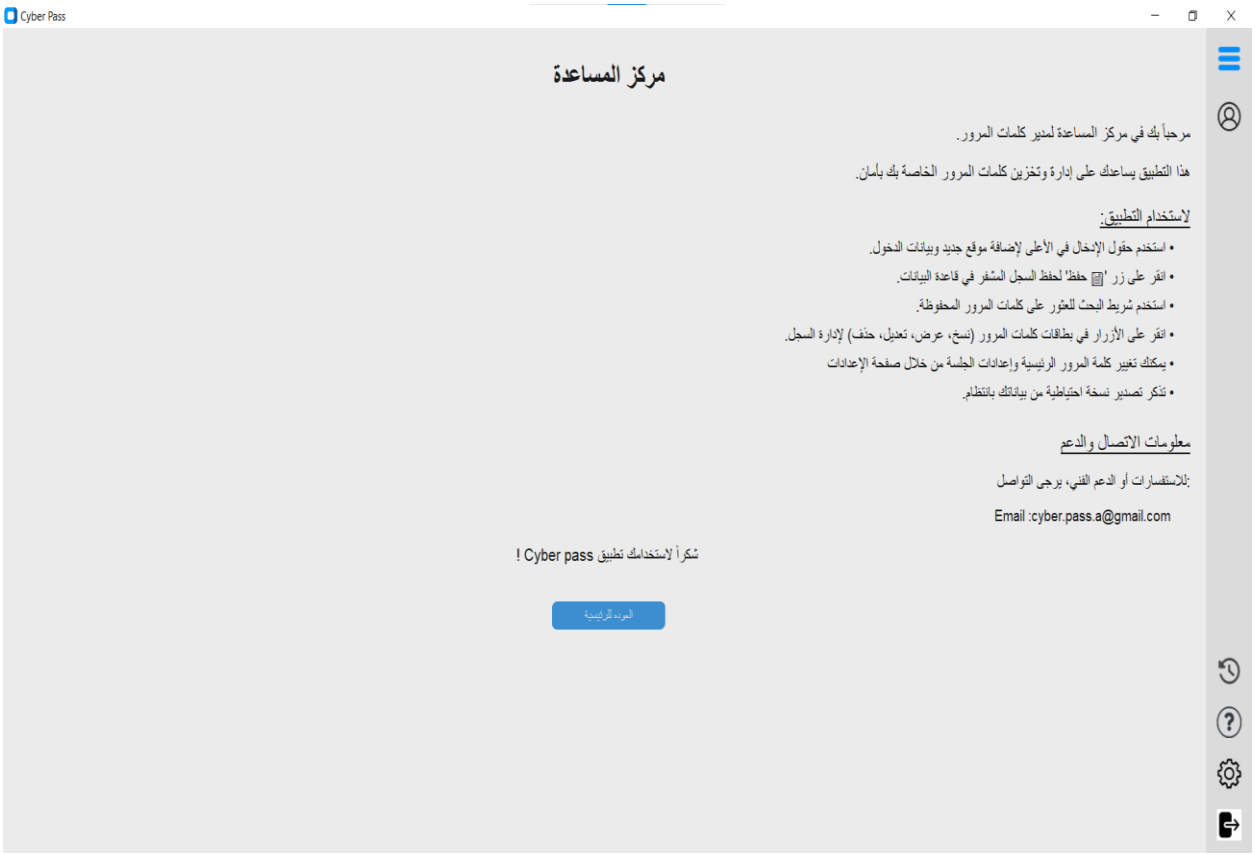
تستخدم CTkFrame لحاوية الصفحة، وCTkLabel للعنوان العام ("سجل النشاطات")، وCTkScrollableFrame لعرض السجلات مع التمرير. داخل الإطار القابل للتمرير، لكل سجل نشاط يُنشأ CTkFrame يحتوي على CTkLabel لنص السجل أو التفاصيل.

