МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ СИСТЕМ ЗАХИСТУ ІНФОРМАЦІЇ

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри Комп'ютеризованих систем захисту інформації

_____ Михайло СТЕПАНОВ
«____»_____ 2023 р.

На правах рукопису
УДК 004.056.5:510.22(043.3)

КВАЛІФІКАЦІЙНА РОБОТА

ЗДОБУВАЧА ВИЩОЇ ОСВІТИ
ОСВІТНЬОГО СТУПЕНЯ «МАГІСТР»

Тема: Програмний застосунок менеджера паролів на основі блокчейн технологій.

Виконавець: Крістіна ДЕМЧЕНКО

Науковий керівник: к.т.н., доц. Анна ІЛЬЄНКО

Консультант розділу «Охорона навколишнього середовища»: к.т.н., доцент Тетяна ДМИТРУХА

Нормоконтролер: к.т.н., доц. Анна ІЛЬЄНКО

Київ 2023

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет: Кібербезпеки та програмної інженерії

Кафедра: Комп'ютеризованих систем захисту інформації

Освітній ступінь: Магістр

Спеціальність: 125 «Кібербезпека»

Освітньо-професійна програма: «Безпека інформаційних і комунікаційних систем»

ЗАТВЕРДЖУЮ
Завідувач кафедри Комп'ютеризованих систем захисту інформації

_____ Михайло СТЕПАНОВ
«__» _____ 2023 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи
здобувача вищої освіти Демченко Крістіни Миколаївни

1. Тема: *Програмний застосунок менеджера паролів на основі блокчейн технологій.*
затверджена наказом ректора від «15» вересня 2023 р. № 1814/ст.
2. Термін виконання: з 16.10.2023 р. по 31.12.2023 р.
3. Вихідні дані: проаналізувати існуючі системи та методики аналізу і оцінки менеджерів паролів; на основі аналізу виділити вхідні і вихідні параметри, сильні та слабкі сторони застосунків такого типу, завдяки ним провести порівняння існуючих систем; дослідити блокчейн технологію та її особливості, розробити методику, алгоритм та на їх основі програмний застосунок менеджера паролів на основі блокчейн технолгій; тестування функціональних можливостей розробленого застосунку.

4. Зміст пояснювальної записки: аналіз існуючих систем та методик забезпечення безпеки паролів користувачів і оцінки ризиків інформаційної безпеки; розробка методики та алгоритму менеджера паролів на основі блокчейн технолгій; створення програмного забезпечення запропонованої системи, верифікація отриманих результатів та порівняння з подібними існуючими системами.

5. КАЛЕНДАРНИЙ ПЛАН
виконання кваліфікаційної роботи

№ з/п	Етапи виконання кваліфікаційної роботи	Термін виконання етапів	Примітка
1.	Уточнення постановки задачі	16.10.2023	*Виконано*
2.	Аналіз літературних джерел	17.10.2023	*Виконано*
3.	Обґрунтування вибору рішення	20.10.2023	*Виконано*
4.	Збір інформації	28.10.2023	*Виконано*
5.	Дослідження сучасних систем і методик аналізу та оцінки менеджерів паролів	05.11.2023	*Виконано*
6.	Дослідження блокчейн технології та її особливостей	14.11.2023	*Виконано*
7.	Розробка методики та алгоритму реалізації програмного застосунку менеджера паролів на основі блокчейн технолгій	21.11.2023	*Виконано*
8.	Створення програмного забезпечення запропонованої системи	29.11.2023	*Виконано*
9.	Перевірка на антиплагіат	12.12.2023	*Виконано*
10.	Оформлення і друк пояснювальної записки	14.12.2023	*Виконано*
11.	Оформлення презентації	17.12.2023	*Виконано*
12.	Отримання рецензій від рецензента	22.12.2023	*Виконано*

6. Консультанти з окремих розділів

Розділ	Консультант (посада, П.І.Б.)	Дата, підпис	
		Завдання видав	Завдання прийняв
Охорона навколишнього середовища	Дмитруха Т.І.		

7. Дата видачі завдання: «16» жовтня 2023 р.

Здобувач вищої освіти Крістіна ДЕМЧЕНКО
(підпис, дата)

Керівник кваліфікаційної роботи Анна ІЛЬЄНКО
(підпис, дата)

РЕФЕРАТ

Кваліфікаційна робота на тему: «Програмний застосунок менеджера паролів на основі блокчейн технологій» складається зі вступу, основної частини, що містить 4 розділи, 3 висновки до кожного розділу, загального висновку, 2 додатків та списку використаної літератури. Загальний обсяг роботи – 98 сторінок. Робота містить 20 рисунків та 6 таблиць. Список використаних джерел включає 45 джерел.

Метою кваліфікаційної роботи є розробка програмного застосунку менеджера паролів на основі блокчейн технологій.

У кваліфікаційній роботі розглянуті питання щодо сучасних систем і методик аналізу та оцінки менеджерів паролів, а також блокчейн технології та її особливостей.

Проведені дослідження базуються на сучасних методах побудови захищених інформаційних мереж та новітніх рішеннях щодо управління паролями, а такожвключають основні принципи технології блокчейн.

Реалізація власного програмного застосунку менеджера паролів на основі блокчейн технологій та вдосконаленняалгоритму з допомогою технології блокчейн, що дозволяє значно покращити рівень довіри досвідчених користувачів до застосунку.

Запропоновані алгоритми та методи дозволяють забезпечити швидкий та надійний захист інформаційно-телекомунікаційних мереж.

Ключові слова: МЕНЕДЖЕР ПАРОЛЕЙ, БЛОКЧЕЙН, УПРАВЛІННЯ ПАРОЛЯМИ, BLOCKCHAIN, ЛАНЦЮЖОК БЛОКЧЕЙНУ, РОЗПОДІЛЕНИЙ РЕЄСТР, БЕЗПЕКА БЛОКЧЙЕНУ.

ЗМІСТ

ВСТУП 7
РОЗДІЛ 1. АНАЛІЗ СУЧАСНИХ РІШЕНЬ ЩОДО УПРАВЛІННЯ ПАРОЛЯМИ 9
1.1 Проблеми щодо забезпечення безпеки інформаційних мереж. 9
1.2. Використання блокчейн технологій для забезпечення інформаційної безпеки. 19
1.3. Огляд сучасних практичних підходів щодо управління паролями. 26
1.4. Висновки до першого розділу. 32
РОЗДІЛ 2. ДОСЛІДЖЕННЯ ОСНОВНИХ ХАРАКТЕРИСТИК BLOCKCHAIN ТЕХНОЛОГІЙ 34
2.1 Фундаментальні складові технології blockchain. 34
2.2 Принцип формування генезис блоку. 41
2.3 Масштабування blockchain. 42
2.4 Властивості технології blockchain та її використання. 48
2.5 Висновки до другого розділу. 55
РОЗДІЛ 3. ПРОГРАМНИЙ ЗАСТОСУНОК МЕНЕДЖЕРА ПАРОЛІВ НА ОСНОВІ БЛОКЧЕЙН ТЕХНОЛОГІЙ 57
3.1 Опис середовища розробки. 57
3.2 Структура та функціонал менеджера паролей. 59
3.3 Оцінка ефективності та порівняння з існуючими системами. 70
3.4. Висновки до третього розділу. 75
РОЗДІЛ 4. ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА 76
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 84
Додаток А 89
Додаток Б 90

ВСТУП

Актуальність. У наші дні люди все частіше використовують паролі, ними захищають банківські рахунки, хмарні сховища, сторінки в соціальних мережах, файли з цінною для них особистою інформацією, тощо. Окрім того, будь який вихід в інтернет відбувається через обліковий запис, який також захищається паролем. Тому проблема надійності паролів, їх збереження та захист від несанкціонованого доступу є однією із найбільш актуальних в області інформаційної безпеки на даний момент. Сьогодні люди бояться забути пароль та втратити доступ до важливої для них інформації чи ресурсу, тому вони використовують занадто прості паролі або однакові паролі на кількох ресурсах. Використання менеджера паролів має змінити це.

Якісні менеджери паролів дозволяють не лише згенерувати пароль, який відповідає усім канонам щодо захисту, а й зберігати усі паролі в захищеному місці й забезпечувати їх конфіденційність, спостережність, невідмовність, цілісність та доступність.

Зараз блокчейн технології дуже активно розвиваються і щодня доводять свою ефективність та безпечність світу. Блокчейн мережі з їх децентралізованою та безпечною структурою, пропонують унікальний підхід до збереження та передачі даних, який обов’язково стане революційним у сфері захисту інформації. Ця технологія дозволяє забезпечити прозорість, незмінність та безпеку даних, що зберігаються та передаються, пропонуючи новітні рішення щодо управління паролями. Саме через це розробка програмного застосунку менеджеру паролів, який базується на блокчейн-технологіях, виглядає особливо перспективною.

Мета - розробка програмного застосунку менеджера паролів на основі блокчейн технологій.

На основі поставленої мети завданнями робти є:

1) аналіз існуючих систем та методик забезпечення безпеки паролів користувачів і оцінки ризиків інформаційної безпеки;

2) розробка алгоритму менеджера паролів на основі блокчейн технологій;

3) оцінка доцільності отриманих результатів і порівняння з подібними існуючими системами.

Обєкт дослідження - процес управління паролями в сучасних інформаційних мережах.

Предмет дослідження - менеджер паролей, блокчейн технологія, інформаційна мережа.

Методи. Проведені дослідження базуються на сучасних методах побудови захищених інформаційних мереж, методах та підходах щодо управління паролями, методах застосування блокчейн технологій.

Наукова новизна полягає у тому, що запропоновано авторський програмний застосунок менеджера паролів на основі блокчейн технологій, що забезпечує інноваційний підхід управління паролями з розподіленим зберігання даних і підвищену безпеку через незмінність записів, а також відсутність у застосунку централізованого контролю знижує ризики атак.

Практична цінність. Розроблено програмний застосунок менеджера паролів з використанням мови програмування Python та веб-додатку Jupyter Notebook, за рахунок впровадження блокчейн технологій. Ця технологія дозволяє забезпечити прозорість, незмінність та безпеку даних, що зберігаються та передаються.

Апробація.

1. Ільєнко А.В., Демченко К. М., Кравчук І. А. Характеристика сучасних рішень щодо управління паролями // Modern problems of science, education and society: VIII Міжнародна науково-практична конференція, 9-11 жовтня 2023 р.: тези доповіді. – К., 2023. – С.281-285.

РОЗДІЛ 1. АНАЛІЗ СУЧАСНИХ РІШЕНЬ ЩОДО УПРАВЛІННЯ ПАРОЛЯМИ

1.1 Проблеми щодо забезпечення безпеки інформаційних мереж.

З розвитком інформаційних мереж розвиваються і проблеми щодо забезпечення їх безпеки. Будь яка інформаційна мережа має бути захищеною від несанкціонованого доступу, будь якого випадкового чи навмисного втручання в мережу, а також убезпечена від можливості руйнування будь-яких її компонентів.

Інформаційна мережа має відповідати 5 основним критеріям оцінки інформаційної безпеки. До таких критеріїв належать:

- Конфіденційність – захист від несанкціонованого доступу до інформації мережі, який зазвичай забезпечується шляхом розподілу прав доступу.
- Цілісність – захист від будь яких несанкціонованих модифікацій інформації в мережі. Забезпечення цілісності передбачає що усі ресурси мережі залишаються невикривленими в наслідок пошкодження або несанкціонованого доступу до мережі.
- Доступність – передбачає забезпечення безперервного доступу до інформації та послуг у мережі усім користувачам, які мають права доступу. Для досягнення доступності важливо підтримувати працездатність мережі, для цього використовують різноманітні механізми відмовостійкості, резервне копіювання або різноманітні мережеві заходи безпеки.
- Спостережність – здатність відстежувати потенційні загрози та аномалії які можуть зашкодити роботі мережі, а також проведення аналізу подій для виявлення інциндентів щодо безпеки інформаційної мережі.

- Невідмовність – здатність мережі виконувати свої функції навіть при виникненні помилок, збої чи відмові програмного чи апаратного забезпечення.

Загалом для досягнення безпеки будь-якої інформаційної мережі застосовують сукупність різних заходів, які мають доповнювати один одного та запобігати, відстежувати та у разі виникнення усувати будь-які спроби несанкціонованого доступу в систему осіб без належних прав доступу. Крім того, для захисту системи важливо захистити її від спотворення, копіювання, блокування або пошкоджень будь-яких даних системи[2].

У наші дні проникнення в систему та крадіжка інформації можуть призвести до вкрай негативних наслідків, таких як матеріальні збитки, удар по репутації компанії чи особи. Наразі сучасні збитки від кібератак, які були зафіксовані оцінюють приблизно у 600 мільярдів доларів[8]. Враховуючи, що багато людей соромляться зізнатись в тому що вони стали жертвою кібератаки, а також невідомо як оцінювати моральну шкоду та збитки від втраченої репутації реальні суми збитків врази перевищують офіційну суму.

Усі загрози безпеці інформаційної мережі призводять до порушення роботи системи на якомусь конкретному носії та виявляються через певні фактори вразливості[12]. До таких факторів зазвичай відносять:

- вади програмного або апаратного забезпечення;
- відмінність параметрів побудови різних автоматизованих систем інформаційної мережі;
- недосконалість протоколів обміну інформації та інтерфейсів;
- складність забезпечення усіх експлуатаційних вимог та створення відповідних умов для розташування інформації;
- неповноцінність деяких процесів функціонування систем.

Найчастіше загрози використовуються з метою незаконного отримання вигоди шляхом завдання шкоди інформації, яка обробляється інформаційною мережею[11]. Однак також можливий випадковий вплив загроз через

недостатній рівень захисту цієї інформації і широкий вплив загрозливих факторів.