4.4.5.1 واجهة المحفوظات (show_history_page) (الكود)

```
def show_history_page(self):
    # جلب السجلات
    history_cursor.execute("SELECT timestamp, activity_type, details FROM history...")

    # عرض البطاقات
    for record in history_records:
        record_frame = ctk.CTkFrame(...)
        ctk.CTkLabel(record_frame, text=f"{timestamp} - {activity_type}")
```

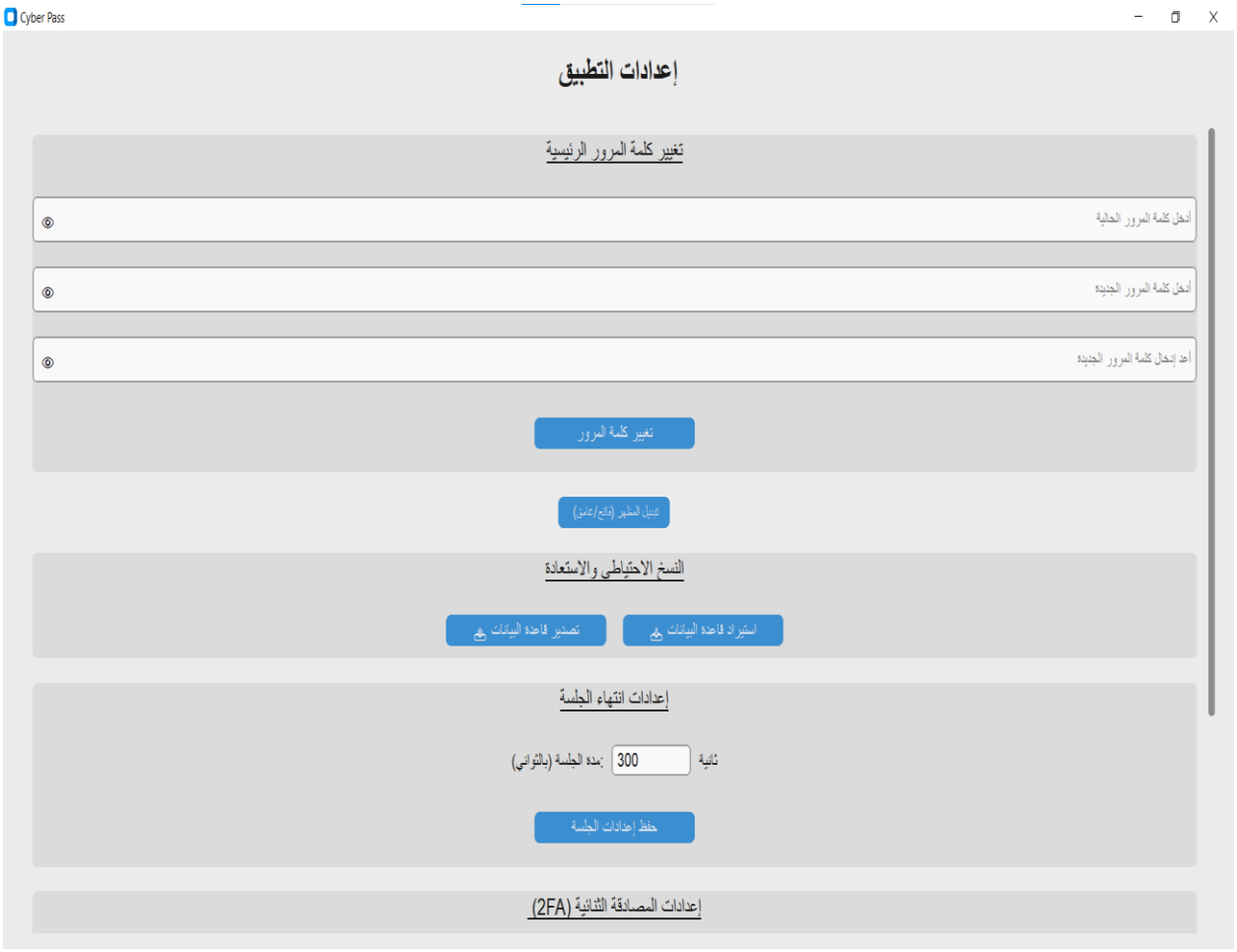
4.4.6 الدعم الفني

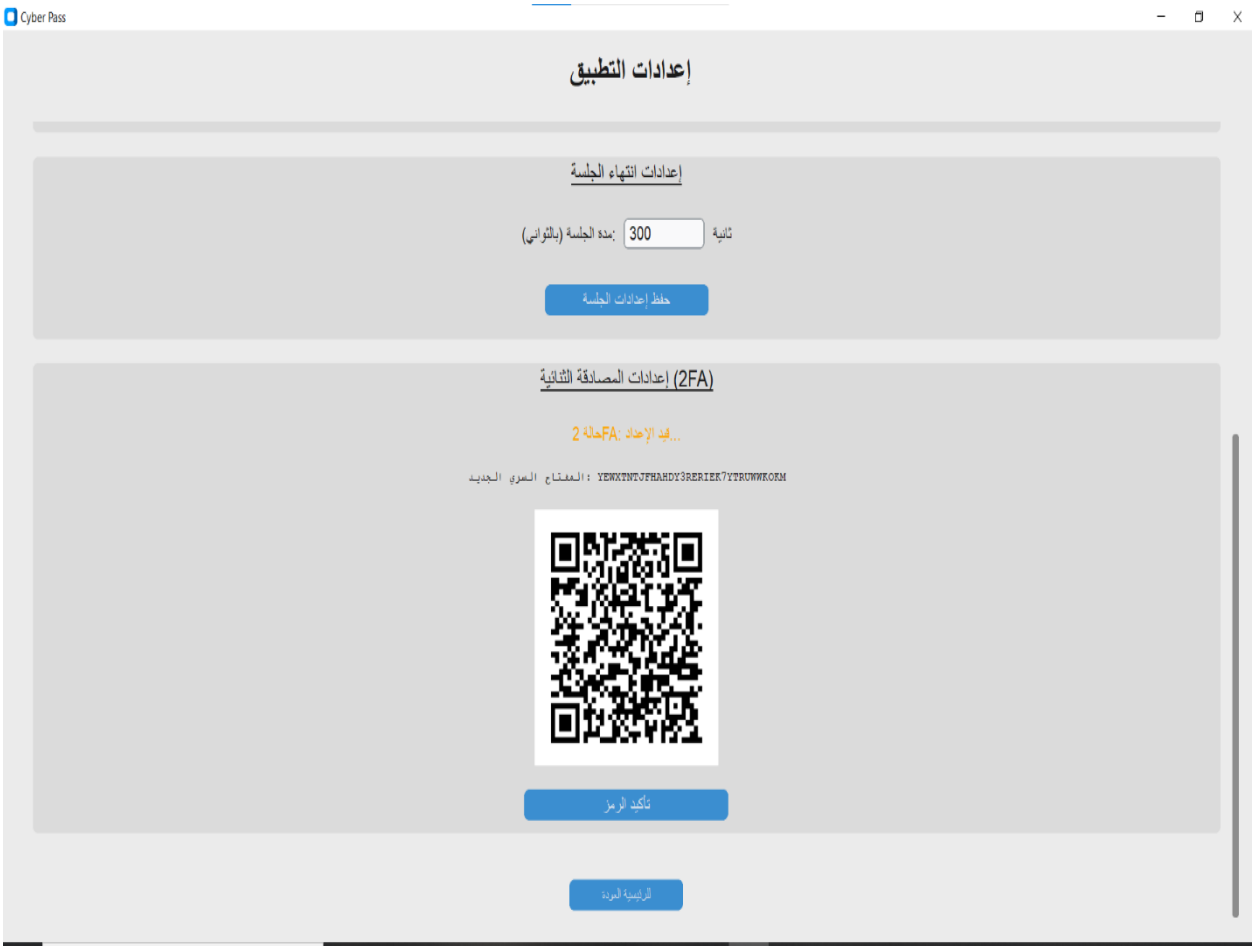
تقدم معلومات إرشادية حول كيفية استخدام التطبيق. تتضمن عدة فقرات (CTkLabel) تشرح خطوات الاستخدام ووظائف الواجهة، تليها قسم "معلومات الاتصال والدعم" مع تفاصيل الاتصال (بريد إلكتروني نموذجي مثلاً)، وزر "للرئيسية العودة" للعودة إلى الواجهة الرئيسية