Всі вразливості інформаційної мережі поділяються на три класи:

1. об'єктивні вразливості;
2. суб'єктивні вразливості;
3. випадкові вразливості.

Шляхом усунення чи послаблення впливу уразливостей можливо запобігти повноцінній загрозі, яка була спрямована на систему зберігання інформації[9].

Кожна вразливість повинна бути знайдена та оцінена фахівцями з кібербезпеки. Тому є спеціальні критерії оцінки ступеня небезпеки виникнення загрози та ймовірності виходу з ладу або обходу захисту оброблюваної в мережі інформації[1]. Ці показники розраховуються за допомогою ранжування. Основними вважаються три найважливіших критерії:

1. Доступність - критерій, що перевіряє, наскільки джерелу загроз «зручно» використовувати той чи інший вид уразливості для порушення інформаційної безпеки. До цього показника включаються технічні дані носія інформації, як наприклад вартість обладнання, його габарити, складність або можливість його використання для атак на інформаційні системи деякими неспеціалізованими засобами або пристроями.

2. Фатальність - критерій, який оцінює рівень впливу вразливості на здатність програмістів подолати наслідки створеної для інформаційної мережі загрози. Якщо розглядати тільки об'єктивні вразливості, то в критерії визначається інформативність загрози, тобто здатність передати корисний сигнал із конфіденційними даними в будь-яке необхідне місце без його зміни чи спотворення.

3. Кількість - критерій розрахунку компонентів системи зберігання інформації або її реалізації, яким властивий певний вид уразливості в системі.

Несанкціонований доступ вважається одним з найпопулярніших методів комп'ютерних порушень. Маючи такий доступ, зловмисники можуть активно

використовувати усі вразливості в системі безпеки, щоб отримати секретну інформацію[1]. Неправильно обрані налаштування та методи захисту значно збільшують можливість несанкціонованого доступу. Доступ і загрози інформаційній безпеці здійснюються як локальними методами, так і з використанням спеціалізованих пристроїв.

Серед основних причин виникнення несанкціонованого доступу фахівці з кібербезпеки особливо виділяють наступні:

- Існування прогалини в організації захисту при використанні різноманітних засобів авторизації. Сюди відносять паролі, які не відповідають канонам, збереження даних, які використовуються для авторизації в системі, тощо.
- Використання застарілого програмного або апаратного забезпечення, яке не підходить під налаштування саме цієї мережі, що призводить до програмних помилок або конфліктів під час роботи. Таку проблему можливо вирішити шляхом використання лише ліцензованого програмного забезпечення, своєчасного оновлення усіх застосунків, дотримання усіх стандартних правил комп'ютерної безпеки, звернення лише до профільних експертів, тощо.
- Зловживання довірою або службовими повноваженнями.
- Використання троянів, кейлоггерів та інших подібних інструментів, що нагадують кібершпигунство.
- Система контролю доступу до певних програм та даних налаштовані неправильно.
- Різні канали зв'язку прослуховуються або перехоплюються різними методами[10].

До основних способів отримання несанкціонованого доступу відносять:

1. Злам інформаційних ресурсів (хмарних сервісів, різноманітних веб-сайтів, корпоративних мереж, окремих комп'ютерів або мобільних пристроїв).
2. Шантаж, підкуп або вимагання.

3. Перехоплення повідомлень. Сюди відносяться будь-які відправлені повідомлення, такі як повідомлення в соціальних мережах, електронна пошта, SMS повідомлення.

4. Крадіжка інформації.

5. Збір даних. Може здійснюватися законними способами, але з незаконною метою.

Завдяки доступу шахраї можуть не тільки проникнути в інформацію та скопіювати її, але й внести зміни та видалити певні дані[14]. Для цього використовуються:

За допомогою такого доступу дуже часто шахраї можуть не лише скопіювати будь яку інформацію та використати її в своїх цілях, а й внести певні зміни, або взагалі видалити дані. Це відбувається за допомогою: міжмережевих екранів, методів виявлення помилок, локальних ліній доступу до даних або панелей регулювання, перехоплення електромагнітних випромінювань від апаратури, каналів зв’язку чи елементів заземлення.

Найсильнішою і найпопулярнішою загрозою витоку в інформаційній мережі є викрадення через шкідливе програмне забезпечення[24].

Розглянемо основні способи такого витоку.

Троянський кінь. Це тип зловмисного програмного забезпечення, який на перший погляд здається нешкідливим але насправді приховує у собі зловмисний код. Серед основних ознак троянських коней виділяють неочікувані зміни в налаштуваннях комп’ютера або незвична активність.

Троянського коня також часто називають вірусом "троянський кінь", але з технічної точки зору це неправильно. На відміну від комп'ютерного вірусу, троянський кінь не здатний до самовідтворення і не може поширюватися без допомоги кінцевого користувача. Зловмисники повинні використовувати тактику соціальної інженерії, щоб обдурити кінцевого користувача і змусити його запустити троянського коня.

Оскільки існує дуже багато різновидів троянських коней, цей термін можна використовувати як загальне позначення для доставки шкідливого

програмного забезпечення. Залежно від намірів зловмисника і структури програми, троянський кінь може працювати по-різному, іноді поводитися як самостійне шкідливе ПЗ, іноді служити інструментом для інших дій, таких як доставка корисного навантаження, відкриття системи для атак або спілкування зі зловмисником[15].

Коли троянський кінь потрапляє в систему, він може почати шкідливу діяльність, навіть не залишаючи слідів, поки користувач не помітить інцидент. Існують різні трояни, деякі можуть залишатись в системі в сплячому режимі, чекаючи команди від хакера, в той час як інші відразу ж починають свою шкідливу діяльність.

Деякі з троянів завантажують на комп'ютер додаткове шкідливе програмне забезпечення і намагаються обійти налаштування безпеки. Інші можуть намагатися відключити антивірусне програмне забезпечення[16]. Трояни можуть навіть захопити комп'ютер і використовувати його для DDoS-атак (розподілена відмова в обслуговуванні).

DOS та DDoS атаки. Іноді трапляються ситуації, коли сервер веб-сайту перевантажується трафіком і виходить з ладу, часто це відбувається під час публікації важливих новин. Однак найчастіше це відбувається під час атаки DoS (відмова в обслуговуванні) або DDoS (розподілена атака відмови в обслуговуванні), коли зловмисники переповнюють веб-сайт шкідливим трафіком. Коли веб-сайт отримує занадто багато трафіку, він не може виконувати запити своїх користувачів належним чином.

DoS-атака виконується однією машиною, підключеною до Інтернету, шляхом надсилання пакетів на веб-сайт і заважає законним користувачам отримати доступ до вмісту перевантаженого веб-сайту.

DDoS-атака, або розподілена атака відмови в обслуговуванні, аналогічна DoS-атаці, але потужніша. Подолати DDoS-атаку складніше, оскільки її запускають із кількох комп'ютерів, кількість яких може варіюватися від кількох до тисяч, і всі ці комп'ютери можуть бути включені в ботнет - мережу скомпрометованих комп'ютерів, що можуть бути в усьому світі[16]. Оскільки

атака виконується з безлічі різних IP-адрес одночасно, її набагато складніше виявити і захиститися від неї.

Вірус. Програма, яка може самостійно копіюватися і передаватися з одного комп'ютера на інший з метою порушити їх нормальну роботу. Вони часто поширюються через вкладення в електронних листах або завантажуються з певних веб-сайтів, щоб заразити комп'ютер та інші комп'ютери які перебувають в мережі[13]. Віруси можуть виконувати різні дії, як наприклад надсилання спаму, вимкнення систем захисту, пошкодження даних і навіть крадіжку конфіденційної інформації, включно з паролями. У деяких випадках вони можуть навіть знищувати дані на жорсткому диску комп'ютера.

Існує кілька поширених видів комп'ютерних вірусів:

Віруси, що перезаписують. Ці віруси переписують себе поверх вихідного коду іншого файлу, не змінюючи його ім'я. У результаті цього оригінальний додаток більше не працює, і під час його запуску активується шкідлива програма.

Віруси компаньйони. Це вид комп'ютерних вірусів, який створює копію зараженого файлу, переміщаючи або змінюючи ім'я оригіналу[23]. Вихідна програма залишається працездатною, але вона починає виконувати свої функції тільки після запуску шкідливого коду.

Паразитичні віруси. Вони є одними з найпоширеніших оскільки вважаються найпростішими. "Паразити" впроваджують вірусний код у будь-яку частину файлу, і додаток продовжує частково або повністю функціонувати.

Ланки. Комп'ютерні віруси цього типу націлені на зміну адреси, за якою знаходиться програмне забезпечення. У результаті операційна система запускає шкідливу програму замість потрібного додатка без жодних змін у його коді.

Деструктивні віруси. Метою таких комп'ютерних вірусів є завдання шкоди програмі. Вони пошкоджують вихідний код програмного забезпечення або його компонентів, що призводить до непрацездатності програми[26].

Spyware. Програмне забезпечення, яке може збирати та передавати зловмиснику інформацію про будь які дії користувача комп'ютера, який

завантажив це програмне забезпечення, через це такі застосунки часто називають програмою-шпигуном. Зазвичай Spyware використовується в маркетингових цілях для того, щоб на основі відстеження дій користувача відображати йому відповідну рекламу. Наразі ці програми зустрічаються дуже часто і вважаються одними з найрозповсюдженіших.

Джерела шпигунських програм можна розділити на два основні канали. Перший із них - нелегальний і мало відрізняється від поширення інших шкідливих програм. Зловмисники можуть обманом приводити користувачів до встановлення шпигунського додатка або непомітно впроваджувати його через незакриті вразливості. Другий канал - легальний. Шпигунський модуль може міститися у звичайному програмному забезпеченні або ж встановлюватися додатково (користувач не помічає цього в процесі встановлення). Прикладом другої групи можуть слугувати деякі версії "фірмових" браузерів і пов'язаних з ними панелей інструментів, які абсолютно законно збирають для своїх розробників велику кількість інформації про дії користувачів. Те ж саме стосується і нових версій операційних систем. Наприклад, на початку свого шляху Windows 10 викликала скандал через виявлені функції для збору "телеметрії". У наш час Spyware набирає обертів оскільки розробники вважають що такі збори інформації дозволять покращити їх продукт, тож аналогічні легальні шпигунські функції, чи то приховані, чи то відкриті, присутні практично у всіх сучасних операційних системах[24].

Фішинг. Цей спосіб обману часто відносять до соціальної інженерії, оскільки основою фішингу. Фішинг - це коли зловмисники надсилають шкідливі електронні листи, щоб обдурити людей і змусити їх стати жертвами шахрайства. Дуже часто їх метою є отримання фінансової інформації, облікових даних системи або інших конфіденційних даних.

Цей метод можна охарактеризувати як набір маніпуляцій і прийомів, які шахраї використовують для впливу на психологію людей. Методи соціальної інженерії, такі як підробка, наведення на хибний слід і обман, часто відіграють ключову роль у фішингових атаках. На базовому рівні фішингові електронні

листи використовують соціальну інженерію, щоб підштовхнути користувачів до дій, часто не замислюючись про наслідки.

Кіберзлочинці досить часто використовують фішингові атаки, оскільки це дешево, легко й ефективно. Отримати доступ до адрес електронної пошти легко, а надсилати електронні листи практично безкоштовно. Зловмисники вкладають мінімум зусиль і ресурсів, щоб швидко отримати доступ до цінної інформації. Люди, які стають жертвами фішингу, можуть зазнати впливу шкідливих програм, втрати особистих даних і важливої інформації.

Інформація, яку переслідують кіберзлочинці, охоплює особисті дані, наприклад інформацію про фінансові рахунки, номери кредитних карток, податкові та медичні записи, а також конфіденційні корпоративні дані, імена клієнтів, контактна інформація, відомості про продукти та конфіденційні повідомлення. Окрім того, кіберзлочинці використовують фішингові атаки для отримання прямого доступу до електронної пошти, соціальних мереж та інших облікових записів, а також для отримання дозволів на зміну і компрометацію пов'язаних систем, платіжних систем і систем обробки замовлень[25].

Атака людина посередник. При такому типі атак кіберзлочинець вступає в гру між вами і веб-сервісом, сайтом чи додатком. Він стає прихованим посередником, який може бачити і навіть змінювати інформацію, передану між вами і цим сервісом.

На перший погляд атака здається такою що не може завдати значних збитків, та насправді це не так. Припустимо, користувач намагається зайти на свій банківський акаунт або надіслати конфіденційне повідомлення. Зловмисник, використовуючи атаку "людина-посередник", може перехопити цю інформацію, навіть якщо вона зашифрована, використати цю інформацію в своїх цілях або навіть змінити її і відправити перероблене повідомлення від імені користувача. Крім того, такий вид атак дозволяє викрасти особисті дані, паролі або навіть гроші в той час коли користувач, навіть не помічає присутності зловмисника. Зазвичай таку атаку можна помітити коли стається

щось несподіване або помічається незвична активність на акаунті, але в цей момент вже занадто пізно.

Тому, важливо завжди бути обережним і використовувати надійні та захищені інтернет з'єднання, а також звертати увагу на підозрілі ознаки, такі як незаплановані запити на введення паролів або зміни у обліковому записі.