4.4.7 الاعدادات

- تعرض عدة أقسام قابلة للتمرير ضمن صفحة إعدادات شاملة:
  - قسم **تغيير كلمة المرور الرئيسية** يحوي حقولًا لتغيير كلمة المرور الحالية والجديدة وتأكيدها، مع زر **"تغيير كلمة المرور."**
  - زر **"تبديل المظهر "**لتغيير الثيم بين غامق-فاتح.
  - قسم **النسخ الاحتياطي** فيه أزرار 🔼 **"تصدير قاعدة البيانات"استيراد قاعدة البيانات"**
  - قسم **إعدادات انتهاء الجلسة** فيه حقل إدخال (CTkEntry) لضبط مدة الجلسة بالثواني وزر لحفظ الإعدادات.
  - قسم **إعدادات المصادقة الثنائية (2FA)** يعرض حالة 2 FA(قيد التمكين أو لا) ويدعم توليد وإظهار رمز الـQR ، وزر **"تمكين المصادقة الثنائية "**أو تعطيله
  - زر **"العودة للرئيسية "**للرجوع إلى الواجهة الرئيسية.
- **العناصر الرسومية الرئيسية :**تستخدم CTkFram لمحتوى الصفحة، و CTkLabel لعناوين وأقسام الإعدادات، و CTkScrollableFrameلتغليف المحتوى القابل للتمرير، بالإضافة إلى CTkEntryلحقول إدخال كلمات المرور ومدة الجلسة، و CTkButton لأزرار تغيير كلمة المرور، التبديل، التصدير/الاستيراد، حفظ الإعدادات، وتمكين/تعطيل FA2




Cyber Pass
إعدادات التطبيق
إعدادات انتهاء الجلسة
مدة الجلسة (بالثواني): 300 ثانية
حفظ إعدادات الجلسة
إعدادات المصادقة الثنائية (2FA)
حالة 2FA: قيد الإعداد...
المفتاح السري الجديد: YEWXTNTJFHAHDY3RERIEK7YTRUWWKOXM
تأكيد الرمز
الرئيسية العودة

4.4.7.1 واجهة الإعدادات(show_settings_interface) (الـكـود)

```
def show_settings_interface(self):
    # تغيير كلمة المرور
    self._add_field(password_settings_frame, "كلمة المرور الحالية:", 'current_password_entry', s
how="*")
    ctk.CTkButton(..., command=self.change_master_password)

    # إعدادات المصادقة الثنائية
    self.tfa_status_label = ctk.CTkLabel(...)
    self.tfa_toggle_button = ctk.CTkButton(..., command=self.toggle_tfa)

    # النسخ الاحتياطي
    ctk.CTkButton(..., text="📤 تصدير", command=self.export_backup)
    ctk.CTkButton(..., text="📥 استيراد", command=self.import_backup)
```

```
# ====== قسم إعدادات الجلسة ======
session_settings_frame = ctk.CTkFrame(settings_scrollable_content)
session_settings_frame.pack(pady=10, padx=10, fill="x")

ctk.CTkLabel(session_settings_frame,
             text="إعدادات انتهاء الجلسة",
             font=("Arial", 20, "underline")).pack(pady=(0, 15))

# حقل إدخال مدة الجلسة
session_input_frame = ctk.CTkFrame(session_settings_frame, fg_color="transparent")
session_input_frame.pack(pady=10)

ctk.CTkLabel(session_input_frame,
             text="مدة الجلسة (بالثواني):",
             font=("Arial", 16)).grid(row=0, column=0, padx=5)

self.session_timeout_entry = ctk.CTkEntry(session_input_frame, font=("Arial", 16), width=10

self.session_timeout_entry.grid(row=0, column=1, padx=5)

ctk.CTkLabel(session_input_frame,
             text="ثانية",
             font=("Arial", 16)).grid(row=0, column=2, padx=5)
```