Отже, наведемо узагальнюючу таблицю для розглянутих типів атак:

Таблиця 1.1

Тип атаки	Опис	Мета атаки	Приклади
Атака "троянський кінь"	Зловмисник маскує шкідливе ПЗ як легітимне і переконує користувача встановити його, після чого воно може виконувати шкідливі дії.	Нелегітимний доступ до системи, встановлення прихованих програм, крадіжка даних.	Pinch, TDL-4, Trojan.Winlock, Clampi, WARRIOR PRIDE, Trojan.Win32 та Backdoor.Trojan.
DOS та DDoS атаки	Атаки, за яких зловмисник перевантажує цільову систему трафіком, роблячи її недоступною для легітимних користувачів.	Перевантаження системи, зниження продуктивності або створення недоступності.	Атаки, що використовують безліч запитів або ботнетів, щоб перевантажити сервер.
Віруси	Шкідливе програмне забезпечення, яке прикріплюється до файлів або програм і може реплікуватися, інфікуючи інші файли.	Пошкодження даних, крадіжка інформації, вимагання, нанесення шкоди системі.	Stuxnet, Conficker, PoisonIvy, ILOVEYOU, Зевс, Slammer, Code Red, Fizzer.

Продовження таблиці 1.1

Тип атаки	Опис	Мета атаки	Приклади
Spyware	Програмне забезпечення, що приховано встановлюється на пристрої користувача, збирає і передає зловмиснику інформацію про дії користувача.	Збір конфіденційних даних про користувача без його згоди.	Кейлоггери, програми-шпигуни для моніторингу активності користувача.
Фішингова атака	Зловмисники надсилають оманливі повідомлення, що видають себе за легітимні джерела, з метою обдурити користувачів і отримати їх конфіденційну інформацію.	Отримання особистих даних, таких як конфіденційна інформація, паролі, номери кредитних карт, тощо.	Підроблені електронні листи, веб-сайти та повідомлення.
Атака "людина-посередник"	Зловмисник вступає між користувачами та метою атаки, перехоплює та маніпулює даними.	Перехоплення конфіденційних даних і комунікації між користувачами та метою.	

1.2. Використання блокчейн технологій для забезпечення інформаційної безпеки.

Блокчейн - це особлива структура даних, яка використовується для створення розподіленої бази даних з відкритим доступом для різних незалежних один від одного учасників. Існує три основні типи блокчейна: публічний, приватний і гібридний [4]. Розглянемо кожен тип більш детально.

Публічний блокчейн - це відкритий і доступний для всіх тип блокчейна. Ці мережі іноді називають "безправними", оскільки в них немає центральної установи, яка регулює участь користувачів. У таких мережах будь-який охочий може стати учасником без будь-яких обмежень[3]. Важливо зазначити, що публічні блокчейни володіють високим ступенем безпеки і прозорості, оскільки всі транзакції є публічними і доступними для перевірки, проте такий тип блокчейну має обмежену пропускну здатність і масштабованість, оскільки може мати значну кількість учасників і вузлів. Схематично публічний блокчейн можна описати наступним чином:

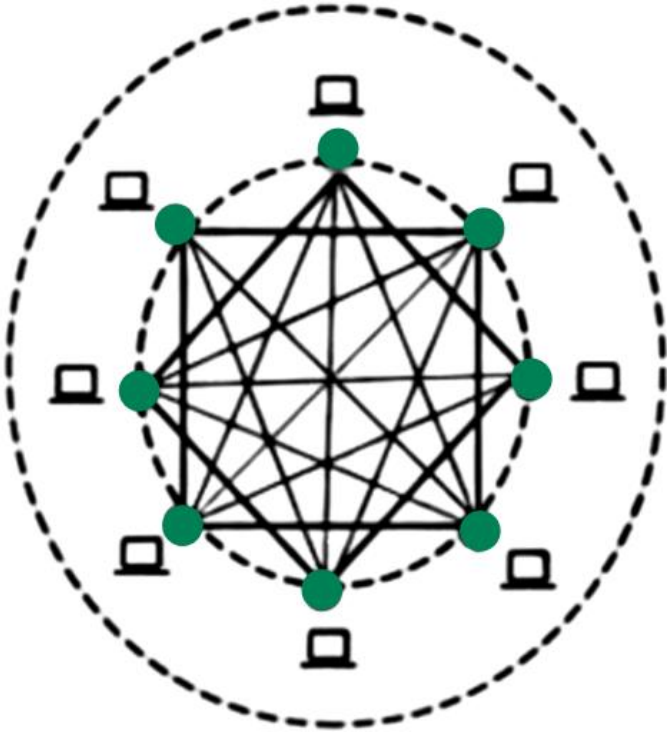


Рис.1.1. Публічний блокчейн

Приватний блокчейн - це тип блокчейна, обмежений жорстким списком учасників. Компанії часто використовують такі мережі для внутрішнього аудиту та контролю. Доступ до приватного блокчейну надається тільки певним користувачам, і центральна організація (як правило, компанія) контролює

створення і перевірку транзакцій[4]. Приватні блокчейни забезпечують вищу швидкість і масштабованість транзакцій, оскільки мають обмежену кількість вузлів, проте вони є найменш безпечним типом блокчейну, оскільки мають єдину точку відмови і централізоване управління. Схематично приватний блокчейн можна описати наступним чином:

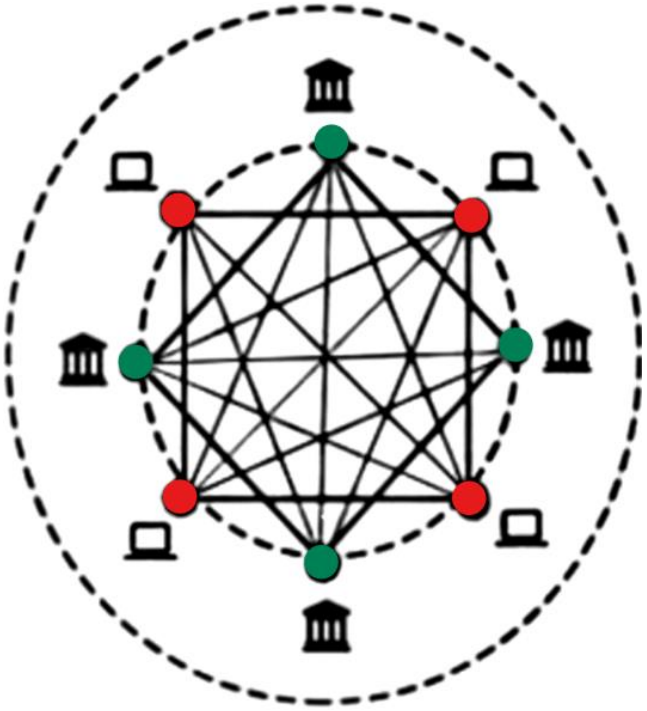


Рис.1.2. Приватний блокчейн

Гібридний блокчейн (або блокчейн-консорціум) - це суміш публічних і приватних блокчейнів. Цими мережами керують обрані вузли й об'єднують у собі елементи обох типів блокчейнів. Однією з ключових відмінностей є спосіб досягнення консенсусу в таких мережах. На відміну від публічних блокчейнів, де кожен може перевіряти блоки, і приватних, де встановлений єдиний валідатор, блокчейн-консорціуми розглядають кілька валідаторів рівнозначних сторін. Правила в блокчейн-консорціумі можуть бути гнучкими, оскільки доступ до перегляду вмісту блоків може бути обмежений тільки для уповноважених осіб або наданий усім без винятку, залежно від рішення учасників. За умови, що валідатори можуть досягти згоди між собою, процес внесення змін до системи стає простішим. У цій схемі роботи блокчейн мережі, система може продовжувати функціонувати навіть якщо лише певна частина

учасників діє чесно. Блокчейн-консорціуми найефективніші, коли кілька організацій працюють в одній галузі та потребують спільної платформи для проведення транзакцій або обміну інформацією[5]. Гібридні блокчейни часто використовують підвищену криптографію для забезпечення безпеки транзакцій і аудиту.

Схематично гібридний блокчейн можна описати наступним чином:

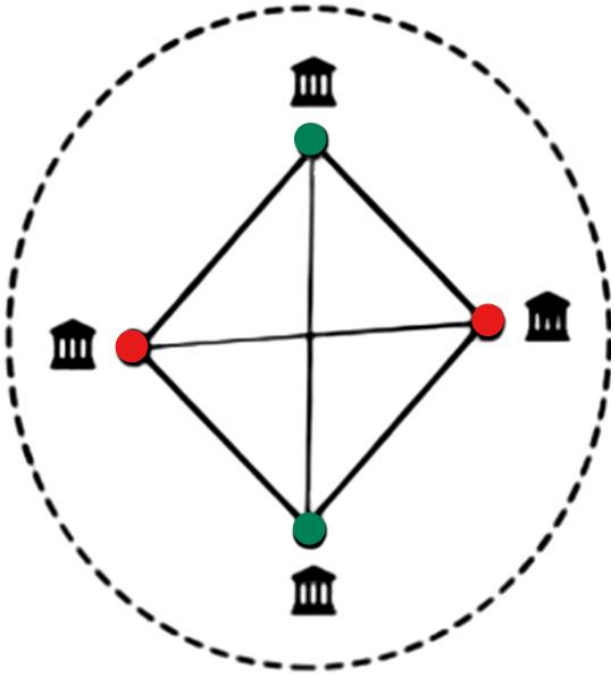


Рис.1.3. Гібридний блокчейн

Кожен із цих типів блокчейна має свої переваги та обмеження, і вибір залежить від конкретних потреб і цілей користувачів або організацій.

Узагальнимо порівняння розглянутих типів блокчейнів у таблиці 3.

Таблиця 1.2

Критерій порівняння	Публічний блокчейн	Приватний блокчейн	Гібридний блокчейн
Загальнодоступність	Так	Ні	Ні
Хто може переглядати інформацію?	Усі	Лише запрошені користувачі	Залежно від потреб мережі

Продовження таблиці 1.2

Хто може робити записи?	Усі	Лише затверджені учасники	Лише затверджені учасники
Критерій порівняння	Публічний блокчейн	Приватний блокчейн	Гібридний блокчейн
Хто володіє мережею?	Ніхто	Одна особа	Кілька осіб
Чи є інформація про користувачів доступною?	Ні	Так	Так
Швидкість транзакцій	Повільна	Швидка	Швидка

До кожного блоку, окрім самих записів та ідентифікатора блоку, також входять хеш-суми попереднього та поточного блоків. Вони є результатом обчислення криптографічних хеш-функцій[17]. Хеш-функції в блокчейні, у поєднанні з його розподіленою архітектурою, забезпечують незворотність і незмінність усього ланцюжка блоків і інформації, яка в них зберігається.

Окрім хеш-функцій важливу роль у блокчейні виконує набір математичних правил і функцій, його називають алгоритмом консенсусу. Головним завданням алгоритму консенсусу є безпосередньо генерація і подальша синхронізація ланцюга блоків у всіх учасників цієї мережі. База даних блокчейна зберігається у спеціальних вузлах мережі всіх учасників блокчейн-мережі, таких учасників може бути нескінченна кількість, такі вузли називають вузлами майнінгу або вузлами консенсусу. Також слід враховувати, що у більшості блокчейнів учасники мережі невідомі заздалегідь і можуть підключатися або відключатися в будь-який момент[18]. Алгоритм консенсусу допомагає досягти спільної угоди про поточний вигляд розподіленої бази даних з усіма учасниками мережі. Вузли консенсусу також проводять об’єднання

нової інформації у блоки і обчислення хеш-функцій поточного і попереднього блоків для захисту ланцюга блокчейну, за це зазвичай кожен вузол отримує винагороду у вигляді певного цифрового активу, до прикладу, монету криптовалюти[19]. Усі ці особливості, а також економічні мотиваційні механізми для учасників блокчейн-мереж, призвели до створення спеціальних алгоритмів консенсусу, які відрізняються від консенсусів, які використовували у розподілених систем до появи блокчейна.

Поєднання властивостей, які має розподілений реєстр з блоковою структурою даних, який ґрунтується на криптографічній зв'язаності, дозволяє блокчейну ефективно підтримувати цілісність, доступність і невідмовність інформації, три з п’яти основних критеріїв безпеки інформаційних мережі. Децентралізована топологія та криптографічні механізми роблять зловмисні маніпуляції даними вкрай складними та дорогими, а інформація залишається доступною для всіх учасників, навіть при значних змінах у розмірах блокчейн-мережі. Проте, традиційна модель публічного децентралізованого блокчейну, яка забезпечує прозорість та стійкість до цензури, не забезпечує третій аспект інформаційної безпеки - конфіденційність даних через свою архітектуру та ідеологію. Саме це і призвело до виникнення концепції приватного блокчейну [6].

Приватний блокчейн насамперед відрізняється від публічного блокчейну тим, що доступ до мережі і право внесення змін до реєстру обмежені і контролюються конкретними учасниками мережі. Зазвичай доступ до записів реєстру такої мережі також обмежений. Приватний блокчейн відрізняється від публічного і «світоглядно», оскільки в ньому з'являється оператор, що не дозволяє мережі лишатись повністю децентралізованою, а лише розподіленою. Однак приватний блокчейн дозволяє забезпечувати конфіденційність даних, оскільки доступ контролюється політиками безпеки. Такі мережі найчастіше використовуються для корпоративних та державних завдань.

Гібридний блокчейн, як було сказано раніше, поєднує обидва підходи. У такому типі блокчейну записи з приватної мережі або їх метадані можуть

додатково зберігатися в публічному блокчейні, що забезпечить додаткову відмовостійкість усього реєстру.

З погляду безпеки, блокчейн слід розглядати як інфраструктуру для конкретного сценарію, а не в якості окремої, самостійної технології, блокчейн мережу можна використовувати в якості бази даних для корпоративної інформаційної системи, середовища виконання децентралізованих додатків або смарт-контрактів, тощо. Аналіз багатьох інцидентів з інформаційної безпеки, пов'язаних із рішеннями на основі блокчейн, показує, що часто найбільш вразливою є не сама мережа блокчейну, а суміжні компоненти та інформаційні системи[7].

До переваг блокчейну відносять:

+ Розподіл. Завдяки зберіганню даних на тисячах пристроїв у розподіленій мережі, блокчейн стає дуже стійким до технічних збоїв і шкідливих атак. Кожен вузол мережі має власну копію бази даних, що забезпечує безпеку всієї системи. На відміну від звичайних баз даних, які часто залежать від одного або декількох серверів і схильні до збоїв, блокчейн набагато надійніший.

+ Стабільність. Підтверджені блоки в блокчейні рідко скасовуються, що робить зміну або видалення даних у ньому вкрай складним. Блокчейн добре підходить для зберігання фінансових даних і виступає в ролі надійного аудиторського журналу, де кожна зміна записується і захищається в загальнодоступному реєстрі. Це дає змогу компаніям запобігати шахрайству з боку співробітників.

+ Система, що не потребує довіри. Блокчейн усуває необхідність у посередниках, таких як банки або платіжні провайдери, оскільки розподілена мережа вузлів перевіряє транзакції через майнінг. Це робить блокчейн системою "без довіри", де немає ризику, пов'язаного з довірою до певних організацій, і знижуються комісії за транзакції, оскільки немає посередників, яким потрібно платити за роботу.[30]

Недоліки блокчейну:

– Атака 51%. Існує потенційна атака, під час якої зловмиснику вдається отримати контроль над більше ніж 50% обчислювальної потужності мережі блокчейна. Це може дозволити злочинцю змінювати блоки. Ця проблема вирішується зі збільшенням розміру мережі, чим більше користувачів, а отже й потужностей тим вища безпека даної мережі.

– Зміни даних. Після додавання даних у блокчейн їх складно змінити. Це є значною перевагою, проте внесення змін або виправлень у блокчейн потребує значних зусиль, іноді навіть створення нової версії блокчейну, що завдає певних незручностей.

– Приватні ключі. Блокчейн використовує криптографію з публічним і закритим ключами. Втрата закритого ключа може призвести до безповоротної втрати доступу до інформації прив’язаної до акаунту.

– Неефективність. Деякі блокчейни, особливо ті, що використовують алгоритм Proof of Work, вимагають величезних обсягів обчислювальних ресурсів та енергії для майнінгу. В деяких випадках це неефективне використання ресурсів, особливо у наш час, коли проблема екології постає так гостро.

– Зберігання. Розміри блокчейн-реєстрів можуть значно зростати з часом, що створює проблеми зі зберіганням і завантаженням для користувачів.

1.3. Огляд сучасних практичних підходів щодо управління паролями.

У наш час проблема надійності паролів є дуже актуальною, оскільки паролі використовуються в різних сферах нашого життя, від доступу до соціальних мереж до банківських рахунків. Ми доволі часто використовуємо недостатньо надійні паролі - паролі з невеликою кількістю символів, паролі, які

можна пов'язати з особистою інформацією, або ж використовуємо однакові паролі для різних ресурсів. Головною причиною таких дій є страху забути пароль і втратити доступ до ресурсу. Ті користувачі, які при створенні паролів дотримуються усіх канонів часто записують їх у записнику або текстовому файлі з секретною назвою "паролі" на робочому столі свого приватного комп'ютера.

Щоб розкрити ступінь гостроти проблеми надійності паролів, розглянемо дослідження, проведене компанією NordPass за період з 2020 по 2022 рік.

Під час цього дослідження вони у співпраці з незалежними дослідниками, які спеціалізуються на аналізі подій у галузі кібербезпеки щорічно аналізували список паролів. Того року їм вдалося оцінити базу даних об'ємом 3 ТБ. Для кожного року було визначено 200 найпопулярніших паролів та підраховано, скільки разів кожен з цих паролів використовувався. Нижче, в таблиці 3 наведено топ-10 паролів за результатами цього дослідження.

Таблиця 1.3

	2020		2021		2022	
	Пароль	Кіл-ть	Пароль	Кіл-ть	Пароль	Кіл-ть
1	123456	2,543,285	123456	103,170,552	password	4,929,113
2	123456789	961,435	123456789	46,027,530	123456	1,523,537
3	picture1	371,612	12345	32,955,431	123456789	413,056
4	password	360,467	qwerty	22,317,280	guest	376,417
5	12345678	322,187	password	20,958,297	qwerty	309,679

Продовження таблиці 1.3

	2020		2021		2022	
	Пароль	Кіл-ть	Пароль	Кіл-ть	Пароль	Кіл-ть
6	111111	230,507	12345678	14,745,771	12345678	284,946
7	123123	189,327	111111	13,354,149	111111	229,047
8	12345	188,268	123123	10,244,398	12345	188,602
9	1234567890	171,724	1234567890	9,646,621	col123456	140,505
10	senha	167,728	1234567	9,396,813	123123	127,762

Результати цього дослідження демонструють шокуючу популярність надзвичайно простих та слабких паролів серед користувачів. Всі перелічені в таблиці паролі не виконують своєї основної функції - захисту інформації від несанкціонованого доступу. Ці паролі є надзвичайно слабкими та ненадійними, оскільки їх можна легко зламати за дуже короткий час, нерідко менше ніж за хвилину [20].

Щоб пароль був надійним і забезпечував високий рівень безпеки, рекомендують дотримуватися наступних порад:

- використовувати складні та унікальні паролі. Такий пароль має містити різні символи, великі та маленькі літери, цифри та спеціальні символи (наприклад, !, @, #, $).
- не використовувати загальнодоступну особисту інформацію. До такої інформації відноситься ім’я, дата народження, номер телефону, та інша інформація, яку можна легко знайти в соціальних мережах.
- використовувати різні паролі для різних облікових записів. Використання одного пароля для всіх сервісів і облікових записів є вкрай

небезпечних, адже якщо зловмисник зможе скомпроментувати один пароль, він зможе отримати доступ до усіх ресурсів і облікових записів.

- використовувати багатофакторну автентифікацію. Не усі сервіси підтримують таку функцію, проте її слід використовувати усюди де є така змога. Багатофакторна автентифікація додає додатковий шар безпеки, такий як додатковий SMS-код, пін-код або відбиток пальця.

Усі ці пункти здаються для користувачів дуже складними, адже потрібно тримати в голові десятки наборів символів. Для збереження усіх паролів в одному зручному і надійно захищеному місці існують спеціальні менеджери паролів, які дозволяють користувачу запам’ятати лише один пароль який дає доступ до додатку [21].

Менеджери паролів - це програми або додатки, які допомагають створювати, зберігати та автоматично заповнювати паролі для різних облікових записів, веб сторінок, програм, тощо. Основна перевага менеджерів паролів полягає в тому, що вони дозволяють мати унікальний і складний пароль для кожного облікового запису не запам’ятовуючи їх.

Розглянемо деякі найпопулярніші менеджери паролів[22]:

Enpass - це програма для управління паролями з дуже зручним інтерфейсом. Під час використання Enpass усі паролі зберігаються локально на комп'ютері користувача, без передавання секретної інформації через мережу. За бажання користувача додаток дозволяє використовувати хмарне сховище, що не лише звільнить простір вашого пристрою, але й дозволить синхронізувати інформацію з різних девайсів. Enpass також аналізує надійність паролів і повідомляє користувача, якщо він використовуєте однакові паролі для різних сервісів. Крім того, в програмі можна також зберігати дані кредитних карток і особисті документи.

В Enpass існує платна преміум версія, яка дозволяє використовувати цей менеджер паролів на кількох пристроях одночасно. Вартість такої версії на момент написання становить 75 гривень.

Підтримується на таких платформах: Windows, Android, iOS та macOS.

NordPass додаток, який не лише допоможе у збереженні паролів, а й може згенерувати надійний пароль, а також перевіряє чи не були ваші облікові записи скомпрометовані. У цьому менеджері паролів також присутня інтеграція в найпопулярніші браузери з можливістю автозаповнення паролей. Користувачі можуть отримати доступ до своїх паролів за допомогою кодової фрази або спеціального додатка для двофакторної аутентифікації.

NordPass можна використовувати безкоштовно, проте він має платну преміум версію, яка дозволяє, наприклад, користуватись одночасно кількома пристроями. Платна версія програми на час написання коштує 55 грн за місяць.

Підтримується на таких платформах: Windows, Android, iOS, macOS, Linux.

Keeper - це менеджер паролів, який зберігає усі паролі на своїх віддалених серверах, проте секретні ключі для розшифровки цих паролей знаходяться лише на приватному комп'ютері користувача, якому ці паролі належать. Окрім майстер-пароля, програма підтримує двофакторну аутентифікацію через SMS або аутентифікатор Google, а також може використовувати біометричний захист. Крім цього Keeper також може провести оцінку надійності облікових записів і пропонує заміну слабких паролів більш надійними.

В цій програмі доступна виключно платна версія, на момент написання вартість її використання становить 110 гривень.

Підтримується на таких платформах: Windows, Android, iOS, macOS, та Linux.

1Password - це багатофункціональний інструмент для збереження і захисту конфіденційних даних. Він не лише дозволяє зберігати паролі, а й підтримує багатофакторну автентифікацію за допомогою біометричного сенсора або мобільного телефону. Окрім цього, програма дозволяє зберігати не лише паролі, а і документи або банківські карти користувачів. Додаток може працювати навіть без Інтернет з'єднання, а також має можливість створювати список довірених контактів, з якими можна обмінюватися обраними паролями.

Важливо зазначити, що 1Password не має безкоштовної версії, проте він пропонує безкоштовний пробний період. На момент написання вартість користування становить 110 гривень на місяць.

Підтримується на таких платформах як Windows, Android, iOS та macOS.

KeePass - це менеджер паролів, який дозволяє зберігати паролі у вигляді зашифрованого файлу на персональному комп'ютері користувача, на будь якому холодному носії або ж в будь якому хмарному сховищі. Важливою особливістю KeePass є те, що він має відкритий вихідний код, що робить його прозорим для користувачів і дозволяє фахівцям з кібербезпеки перевіряти програму на наявність можливого шкідливого коду.

Програма KeePass є абсолютно безкоштовною.

Підтримується на таких платформах: Windows, Android, iOS, macOS та Linux.

Підсумуємо результат огляду найпопулярніших на даний момент менеджерів паролів за кількома критеріями та оцінимо кожен з них від 0 до 5, Результат наведено в таблиці 2.

Таблиця 1.4

Критерії оцінювання	Enpass	NordPass	Keeper	1Password	KeePass
Додаткове зберігання карток чи документів	5	0	0	5	0
Прозорість	4	3	4	3	5
Вартість повної версії	3	4	2	1	5
Мультиплатформеність	4	5	5	4	5

Варто зазначити, що розглянуті застосунки є одними з найкращих та найпопулярніших на ринку. Усі запропоновані програми мають зручний та інтуїтивно зрозумілий для більшості людей інтерфейс, проте для нас найважливійшим показником є захист інформації та можливість мережі відповідати основним критеріям інформаційної безпеки. Розглянуті застосунки

намагаються бути прозорими за рахунок збереження інформації лише на пристрої користувача, проте це майже не додає захисту вашій інформації а натомість робить їх використання менш зручним. У наш час рідко можна зустріти людину, яка використовує тільки один пристрій, ми маємо не лише персональні телефон, планшет і комп'ютер, а й робочі. Саме через це в менеджерах паролів існує проблема збереження надцінної інформації так щоб була можливість не лише надійно зберегти пароль, а й надати користувачу доступ до нього з будь якого пристрою у будь який час.

1.4. Висновки до першого розділу.

У цьому розділі ми розглянули основні проблеми щодо забезпечення безпеки інформаційних мереж, визначили найпопулярніші загрози та види атак, розглянули фактори які послабляють системи захисту, визначили основні критерії яким повинна відповідати захищена мережа. Ми бачимо наскільки насправді важливо на сьогоднішній день мати грамотно побудовану систему захисту інформаційної мережі.

У наш час питання інтернет гігієни піднімається доволі часто, проте це нічого не змінює, люди продовжують використовувати неякісні паролі з року в рік і ігнорують усі рекомендації. Хтось не довіряє менеджерам паролей оскільки вважає їх недостатньо безпечними адже на даний момент немає менеджера паролей який можна використовувати на кількох пристроях і в той самий час зберігати інформацію не на сервері компанії чи хмарному сховищі, а в надійно захищеному місці. Саме цю проблему можна вирішити з використанням блокчейн технологій. Публічна мережа blockchain використовує розподілену базу даних, що дозволяє децентралізувати процес збереження та обробки інформації. Використання blockchain вирішить питання довіри

користувачів, а також дозволить використовувати збережену інформацію з будь якого пристрою в будь якій точці земної кулі.

РОЗДІЛ 2. ДОСЛІДЖЕННЯ ОСНОВНИХ ХАРАКТЕРИСТИК BLOCKCHAIN ТЕХНОЛОГІЙ

2.1 Фундаментальні складові технології blockchain.