الـــفـــــصـــل الـــخـامـس

❖ الفصل الخامس: الاستنتاجات والتوصيات

5.1 المقدمة

في هذا الفصل، الاستنتاجات والتوصيات، سنستعرض الثمار العملية لهذا الجهد، بدءاً من تحليل مدى تحقيق الأهداف المُعلنة، مروراً بالتحديات التقنية والإجرائية التي واجهت التنفيذ، ووصولاً إلى الرؤى المستقبلية التي يمكن أن تُطور المشروع ليكون أداة أكثر قوة ومرونة. كما سنقدم توصياتٍ عمليةً لتعزيز الأمان، وتحسين الأداء، وتوسيع نطاق الوظائف، مع التأكيد على الأثر المجتمعي الإيجابي الذي يمكن أن يجلبه هذا النظام في تعزيز الثقافة الأمنية الرقمية.

5.2 الاستنتاجات

1 تحقيق الأهداف: نجح التطبيق في تقديم حل آمن وفعّال لإدارة كلمات المرور، حيث وفر ميزات أساسية مثل التشفير باستخدام خوارزمية Fernet، وتخزين البيانات في قواعد بيانات منفصلة، وإدارة سجل النشاطات لتعزيز الشفافية.

2 الأمان: تم تعزيز أمان البيانات من خلال استخدام الملح (*Salt*) ووظائف اشتقاق المفاتيح (*PBKDF2HMAC*)، مما يجعل اختراق كلمات المرور مهمة معقدة.

3 واجهة استخدام تفاعلية: استخدام مكتبة CustomTkinter وفر واجهة رسومية جذابة وسهلة الاستخدام لإدارة الحسابات وكلمات المرور.

4 تجربة المستخدم: واجهة المستخدم الرسومية (GUI) المبسطة مع ميزات مثل الشريط الجانبي القابل للطي، وعرض البطاقات التفاعلية، وأدوات نسخ وعرض كلمات المرور، ساهمت في تجربة مستخدم سلسة.

5 التوثيق الثنائي (FA2): إضافة ميزة المصادقة الثنائية عززت طبقة الحماية، مما يجعل التطبيق مناسبًا للمستخدمين الذين يبحثون عن أمان إضافي.

6 نظام سجلات متكامل:(Logging) تم تسجيل جميع الأنشطة في قاعدة بيانات منفصلة (history. dB)مما يساعد في التتبع والمراجعة.

7 دعم النسخ الاحتياطي والتقارير: توفير خيارات لتصدير واستيراد البيانات، بالإضافة إلى توليد تقارير PDF لأنشطة المستخدم.

5.3 التوصيات

5.3.1 تحسينات تقنية:

- دعم خوارزميات تشفير إضافية: مثل *AES-256* لتعزيز المرونة الأمنية.
- المزامنة السحابية: إضافة خيار لمزامنة البيانات عبر منصات سحابية (مثل Google Drive أو Dropbox) مع الحفاظ على التشفير.
- المصادقة البيو مترية: دعم بصمة الأصبع أو التعرف على الوجه لتسجيل الدخول.

5.3.2 تحسينات وظيفية:

- تكامل مع المتصفحات: إضافة إضافة (*extension*) لمتصفحات الويب لملء كلمات المرور تلقائيًا.
- تقارير أمان مخصصة: توليد تقارير دورية عن قوة كلمات المرور المخزنة ونقاط الضعف.
- مشاركة محدودة: إمكانية مشاركة كلمات مرور مع مستخدمين آخرين مع تحديد الصلاحيات (مثل "للقراءة فقط").

5.3.3 تحسينات تجربة المستخدم:

- وضع الطوارئ: إضافة ميزة لحذف جميع البيانات عن بُعد في حالة فقدان الجهاز.
- إشعارات انتهاء الصلاحية: تنبيه المستخدمين عند استخدام كلمات مرور قديمة أو ضعيفة.
- دعم اللغات: إضافة واجهة متعددة اللغات (مثل الإنجليزية) لتوسيع نطاق الاستخدام.