Технологія блокчейн базується на кількох фундаментальних принципах і компонентах, які разом забезпечують її унікальні властивості. До основних складових блокчейну належать:

- Ланцюжок блоків (Chain of Blocks). Блокчейн складається з послідовності блоків, кожен з яких містить набір транзакцій. Кожен новий блок у ланцюжку містить хеш попереднього блоку, що забезпечує цілісність і безперервність даних.
- Розподілений реєстр (Distributed Ledger). Блокчейн є розподіленим реєстром, у якому записи зберігаються на безлічі комп'ютерів по всьому світу. Це означає, що немає єдиної точки відмови і дані не можуть бути легко підроблені або змінені.
- Криптографічне хешування. Блоки в блокчейні захищені за допомогою криптографічного хешування. Це забезпечує безпеку даних і гарантує, що одного разу записані дані не можуть бути змінені без зміни всіх наступних блоків.
- Консенсусний алгоритм. Для валідації нових блоків і підтримки узгодженості даних між вузлами мережі використовуються алгоритми консенсусу, такі як Proof of Work (PoW), Proof of Stake (PoS) або інші варіанти.
- Децентралізація. Замість того щоб покладатися на центральний орган або посередника, блокчейн розподіляє контроль і управління між усіма учасниками мережі.

- Прозорість і анонімність. Не зважаючи на те, що всі транзакції в блокчейні є публічними і такими, що перевіряються, особистості учасників можуть залишатися анонімними або псевдоанонімними.
- Незмінність. Записи в блокчейні не можуть бути змінені після їх включення в ланцюжок. Це забезпечує високий рівень безпеки даних.[21]

Розглянемо найважливіші з цих компонентів більш детально

Ланцюжок блоків, є ключовим компонентом технології блокчейн. Він являє собою послідовність блоків, кожен з яких містить дані і є пов'язаним з попереднім блоком. Ця структура забезпечує унікальні властивості блокчейна, такі як прозорість, безпека та немутабельність (незмінність). Схематичне зображення ланцюжку блоків представлене на рисунку 2.1.

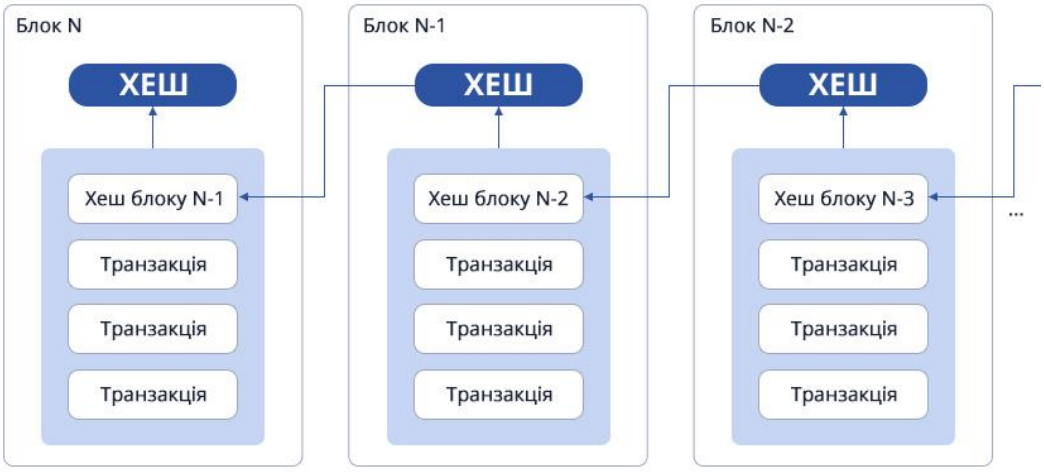


Рис.2.1 Схематичне представлення блокчейн ланцюжку.

Кожен блок у ланцюжку блоків зазвичай містить такі елементи:

- Хеш-значення попереднього блоку. Це забезпечує зв'язок між блоками, формуючи ланцюжок. Хеш попереднього блоку включається в новий блок, тож будь-яка зміна в попередньому блоці призведе до зміни хеша і порушення ланцюга.
- Набір транзакцій. Кожен блок містить список транзакцій, які були підтверджені і додані в блокчейн. У разі криптовалют, таких як Bitcoin, це

фінансові транзакції. В інших застосуваннях блокчейна це можуть бути будь-які види даних.

- Часова мітка. Час створення блоку.
- Хеш блоку. Унікальний ідентифікатор блоку, який генерується на основі його вмісту. Будь-яка зміна в блоці змінить його хеш, роблячи зміни легко помітними.
- Nonce. Це число, яке блокчейн-мережі (особливо в Proof of Work системах) використовують для знаходження підходящого хеша блоку, таке число використовується лише один раз.[26]

Принцип формування блокчейн ланцюга складається з наступних етапів:

- Створення блоку. Коли транзакції ініціюються, їх перевіряють вузли мережі і збирають у блок.
- Майнінг (для Proof of Work). Майнери використовують обчислювальну потужність для розв'язання складної криптографічної задачі, що включає пошук nonce, який під час хешування разом із даними блоку дасть хеш, що відповідає певним критеріям (наприклад, певна кількість нулів на початку). Цей процес називається майнінгом.
- Додавання в ланцюжок. Щойно блок майниться (або підтверджується в системах без Proof of Work), його хеш обчислюють і блок додають у ланцюжок. Кожен новий блок посилається на хеш попереднього блоку, що утворює незмінний ланцюжок.
- Підтвердження. Після додавання блоку в ланцюжок, його вміст не може бути змінено без зміни всіх наступних блоків і без досягнення консенсусу в мережі, що практично неможливо у великих блокчейн-мережах.

Ланцюжок блоків надає надійний і прозорий запис усіх транзакцій, який доступний усім учасникам мережі[40]. Це забезпечує цілісність даних і запобігає спробам подвійної витрати, шахрайства і тамперінгу (несанкціонованої зміни даних). Завдяки цим властивостям блокчейн знаходить

застосування в безлічі сфер, від фінансових послуг і ланцюжків поставок до голосувань і управління ідентифікаційними даними.

Розподілений реєстр (Distributed Ledger Technology, DLT) є ключовою основою технології блокчейн. Саме ця технологія, дає змогу записувати, розділяти і синхронізувати транзакції в різних місцях розташування без централізованого зберігання даних або адміністрації.[28] Розподілений реєстр можна описати як базу даних, яка фізично розподілена по декількох вузлах або місцях розташування, які можуть бути на різних географічних територіях і управляються різними учасниками. На рисунку 2.2 представлене порівняння централізованого та розподіленого реєстрів де зліва зображено централізований, а справа розподілений реєстри.

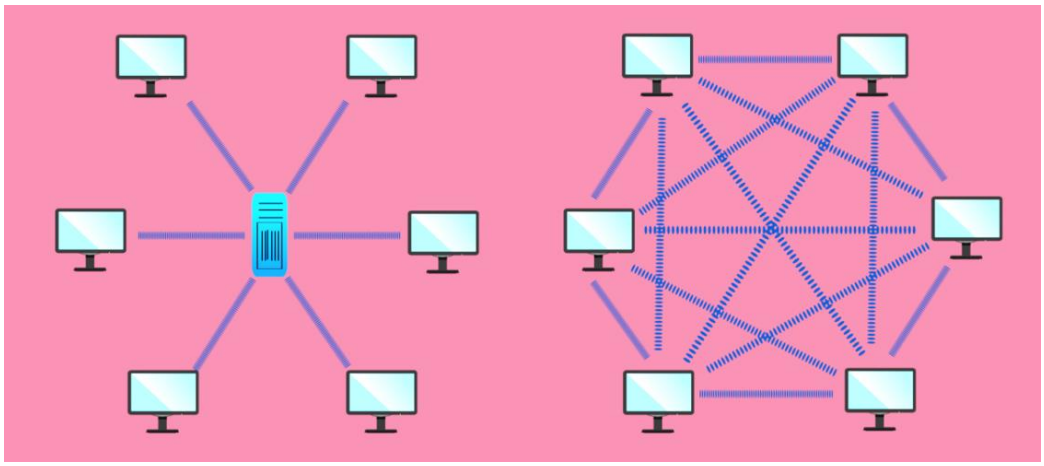


Рис.2.1 Централізований та розподілений реєстри

Проведемо порівняльний аналіз систем які побудовані на часто використовуваному централізованому реєстрі та на розподіленому реєстрі

Безпека. З точки зору безпеки обидві системи мають свої сторони. Децентралізовані системи, завдяки своїй конструкції, не мають єдиної точки відмови, що робить їх менш уразливими для цільових атак. Розподілені системи також забезпечують підвищену безпеку з огляду на їхню надмірність. Однак, якщо розподілена система працює під управлінням центрального органу, вона може бути більш вразливою для цільових атак.

Ефективність і продуктивність. Розподілені системи часто мають перевагу з точки зору ефективності та продуктивності завдяки їх здатності розділяти й обробляти завдання одночасно на різних вузлах. Децентралізовані системи, хоча й стійкі та надійні, іноді можуть бути менш ефективними через необхідність консенсусу в мережі, що може уповільнити час транзакцій.

Масштабованість. Цей пункт є проблемним для обох систем. Не зважаючи на те, що розподілені системи можуть впоратися з підвищеним навантаженням за рахунок додавання додаткових вузлів, координація цих вузлів стає більш складною. У децентралізованих системах у міру збільшення кількості вузлів, досягнення консенсусу може займати більше часу, що впливатиме на масштабованість.

Прозорість і довіра. У сценаріях, де довіра і прозорість мають першорядне значення, децентралізовані системи значно виграють. Без центрального органу, що контролює дані, користувачі можуть бути впевнені в чесності системи. З іншого боку, розподілені системи, особливо ті, що працюють під управлінням центрального органу, можуть зіткнутися з труднощами під час встановлення довіри, оскільки об'єкт, що контролює, потенційно може маніпулювати даними.

Взаємодія. Інтероперабельність, здатність різних систем взаємодіяти та обмінюватися інформацією, є ще одним важливим фактором під час порівняння. Децентралізовані блокчейни, призначені для забезпечення взаємодії між ланцюжками, підвищуючи загальну корисність екосистеми блокчейнів. З іншого боку, розподілені системи, особливо приватні, можуть зіткнутися з проблемами функціональної сумісності з огляду на їх ізольований характер і потребу в контрольованому доступі[29].

Основними характеристиками розподіленого реєстру є:

- Децентралізація. На відміну від традиційних баз даних, керованих однією організацією, розподілений реєстр не має центрального адміністратора або централізованого сховища даних. Це означає, що записи зберігаються на безлічі вузлів, роблячи систему більш стійкою до збоїв і атак.

- Прозорість. Зміни в реєстрі можуть бути переглянуті всіма учасниками мережі, що забезпечує високий рівень прозорості. У блокчейні, наприклад, кожна транзакція доступна для перегляду всіма учасниками мережі.
- Немутабельність. Щойно запис додано до реєстру, його не можна змінити або видалити без зміни всіх наступних записів і без згоди більшості вузлів у мережі. Це забезпечує захист від підробок і шахрайства.
- Консенсус. У DLT механізми консенсусу використовуються для затвердження нових записів у реєстрі. Це гарантує, що всі копії реєстру синхронізовані та узгоджені.
- Стійкість до збоїв. Розподілений реєстр має високу стійкість до технічних збоїв і зловмисних атак. Якщо один вузол виходить з ладу, це не впливає на доступність або цілісність іншої частини реєстру.

Криптографічне хешування відіграє центральну роль у технології блокчейн. Це метод перетворення вхідних даних (незалежно від їх розміру) в унікальний рядок фіксованої довжини, який діє як односторонній відбиток вихідних даних. Це означає, що за хешем не можна визначити вихідні дані, а будь-яка зміна у вихідних даних призведе до зовсім іншого хешу.

Основні характеристики криптографічного хешування:

- Детермінованість. Однакові вхідні дані завжди призводять до одного і того ж хешу.
- Швидке обчислення. Хеш для будь-яких даних можна обчислити швидко.
- Незворотність. Неможливо відновити вихідні дані за їхнім хешем (односторонній процес).
- Стійкість до колізій. Дуже малоймовірно, що два різні набори вхідних даних дадуть однаковий хеш.
- Чутливість до змін Навіть незначна зміна у вихідних даних призведе до зовсім іншого хешу.

Застосування в блокчейн:

Створення ланцюжка блоків. Кожен блок у блокчейні містить хеш попереднього блоку, що створює ланцюжок. Якщо дані в одному блоці змінюються, його хеш зміниться, і це порушить увесь ланцюжок, роблячи зміни такими, що легко виявляються.

Безпека транзакцій. Транзакції хешуються і включаються в блоки, що забезпечує їхню безпеку і немутабельність.

Доказ роботи (Proof of Work). У системах, що використовують Proof of Work, для створення нового блоку майнери мають знайти хеш, що відповідає певним вимогам (наприклад, певна кількість нулів на початку). Це вимагає значних обчислювальних зусиль.

Найпопулярнішими в блокчейн є алгоритми SHA:

Розглянемо найпростіший з них, це алгоритм SHA-1, або Secure Hash Algorithm 1, це метод створення криптографічного хеша, що використовує алгоритм стиснення. Цей процес включає в себе обробку блоків вхідного повідомлення фіксованої довжини і попереднього виходу блоку, генеруючи хеш-значення всіх блоків. Підсумковий хеш усього повідомлення відповідатиме результату останнього блоку.

Процес роботи SHA-1 виглядає наступним чином:

- Підготовка повідомлення. Вхідне повідомлення ділиться на блоки певної довжини, що позначаються як W. Останній блок доповнюється, спочатку додається один біт, потім нулі, і в кінці - довжина всього повідомлення. Якщо довжина повідомлення не поміщається в останній блок, додається ще один блок, заповнений за тим самим принципом.
- Ініціалізація змінних. Починаємо з п'яти шістнадцяткових змінних: H0, H1, H2, H3, H4, які будуть використовуватися в хеш-функції.
- Обробка блоків W. Кожен блок W ділиться на 16 рівних частин w, які потім складаються за модулем 2. Таким чином, блок набуває вигляду: $W(k) = w(0) \oplus w(1) \oplus .... \oplus w$.

- Циклічний зсув вліво. У кожному блоці проводиться лівосторонній циклічний зсув кожного біта даних. Схематичне зображення зсуву вліво представлене на рисунку 2.3

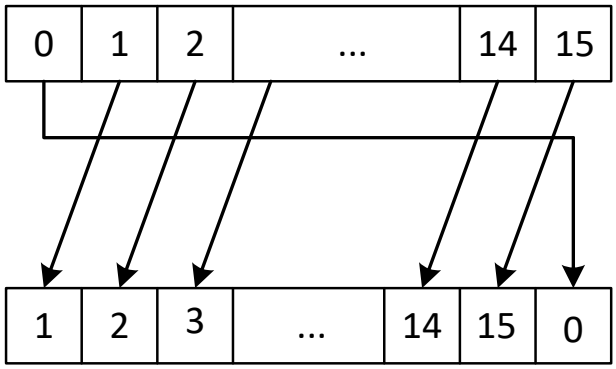


Рис.2.3. Зсув вліво

- 80 ітерацій хешування. У SHA-1 проводиться 80 ітерацій, що позначаються як i, де $0 \leq i \leq 79$. На кожній ітерації виконуються певні операції залежно від значення i.

- Фінальні обчислення. Після кожної ітерації змінні H оновлюються, унаслідок чого отримують підсумковий хеш-код H, який є комбінацією кінцевих значень змінних H0, H1, H2, H3 і H4 з їх початковими значеннями, зсунутими на певну кількість біт.

2.2 Принцип формування генезис блоку.

Генезис-блок у технології блокчейн є першим блоком будь-якої блокчейн-мережі та відіграє ключову роль у її структурі та функціонуванні. Створення генезис-блоку має кілька особливостей і принципів.

Генезис-блок є першим блоком у ланцюжку і не має попередника. На відміну від усіх наступних блоків, генезис-блок не має посилання на попередній блок (адже його ще не існує).

Як і будь-який інший блок у блокчейні, генезис-блок містить структуровані дані, зазвичай це часова мітка створення, транзакції (якщо вони застосовні) і хеш. Важливо зазначити, що у випадку генезис-блока, ці дані задаються вручну при створенні блокчейна і залежать лише від задуму розробника.

Ініціалізація мережі генезис-блок слугує відправною точкою для блокчейн-мережі. Він використовується для ініціалізації системи і початку додавання нових блоків до ланцюжка. Генезис-блок зазвичай жорстко закодований у вихідному коді блокчейн-платформи. А отже його дані не генеруються динамічно, а задаються розробниками при створенні програмного забезпечення блокчейна. Усі наступні блоки в блокчейні посилаються на попередній блок, створюючи ланцюжок, який веде назад до генезис-блока. Це забезпечує цілісність і безперервність ланцюжка.

У деяких блокчейнах генезис-блок не містить реальних транзакцій, за винятком можливих початкових параметрів або символічних повідомлень.

В процесі валідації блоків учасники мережі можуть використовувати генезис-блок як відправну точку для підтвердження легітимності всього ланцюжка блоків.

Створення генезис-блока - це фундаментальний крок у запуску блокчейн-мережі. Він лежить в основі архітектури блокчейна і відіграє центральну роль у забезпеченні його цілісності та безпеки.

2.3 Масштабування blockchain.

Блокчейн являє собою революційну технологію сучасності, але стикається з серйозними проблемами масштабованості. Ці труднощі пов'язані з особливостями архітектури блокчейна і є предметом активних досліджень як в академічних колах, так і серед передових компаній[35]. У практичному

застосуванні блокчейн часто не використовується для заміщення традиційних валют через його обмежену масштабованість, наприклад, під час оплати дрібних покупок, таких як кава, через тривалий час опрацювання транзакцій. Саме через це біткоїн найчастіше слугує активом для інвестицій.

Основна мета таких алгоритмів як Proof of Work або Proof of Stake - забезпечення стійкості системи. У блокчейні кожен вузол, особливо повні вузли, підтримує актуальну копію всього ланцюжка блоків, перевіряє транзакції та блоки, а також обробляє запити від інших вузлів. Це все сприяє децентралізації, але водночас є перешкодою для масштабування. У централізованих системах масштабування досягається за рахунок додавання більшої кількості серверів, проте в децентралізованих мережах збільшення кількості вузлів призводить до збільшення затримок[29]. Хоча збільшення кількості вузлів підвищує рівень децентралізації, це також призводить до зростання вимог до обчислювальних і сховищних ресурсів. Варто зазначити, що ця проблема більш актуальна для публічних блокчейнів, тоді як приватні блокчейни масштабуються легше, завдяки більш оптимальному розподілу навантаження на потужні вузли.

Важливо розуміти, що не всі методи масштабування підходять для всіх сценаріїв використання блокчейна. Найкращий підхід - це ретельно вивчити всі доступні техніки і вибрати найбільш підходящий метод для конкретної ситуації.

Масштабованість блокчейна - це ключове питання, яке стримує широке поширення цієї технології. Ось кілька додаткових аспектів і рішень, пов'язаних із масштабованістю блокчейна:

Основні Проблеми Масштабованості:

- Пропускна здатність. Одна з основних проблем блокчейна - це обмежена пропускна здатність транзакцій. Наприклад, Bitcoin може обробляти лише близько 7 транзакцій на секунду, тоді як Ethereum - близько 15-30. Це значно менше порівняно з традиційними платіжними системами.

- Затримка транзакцій. Час, необхідний для підтвердження транзакцій, може бути досить великим, що робить блокчейн непрактичним для операцій, які потребують швидкого виконання.
- Масштабування Мережі. Як більше людей починають використовувати блокчейн, мережеве навантаження збільшується, що може призвести до збільшення часу обробки транзакцій і підвищення комісій.

Можливі рішення щодо масштабованості блокчейну:

- Сегрегований Свідок (SegWit). Це одне з технічних рішень, запропонованих для Bitcoin, яке збільшує розмір блоку, видаляючи дані підпису з основної частини транзакції.
- Lightning Network і Сайдчейни. Lightning Network для Bitcoin і аналогічні рішення для інших криптовалют пропонують механізм проведення транзакцій поза основним блокчейном, що значно збільшує швидкість транзакцій і знижує комісії.
- Sharding (Шардування). Цей підхід полягає в поділі блокчейна на кілька сегментів (шардів), кожен з яких може обробляти транзакції паралельно. Це дає змогу значно збільшити загальну пропускну здатність системи.
- Proof of Stake (PoS). Перехід від Proof of Work до Proof of Stake може допомогти поліпшити масштабованість, оскільки PoS зазвичай вимагає менше обчислювальних ресурсів.
- Оптимізація Протоколів. Впровадження нових і поліпшених алгоритмів консенсусу та оптимізація наявних може допомогти збільшити швидкість і ефективність блокчейна.
- Блокчейни Нового Покоління. Розробка нових блокчейнів, які спочатку створюються з урахуванням масштабованості, таких як Polkadot, Cardano і Solana, пропонують нові архітектури та підходи до обробки транзакцій.

Обчислення поза блокчейном є важливим методом масштабування блокчейн-технологій. Основна ідея полягає в тому, щоб знизити навантаження

на блокчейн, проводячи основну частину обробки даних за його межами, а в сам блокчейн записувати тільки підсумкові результати. Підходи до реалізації таких обчислень варіюються залежно від специфіки завдання і використовуваного типу блокчейна[28].

Ця концепція вводить додатковий шар, який бере на себе більшість вимогливих до ресурсів операцій, використовуючи блокчейн лише для ключових функцій. При цьому важливо розуміти, що обчислення поза ланцюжком не завжди можуть забезпечити всі характеристики блокчейна, але це і не завжди необхідно, адже блокчейн може використовуватися лише в критично важливих моментах системи.

Одним зі способів реалізації таких обчислень є бічні ланцюжки (sidechains), які функціонують незалежно від основного блокчейна. Це не тільки полегшує масштабування, а й запобігає розповсюдженню можливих проблем з бічного ланцюжка в основний блокчейн[24]. Прикладом такого бічного ланцюжка для Bitcoin є Lightning Network, призначена для прискорення транзакцій з низькою комісією, що особливо актуально для мікроплатежів. Ще один приклад - RSK, який фокусується не на масштабованості, а на конфіденційності.

У контексті перевірки автентичності транзакцій, виконаних поза блокчейном, ключовим елементом є використання асиметричної криптографії, застосовуваної в блокчейні. Для здійснення транзакції необхідно володіти особистим ключем і підписати її. Такі транзакції стають захищеними і такими, що перевіряються, з моменту їхнього запису в блокчейн[25].

Блокчейни типу Bitcoin не підтримують баланс рахунку в традиційному розумінні - вони оперують транзакціями без явного відображення остаточного балансу. На відміну від них, блокчейн Ethereum містить інформацію про баланси рахунків. Це призводить до того, що кожен вузол у мережі повинен підтримувати значний обсяг даних.

Для зниження навантаження на блокчейн і поліпшення масштабованості застосовуються канали стану, які дозволяють оновлювати блокчейн на основі

кінцевого результату взаємодій, а не кожної окремої транзакції. Схематичне зображення Канали стану для обчислення поза блокчейном наведено на рисунку 2.4 Ці канали являють собою захищені криптографічні канали зв'язку, що дозволяють користувачам або системам обмінюватися даними. Обчислення в каналах стану зазвичай приватні й обмежені певною групою учасників, і тільки остаточні результати транзакцій фіксуються в блокчейні.

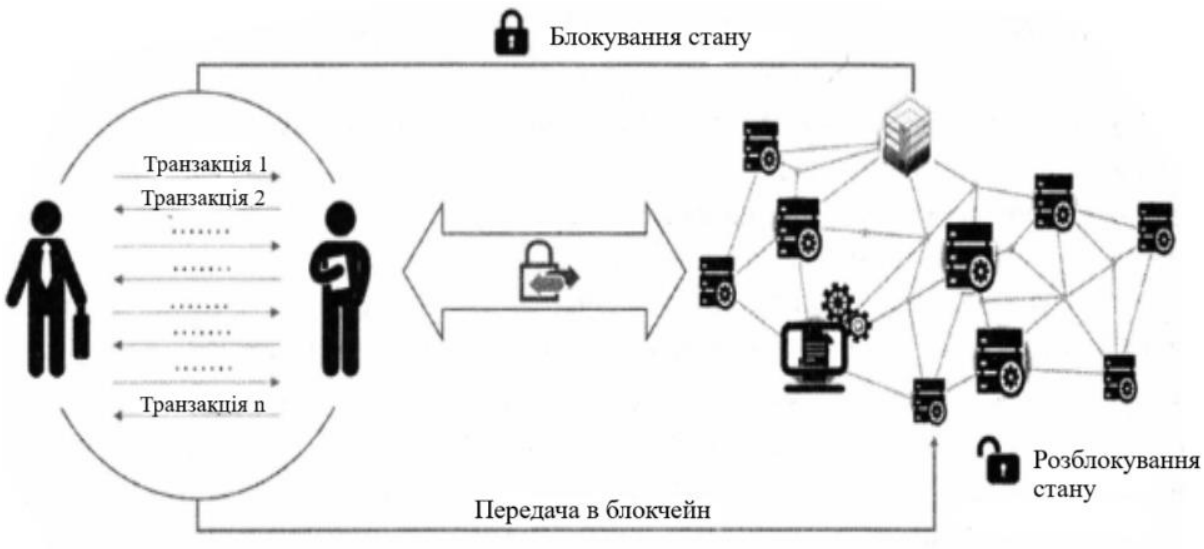


Рис.2.4. Канали стану для обчислення поза блокчейном

Шардинг, або сегментування, є однією з методик масштабування, яку використовували протягом багатьох років, особливо у сфері баз даних. Ця технологія знаходить різні застосування для вирішення проблем масштабування. Для розуміння застосування шардингу в контексті блокчейна важливо спочатку розібратися в його суті[27].

Традиційно, читання і запис на диск були обмеженням у роботі з великими обсягами даних. Шардинг дає змогу розподіляти дані по різних дисках, що забезпечує можливість паралельної роботи і знижує загальний час затримки. Це досягається шляхом поділу бази даних на менші, більш керовані частини, або шарди.

Розглянемо приклад поділу таблиці бази даних об'ємом 300 ГБ на три шарди по 100 ГБ кожен, що зберігаються на різних серверах (рис.2.5). Ця

концепція також може бути застосована до блокчейну, де загальний стан блокчейну ділиться на окремі сегменти, кожен з яких зберігає свій підстан.[27] Однак варто зазначити, що шардинг у блокчейні набагато складніший, ніж простий горизонтальний поділ бази даних.