5.3.4 اختبارات مستقبلية:

- اختبارات اختراق: إجراء اختبارات أمنية مكثفة بالتعاون مع مختصين في الأمن السيبراني.
- تجارب المستخدمين: جمع ملاحظات المستخدمين النهائيين لتحسين التصميم الوظيفي والجمالي.

5.4 التحديات

- التحديات التقنية: واجهنا بعض التحديات في إدارة قواعد البيانات والتأكد من أمان البيانات الحساسة المشفرة. كان اختيار وتطبيق خوارزميات التشفير المناسبة (مثل Fernet المعتمدة على PBKDF2HMAC) أمراً حاسماً، وتطلب فهماً دقيقاً لكيفية التعامل الآمن مع المفاتيح والملح (salt).
- واجهة المستخدم وتجربة المستخدم (UI/UX): كان تصميم واجهة مستخدم جذابة وسهلة الاستخدام تمثل تحدياً، خاصةً مع الحاجة لتوفير ميزات متنوعة مثل إضافة، تعديل، حذف، وبحث كلمات المرور، بالإضافة إلى إدارة الإعدادات والمحفوظات. كان تحقيق استجابة سلسة للعناصر وتكامل الأيقونات والتلميحات يتطلب جهداً إضافياً.
- المصادقة الثنائية (FA2): كان دمج المصادقة الثنائية باستخدام مكتبة pyotp وتوليد رموز QR خطوة مهمة لتعزيز الأمان، لكنها تطلبت فهماً لكيفية توليد المفاتيح السرية، تخزينها بشكل آمن، والتحقق من الرموز المدخلة في وقت تسجيل الدخول.
- النسخ الاحتياطي والاستيراد: تنفيذ وظائف النسخ الاحتياطي والاستيراد لقاعدة البيانات تطلب التعامل مع عمليات نسخ الملفات والتأكد من استمرارية عمل التطبيق بعد عملية الاستيراد.

5.5 جوانب القصور

- التشفير على مستوى المستخدم: حالياً، يعتمد التشفير على كلمة المرور الرئيسية للمستخدم في حالة نسيان كلمة المرور الرئيسية، لا يوجد طريقة لاستعادة البيانات المشفرة، وهذا يمثل قيداً كبيراً من حيث سهولة الاستخدام واستعادة الوصول.
- قابلية التوسع: قد يواجه التطبيق قيوداً في الأداء مع زيادة عدد سجلات كلمات المرور المخزنة، خاصةً عند عمليات البحث أو التحليل التي تتطلب فك تشفير عدد كبير من السجلات.

- المزامنة السحابية: لا يدعم التطبيق حالياً المزامنة عبر أجهزة متعددة أو النسخ الاحتياطي ، مما يتطلب من المستخدم إجراء النسخ الاحتياطي يدوياً.
- التوافق مع المنصات: التطبيق مصمم كواجهة رسومية مكتبية وقد لا يكون متوافقاً أو محسناً للعمل على جميع أنظمة التشغيل أو الأجهزة المحمولة دون تعديلات كبيرة.
- ميزات تحليل الأمان المتقدمة: يوفر التطبيق تحليلاً أساسياً لقوة كلمة المرور، ولكنه يفتقر إلى ميزات تحليل الأمان المتقدمة مثل الكشف عن كلمات المرور المكررة، كلمات المرور المخترقة (عبر قواعد بيانات عامة)، أو تقديم اقتراحات لتحسين الأمان بشكل شامل.