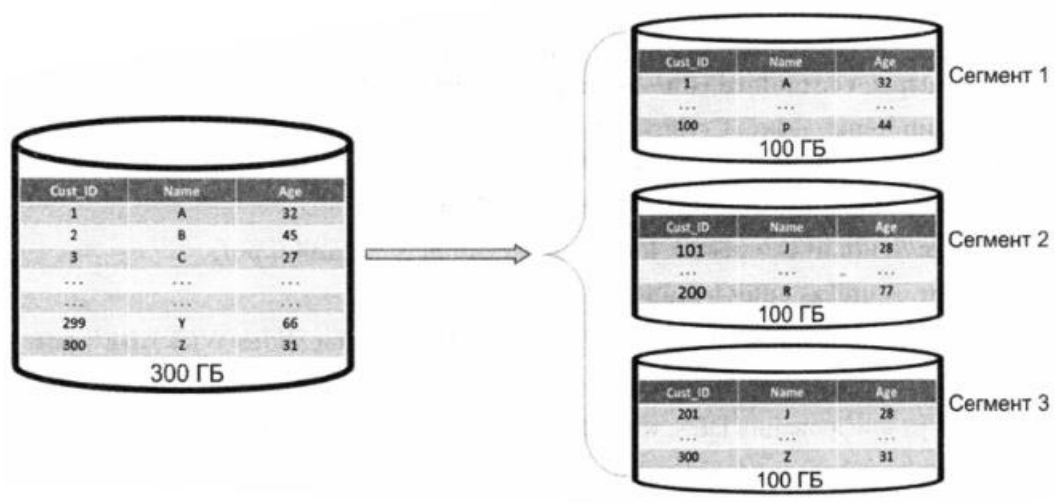


Рис.2.5 Шардинг бази даних

В блокчейні шардинг передбачає, що вузлам не потрібно завантажувати і зберігати копію всього ланцюжка блоків. Замість цього вони працюють тільки з тими частинами (шардами), які їм необхідні. Це дає змогу вузлам економити ресурси і підвищує загальну продуктивність системи, розподіляючи навантаження і збільшуючи пропускну здатність мережі. Вузли працюють тільки з транзакціями, які пов'язані з даними, що зберігаються в їхніх сегментах. Це дає змогу здійснювати транзакції паралельно, значно збільшуючи ефективність системи. Коли відбувається транзакція, вона спрямовується виключно на ті вузли, які відповідають за обслуговування відповідних шардів. Таким чином, не кожному вузлу необхідно обробляти всі транзакції мережі, що знижує загальне навантаження.

Однак у разі транзакцій, що зачіпають дані в декількох шардах, виникає необхідність у розробленні спеціального протоколу і механізму для обміну даними між цими шардами. Важливо зазначити, що різні блокчейн-системи

можуть реалізовувати шардинг по-різному, залежно від своїх особливостей і вимог.

Як приклад шардингу в блокчейні можна навести систему, де в різних шардах розташовуються різні унікальні облікові записи[32]. Це особливо актуально для блокчейнів, подібних до Ethereum, які підтримують стан рахунків. У такому разі, проведення транзакцій усередині одного шарду між обліковими записами є відносно простим процесом. Однак для ефективної роботи шардингу необхідний додатковий координаційний шар, що знаходиться вище рівня шардів, оскільки кожен вузол у системі зберігає лише частину загальних даних.

Масштабованість блокчейна залишається активною сферою досліджень і розробок[31]. Вирішення проблеми масштабованості матимуть вирішальне значення для майбутнього блокчейна і його здатності революціонізувати різні галузі, від фінансів до логістики і за їх межами.

2.4 Властивості технології blockchain та її використання.

Блокчейн з самого відкриття еволюціонує і зараз він став ключовим елементом у сучасних технологічних рішеннях. Блокчейн не просто технологія, це революція, яка змінює основні принципи ведення бізнесу та взаємодії в мережі.

Інтернет 1990-х років радикально трансформував бізнес і комунікацію, прибравши бар'єри для поширення інформації та відкривши нові ринки.[29] Аналогічним чином, блокчейн є наступним кроком у розвитку Інтернету, долаючи перешкоди в трьох основних аспектах: управлінні, довірі та передачі цінності:

- Управління. Блокчейн децентралізує управління, що дає змогу створити системи, які не залежать від однієї центральної організації або особи.

- Довіра. Блокчейн пропонує незмінну і захищену від несанкціонованого доступу систему записів, забезпечуючи надійне і загальне джерело інформації для всіх учасників. Це створює умови, за яких довіра вже вбудована в систему, унеможливлюючи необхідність у посереднику під час укладання угод, навіть із незнайомими особами або організаціями.
- Цінність. Блокчейн відкриває можливості для передання цінностей у будь-якому їхньому вигляді, даючи змогу створювати і передавати активи без необхідності центральних органів управління або посередників. Це означає, що будь-які активи, чи то гроші, чи то контракти, чи то дані, можна передавати та керувати в блокчейн-мережі.

Технологія блокчейн має низку унікальних властивостей і характеристик, які роблять її особливо привабливою в різних галузях застосування, від фінансів до управління ланцюжками поставок і за їх межами[27].

1. Децентралізація

У блокчейні немає централізованого керівного органу або центрального сервера. Замість цього він використовує розподілену мережу вузлів (користувачів або комп'ютерів), кожен з яких має копію всього реєстру транзакцій. Це усуває єдину точку відмови, підвищує стійкість системи до атак і збоїв.

2. Незмінність (Немутабельність)

Після того як дані додано в блокчейн, їх неможливо змінити або видалити без внесення змін до всіх наступних блоків і без згоди більшості мережі. Це забезпечує високий рівень цілісності даних.

3. Прозорість

Більшість блокчейнів мають ступінь прозорості, коли кожен учасник мережі може переглядати транзакції. У публічних блокчейнах, таких як Bitcoin, будь-яка людина може переглянути будь-яку транзакцію в будь-який час.

4. Безпека

Завдяки криптографічному хешуванню та механізмам консенсусу, таким як Proof of Work або Proof of Stake, блокчейн забезпечує високий рівень безпеки[33]. Кожен блок містить унікальний хеш попереднього блоку, створюючи ланцюжок, який захищений від несанкціонованих змін.

5. Стійкість до цензури

Дані в блокчейні не можуть бути піддані цензурі або змінені одним учасником мережі, що робить технологію особливо привабливою для застосунків, де важлива свобода слова та опір цензурі.

6. Зниження витрат

Усунення посередників і автоматизація процесів за допомогою розумних контрактів дають змогу знизити операційні витрати в різних галузях.

7. Поліпшення швидкості транзакцій

Хоча швидкість транзакцій може варіюватися залежно від конкретної реалізації блокчейна, багато систем здатні проводити операції швидше традиційних методів, особливо в міжнародних транзакціях.

8. Токенізація

Блокчейн дає змогу створювати й управляти цифровими токенами, які можуть являти собою різні активи або права, відкриваючи нові можливості для цифрової економіки[35].

Усі ці властивості роблять блокчейн потужним інструментом для створення надійних, прозорих і ефективних систем у найрізноманітніших галузях, таких як фінанси, страхування, банківська справа, охорона здоров'я, державне управління, логістика, Інтернет речей (IoT), а також у сфері медіа та розваг[38]. Блокчейн відкриває перед нами безмежні можливості, зокрема у сферах, де раніше було складно забезпечити ефективну взаємодію і довіру в централізованих системах.

Блокчейн дає змогу зареєструвати будь-який тип майна або активів, включно з фізичними та цифровими активами, як-от електронні реєстрації, цифрові файли тощо. Це спрощує процеси передачі активів, ведення обліку транзакцій і підтвердження прав власності[42]. Можливості включають цифрові

нотаріальні послуги, підтвердження існування активів, індивідуальні страхові схеми та багато іншого.

Фінансові застосування блокчейна різноманітні: від транскордонних платежів і торгівлі акціями до систем лояльності та банківських систем "Знай свого клієнта" (KYC). Первинні випуски монет (ICO) стали популярним способом використання блокчейна для краудсорсингу із застосуванням криптовалюти як цифрових активів[39].

Блокчейн може стимулювати "Мудрість натовпу", змінюючи підходи до управління бізнесом, економікою і національними інститутами[39]. Він може бути використаний для фінансових та економічних прогнозів, децентралізованих ринків прогнозування, голосувань і торгівлі акціями.

У сфері медіа та розваг блокчейн може спростити розподіл ліцензійних платежів і забезпечити прозорість у правах власності на контент. В епоху Інтернету речей блокчейн пропонує платформу для створення децентралізованих однорангових систем, які дозволяють пристроям IoT взаємодіяти один з одним без централізованого контролю[26].

У державному секторі блокчейн знаходить застосування в таких сферах, як реєстрація землі, власності, управління транспортними засобами та електронне голосування[42]. У галузі ланцюжків поставок блокчейн може поліпшити прозорість і надійність системи, забезпечуючи точне відстеження походження товарів і послуг.

Блокчейн надає широкий спектр можливостей для поліпшення різних галузей і галузей, пропонуючи нові способи управління, забезпечення довіри та обміну цінностями[45].

Розглянемо сценарій, при якому зловмисник буде намагатись згенерувати довший ланцюг блоків, ніж у легальних учасників блокчейну. Навіть якщо він досягне успіху, це не дозволить йому привласнювати собі чужі записи або вносити інші довільні зміни в ланцюжок блокчейну. Оскільки вузли не приймуть некоректну транзакцію або блок, що містить таку транзакцію. Зловмисник зможе тільки спробувати змінити одну зі своїх транзакцій[34].

Перегони між чесними учасниками блокчейну і зловмисником можна уявити як біноміальне випадкове блукання[37]. Успішна подія, при якій хороший ланцюг подовжується на один блок, призводить до збільшення відриву на одиницю, а неуспішна, при якій блок створює зловмисник, - до його скорочення. Імовірність надолужити різницю в кілька блоків така сама, як і в задачі про розорення гравця[36]. Суть цієї задачі базується на принципі що гравець має необмежений кредит, починає з деяким дефіцитом і в нього є нескінченно багато спроб, щоб відігратися.

Імовірність того, що такий гравець досягне успіху, подібна ймовірності зловмисника наздогнати легітимних учасників блокчейн мережі, обчислюється таким чином:

$$q_z = \begin{cases} 1, \text{якщо } p \leq q \\ (\frac{q}{p})^z, \text{якщо } p > q \end{cases},$$

де:

p - ймовірність появи блоку в легітимному ланцюжку,

q - ймовірність того, що блок створить зловмисник,

q_z - ймовірність того, що зловнмисник надолужить різницю в z блоків

У разі p > q імовірність експоненціально зменшується зі зростанням числа блоків, на яке відстає зловмисник. Так як всі ставки проти нього, без вдалого прориву на початку його шанси на успіх стають мізерно малими.

Тепер розглянемо, як довго одержувачу платежу варто чекати, перш ніж він буде абсолютно впевнений у тому, що колишній власник не зможе скасувати транзакцію. Ми припускаємо, що зловмисник-відправник дозволяє адресату деякий час вірити, що платіж було проведено, після чого змінює запис.

Легітимний блокчейн користувач створює нову пару ключів і повідомляє свій публічний ключ просто перед підписанням транзакції. Це не дозволить зловмиснику заздалегідь почати працювати над ланцюжком і зробити ривок уперед [44]. Після відправлення транзакції шахрай починає

потай працювати над паралельною версією ланцюжка, що містить альтернативну транзакцію.

Одержувач чекає, поки транзакцію не буде додано в блок і поки той не буде продовжено ще z блоками. Йому невідомий прогрес зловмисника, але якщо середня швидкість генерації чесних блоків - відома величина, то число блоків нападника підпорядковується розподілу Пуассона з математичним очікуванням:

$$\lambda = z\frac{q}{p},$$

Щоб отримати ймовірність того, що зловмисние обжене чесних учасників, ми множимо значення випадкової величини (число створених ним блоків) на ймовірність того, що він зможе надолужити різницю, що залишилася:

$$\sum_{k=0}^{\infty} \frac{\lambda^k e^{-\lambda}}{k!} \begin{cases} (\frac{q}{p})^{(z-k)}, \text{якщо } k \leq z \\ 1, \text{якщо } k > z \end{cases},$$

Після перегрупування доданків і позбувшись нескінченного ряду, отримуємо:

$$1 - \sum_{k=0}^{z} \frac{\lambda^k e^{-\lambda}}{k!} (1 - (\frac{q}{p})^{(z-k)}).$$

Результати розрахунків для різних значень q, z та P наведено в таблиці 3:

Таблиця 2.1

Значення q	Значення z	Значення P
q=0.1	z=0	P=1.0000000
q=0.1	z=1	P=0.2045873
q=0.1	z=2	P=0.0509779
q=0.1	z=3	P=0.0131722
q=0.1	z=4	P=0.0034552
q=0.1	z=5	P=0.0009137
q=0.1	z=6	P=0.0002428

Продовження таблиці 2.1

q=0.1	z=7	P=0.0000647
q=0.1	z=8	P=0.0000173
q=0.1	z=9	P=0.0000046
q=0.1	z=10	P=0.0000012
q=0.3	z=0	P=1.0000000
q=0.3	z=5	P=0.1773523
q=0.3	z=10	P=0.0416605
q=0.3	z=15	P=0.0101008
q=0.3	z=20	P=0.0024804
q=0.3	z=25	P=0.0006132
q=0.3	z=30	P=0.0001522
q=0.3	z=35	P=0.0000379
q=0.3	z=40	P=0.0000095
q=0.3	z=45	P=0.0000024
q=0.3	z=50	P=0.0000006
q=0.10	z=5	P < 0.001
q=0.15	z=8	P < 0.001
q=0.20	z=11	P < 0.001
q=0.25	z=15	P < 0.001
q=0.30	z=24	P < 0.001
q=0.35	z=41	P < 0.001
q=0.40	z=89	P < 0.001
q=0.45	z=340	P < 0.001

Даний експеримент показує, що ймовірність експоненціально падає зі зростанням значення z [43].

2.5 Висновки до другого розділу.

Технологія блокчейн в останнє десятиліття зарекомендувала себе як один із найбільш значущих і перспективних напрямів у сфері інформаційних технологій. Ґрунтуючись на принципах децентралізації, прозорості та безпеки, блокчейн пропонує новий підхід до зберігання й обробки даних, що знаходить широке застосування в найрізноманітніших галузях, від фінансів і банківської справи до логістики, охорони здоров'я і навіть державного управління.

У сучасному світі технології розвиваються стрімко, і блокчейн вирізняється як одна з найперспективніших та найінноваційніших технологій. Його переваги порівняно з традиційними технологіями роблять його ідеальним кандидатом для впровадження в широкий спектр проектів і галузей. Розглянемо ключові аспекти, які роблять блокчейн настільки важливим для майбутнього розвитку різних сфер.