5.6 الأعمال المستقبلية المقترحة

- تحسين نظام استعادة كلمة المرور الرئيسية: يمكن استكشاف طرق آمنة لاستعادة الوصول إلى الحساب في حالة نسيان كلمة المرور الرئيسية، مثل استخدام أسئلة الأمان، مفاتيح الاستعادة، أو الدمج مع خدمات البريد الإلكتروني/الهاتف بطريقة آمنة (مع الأخذ في الاعتبار المخاطر الأمنية المرتبطة بذلك).
- إضافة ميزة المزامنة السحابية: تطوير نظام لمزامنة قاعدة بيانات كلمات المرور بين أجهزة المستخدم المختلفة بشكل آمن ومشفر. يمكن استخدام خدمات تخزين سحابية شائعة مع التأكد من أن عملية المزامنة تتم بشكل يحافظ على خصوصية البيانات.
- توسيع نطاق التطبيق: استكشاف إمكانية تطوير نسخ من التطبيق لأنظمة تشغيل أخرى (مثل iOS/Android) أو كإصدار ويب، مع الحفاظ على نفس مستوى الأمان.
- تسجيل الدخول البيو متري: دعم تسجيل الدخول باستخدام بصمة الإصبع أو التعرف على الوجه (خاصة على الأنظمة الأساسية التي تدعم ذلك) كوسيلة إضافية لتعزيز الأمان.

5.7 الخاتمة

يمثل هذا المشروع خطوة أولى نحو بناء نظام متكامل لإدارة كلمات المرور، وقد حقق توازنًا بين الأمان وسهولة الاستخدام. مع ذلك، فإن التطوير المستمر وتبني التقنيات الناشئة سيضمنان بقاء التطبيق قادرًا على مواكبة التحديات الأمنية المتطورة في المستقبل.

❖ المراجع

[1] Jonathan Katz and Yehuda Lindell, “Introduction to Modern Cryptography”, 2ed edition, CRC Press Taylor & Francis Group, 2015.

[2] Thomas Connolly and Carolyn Begg, “Database Systems A Practical Approach to Design, Implementation, and Management”, 6th edition, University of the west of Scotland, Pearson Education Limited 2015.

[3] Carlos Coronel and Steven Morris, “ Database Systems: Design, Implementation, and Management”, 12th edition, Cengage Learning, 2017.

[4] Al Sweigart, “Automate the Boring Stuff with Python”, 3rd edition, William Pollock, 2025.

[5] Casey Fiesler, Samantha Dalal and Joshua Paup, “Passwords and Python: Introducing Security Concepts in Lower-Division Programming”, In ACM EngageCSEdu. ACM, New York, NY, USA, 2023, 2 pages. https://doi.org/10.1145/3631988

[6] Cleopatra Borg Goga, “Security and Usability: Recommendations for Password User Interfaces”, Master Thesis in Informatics with a specialization in Data Science/Privacy, Information and Cyber Security 30 ECTS, University of Skovde, Autumn 2023

[7] Bimal Krishna k.s, and others, “ Survey on Password Managers”, JARIIE-ISSN(O)-2395-4396, Vol-11 Issue-2 2025.

[8] Hrithik Padalia, Hitesh Patel and Amarjit Deshmukh, “A Study on Password Manager: Users’ Perspective”, International Conference on Computational Intelligence for Information, Security and Communication Applications (CIISCA), June 2023. DOI: 10.1109/CIISCA59740.2023.00024

[9] Ian Maddox and Kyle Moschetto, Modern password security for users User-focused recommendations for creating and storing passwords, Google Cloud Solutions Architects https://cloud.google.com/solutions/modern-password-security-for-users.pdf

[10] Daniel Pecuch, “Password managers: a survey”, Bachelor’s project, Masaryk University Faculty of Informatics, Brno, Fall 2020.

[11] Danuvasin Charoen, “Password Security”, International Journal of Security (IJS), Volume (8) : Issue (1) : 2014.

[12] Shomope, Adewale A.and Dr. Akanni Adeniyi, “Enhancing Digital Security: A Comprehensive Review of Password Management Practices and Tools“, International Journal of Mathematics and Computer Research, Volume 13 Issue 02 February 2025.

[13] Lip Yee Por, Yen-Lin Chen and Jing Yang, “A Systematic Literature Review on the Security Attacks and Countermeasures Used in Graphical Passwords ”, Digital Object Identifier 10.1109/ACCESS.2024.3373662, 2024.