1. Посилена безпека

Блокчейн забезпечує рівень безпеки, який недосяжний для багатьох традиційних систем. Використовуючи розподілену мережу і криптографічні методи шифрування, він гарантує, що дані не можуть бути змінені або знищені несанкціоновано. Це особливо важливо для секторів, де потрібен надійний захист даних, наприклад, у банківській сфері, охороні здоров'я та державному управлінні.

2. Прозорість і відстежуваність

Для проєктів, що вимагають високого ступеня прозорості та можливості відстеження операцій, блокчейн надає ідеальне рішення. Усі записи є публічними і легко перевіряються, що робить процеси прозорими і відкритими для аудиту. Це особливо цінно для логістичних компаній, виробників товарів і послуг, а також для державних контрактів.

3. Децентралізація

Усунення центрального вузла управління і заміна його розподіленою мережею запобігає монополізації влади і контролю над системою. У сфері фінансів це може означати меншу залежність від банків та інших фінансових інститутів, у той час як у медіа та розвагах - більш справедливий розподіл доходів між творцями контенту.

4. Зниження витрат

Використання блокчейна може знизити операційні витрати за рахунок усунення посередників і автоматизації процесів. У бізнесі це може призвести до зниження комісій за транзакції, зменшення витрат на ведення обліку та управління даними.

5. Покращена взаємодія

Блокчейн полегшує взаємодію між різними сторонами, забезпечуючи обмін даними та цінностями без необхідності довіри між учасниками. Це може радикально змінити способи проведення бізнесу, відкриваючи нові можливості для співпраці та партнерства.

6. Інновації в продуктах і послугах

Блокчейн дає змогу розробляти інноваційні продукти та послуги, як-от смарт-контракти, децентралізовані додатки (DApps) і токенізовані активи. Ці інструменти можуть бути використані для створення нових бізнес-моделей у різних галузях.

Отже, блокчейн являє собою технологію, здатну кардинально змінити багато аспектів нашого життя і роботи. Його впровадження в різні проєкти та галузі обіцяє не тільки поліпшення наявних процесів, а й створення абсолютно нових, ефективніших і безпечніших систем. З огляду на його переваги порівняно з традиційними технологіями, блокчейн, без сумніву, варто використовувати дедалі частіше й активно інтегрувати в нові та наявні проєкти.

РОЗДІЛ 3. ПРОГРАМНИЙ ЗАСТОСУНОК МЕНЕДЖЕРА ПАРОЛІВ НА ОСНОВІ БЛОКЧЕЙН ТЕХНОЛОГІЙ

3.1 Опис середовища розробки.

Даний менеджер паролей повністю побудований з допомогою мови програмування Python.

Python - це високорівнева інтерпретована мова програмування, відома своєю легкістю у вивченні та читабельністю. Вона підкреслює простоту і гнучкість, дозволяючи розробникам писати чіткий, логічний код для малих і великих проектів. Python підтримує декілька парадигм програмування, включаючи процедурне, об'єктно-орієнтоване та функціональне програмування.

Основні можливості мови включають в себе:

Динамічну типізацію. Python не вимагає явного оголошення змінних перед їх використанням, що забезпечує більшу гнучкість у кодуванні.

Управління пам'яттю. Виділення пам'яті відбувається автоматично за допомогою вбудованого збирача сміття.

Велику стандартну бібліотеку. Стандартна бібліотека Python дуже велика і включає модулі та функції для широкого спектру завдань.

Відкритий вихідний код. Розробляється під ліцензією з відкритим вихідним кодом, затвердженою OSI, що робить його вільно використовуваним і розповсюджуваним, навіть для комерційного використання.

Велику спільноту поціновувачів. Python має велику та активну спільноту, яка сприяє створенню великої екосистеми бібліотек та фреймворків.

Python широко використовується у веб-розробці, наукових та математичних обчисленнях, штучному інтелекті, розробці програмного забезпечення та системних сценаріях. Її синтаксис дозволяє розробникам

виражати концепції у меншій кількості рядків коду, ніж це було б можливо у таких мовах, як C++ або Java.

Для запуску коду я використовувала Jupyter Notebook.

Jupyter Notebook - це веб-додаток з відкритим вихідним кодом, який дозволяє створювати та ділитися документами, що містять живий код, рівняння, візуалізації та описовий текст. Він широко використовується для очищення та перетворення даних, чисельного моделювання, статистичного моделювання, машинного навчання та багато іншого.

Серед переваг Jupyter Notebook виділяють:

Інтерактивність. Дозволяє інтерактивне кодування, що чудово підходить для експериментів та ітеративного аналізу даних.

Підтримка декількох мов. В основному використовується для Python, але підтримує багато мов, таких як R, Julia та Scala.

Візуалізація. Інтегрується з бібліотеками візуалізації даних.

Спільне використання. Блокнотами можна легко ділитися з іншими.

До недоліків данного додатку можна віднести:

Контроль версій. Складно використовувати з системами контролю версій, такими як Git.

Великі блокноти. Можуть повільно завантажуватися і в них важко орієнтуватися, коли вони стають дуже великими.

Порядок виконання. Порядок виконання може заплутатися, оскільки комірки можуть виконуватися в неправильному порядку.

Не ідеальний для виробництва. Вони не призначені для створення готового до виробництва коду.

Безпека. Запуск блокнотів Jupyter може становити ризик для безпеки, якщо їх не налаштовано належним чином.

3.2 Структура та функціонал менеджера паролей.

Для реалізації менеджера паролей мною були використані такі бібіотеки мови програмування Python:

- PySimpleGUI
- hashlib
- time
- requests
- cryptocode

Розглянемо кожну з цих бібліотек детальніше:

Бібліотека PySimpleGUI в Python - це інструмент для створення графічних користувацьких інтерфейсів (GUI). Вона призначена для того, щоб зробити процес розробки інтерфейсів якомога простішим і доступнішим.

PySimpleGUI дає змогу розробникам легко створювати вікна, кнопки, текстові поля, мітки та інші елементи інтерфейсу без необхідності глибоких знань у сфері GUI.

PySimpleGUI сумісна з кількома основними GUI фреймворками Python, такими як Tkinter, Qt, WxPython і веб-версії. Це дає гнучкість у виборі фреймворку залежно від потреб і переваг. PySimpleGUI можливо використовувати для створення інтерактивних елементів, таких як поля для введення і кнопки. Це робить додаток більш інтуїтивно зрозумілим і зручним для користувача.

PySimpleGUI працює на усіх основних операційних системах, Windows, macOS і Linux, що забезпечує платформну незалежність розробленого застосунку.

В розробленому в ході даної роботи менеджері паролей з допомогою PySimpleGUI використовується для реалізації:

- Вікон інтерфейсу. PySimpleGUI використовується для створення різних вікон інтерфейсу, включно зі стартовим вікном, вікном реєстрації, вікном входу в систему та вікном керування паролями. Ці вікна надають візуальний інтерфейс для взаємодії з користувачем.
- Елементів введення даних. У цих вікнах PySimpleGUI використовується для створення текстових полів, куди користувачі можуть вводити дані, як-от логін, пароль і назву сайту. Це забезпечує зручний спосіб збору даних від користувача.
- Кнопок для взаємодії з користувачем. PySimpleGUI дає змогу легко додавати кнопки для різноманітних дій, як-от "Увійти", "Зареєструватись", "Додати новий пароль" та "Подивитись пароль". Ці кнопки активують відповідні функції в додатку.
- Відображення інформації. Бібліотека використовується для відображення текстової інформації, наприклад, вітальних повідомлень, інструкцій, інформації про стан (наприклад, помилок входу) і списку збережених паролів.
- Організації лейауту. PySimpleGUI дає змогу зручно організовувати розташування елементів інтерфейсу у вікні, що забезпечує чітке та зрозуміле компонування інтерфейсу.
- Обробки подій. Через PySimpleGUI реалізується обробка подій, таких як натискання на кнопки. Кожна дія користувача (наприклад, натискання на кнопку) викликає відповідну функцію або метод у коді, який виконує необхідні операції (наприклад, перевірку введених даних або додавання нового пароля).

Бібліотека hashlib у Python - це стандартна бібліотека, що надає функціональність для криптографічного хешування. Хешування - це процес перетворення вхідних даних (наприклад, тексту) в унікальний рядок фіксованої довжини, відомий як хеш.

Основні характеристики та використання hashlib:

- Криптографічні хеш-функції. Hashlib підтримує кілька популярних і широко використовуваних хеш-функцій, таких як SHA-1, SHA-256, SHA-512, і MD5. Кожна з цих функцій генерує хеш-значення, яке являє собою "відбиток" вхідних даних.
- Безпека. Хеш-функції використовуються для забезпечення безпеки даних. Вони дають змогу перевіряти цілісність даних, не розкриваючи їхній вміст. Наприклад, хеші паролів використовуються для безпечного зберігання паролів, де в базі даних зберігається тільки хеш пароля, а не сам пароль.
- Необоротність. Хеш-функції є необоротними, тобто неможливо відновити вихідні дані з хеш-значення. Це робить їх корисними для захисту конфіденційної інформації.
- Унікальність. Хеш-функції розроблені таким чином, щоб мінімізувати ймовірність "колізій" - ситуацій, коли різні вхідні дані дають однаковий хеш.

Бібліотека hashlib у цьому менеджері паролів використовується для створення безпечних хешів даних. Хеш-функції перетворюють вхідні дані у фіксований набір символів, який виступає в ролі унікального "відбитка" для цих даних. Це забезпечує безпеку та цілісність інформації. Hashlib використовується для створення SHA-256 хешів, коли відбувається додавання нових паролів і формування блоків у блокчейні. Функція calculate_hash приймає кілька параметрів (індекс, попередній хеш, тимчасова мітка, дані), конкатенує їх у рядок, кодує в байти і створює хеш для цього рядка. Створені хеші використовуються в блоках блокчейна, представляючи цифрові підписи блоків і забезпечуючи їхню унікальність і цілісність. Окрім того, при реєстрації пароль користувача також хешується за допомогою алгоритму sha1 цієї бібліотеки.

Бібліотека time в Python надає функції для роботи з часом, вимірювання часових інтервалів і керування часом у різних форматах.

Вона має велику кількість різноманітних можливостей, найпопулярніші з них:

- Поточний час. Функція time.time() повертає поточний час у секундах від початку епохи (зазвичай 1 січня 1970 року). Це корисно, наприклад, для фіксації часу подій у програмі.
- Затримка виконання. Функція time.sleep(секунди) призупиняє виконання програми на певну кількість секунд.
- Вимірювання минулого часу. Можна вимірювати різницю часу між двома точками в коді для визначення часу виконання певної ділянки.

У реалізованому мною менеджері паролей бібліотека time застосовується в таких моментах:

- Позначка часу для блоків у блокчейні. У блокчейні кожен блок має містити мітку часу, яка фіксує момент створення блоку. Бібліотека time дає змогу зручно отримувати поточний час, який потім використовується під час формування блоку.
- Вимірювання часу виконання операцій. Під час роботи з блокчейном важливо вимірювати час виконання певних операцій, наприклад додавання нового пароля в блокчейн. Це дає змогу оцінити продуктивність і ефективність роботи програми.
- Синхронізація часу між вузлами блокчейна. У деяких випадках може знадобитися синхронізувати час між вузлами блокчейна. Бібліотека time дає змогу встановлювати часові позначки та порівнювати їх для забезпечення узгодженості часу між різними частинами системи.

Requests - це бібліотека мови Python, призначена для надсилання HTTP-запитів і обробки отриманих відповідей. Ось кілька ключових аспектів і можливостей бібліотеки requests:

- Надсилання HTTP-запитів. Requests дозволяє легко надсилати HTTP-запити, такі як GET, POST, PUT, DELETE та інші.

- Передача параметрів. Можна включати параметри в запит, наприклад, для передачі даних у запиті GET.
- Надсилання даних. Для запитів типу POST можна надсилати дані, наприклад, у вигляді форми.
- Обробка відповідей. Requests дає змогу легко обробляти отримані відповіді, включно з доступом до заголовків, вмісту та статусу.
- Обробка JSON. Вбудовані засоби для роботи з JSON, такі як json() метод, роблять обробку JSON-відповідей простою.
- Управління сеансами. Requests підтримує використання сеансів, що дає змогу ефективно працювати з кількома запитами.
- Обробка винятків. Requests генерує винятки для різних помилок HTTP, що полегшує обробку помилок.

Бібліотека cryptocode в Python призначена для спрощення процесу шифрування і дешифрування текстових даних. Вона забезпечує базову функціональність для виконання цих завдань за допомогою різних криптографічних алгоритмів. Основне застосування cryptocode - це забезпечення безпеки даних шляхом їхнього шифрування перед передаванням або зберіганням і подальшого дешифрування за потреби.

Основна функція cryptocode - це шифрування текстових даних за допомогою заданого ключа і дешифрування цих даних назад у вихідний текст з використанням того ж ключа.

Бібліотека cryptocode підтримує поширені алгоритми шифрування, такі як AES. Хоча cryptocode забезпечує базову безпеку, важливо пам'ятати, що безпечне шифрування залежить не тільки від обраної бібліотеки, а й від правильного використання криптографічних алгоритмів, керування ключами та захисту ключів шифрування.

Ця бібліотека використовується для безпосереднього шифрування паролів користувачів перед їх записом у блокчейн і дешифруванням при спробі користувача переглянути свій пароль.

Узагальнена блоксхема реалізації менеджера паролей з використанням блокчейн технологій представлена в додатку А.

Розглянемо процес взаємодії користувача з програмою. При відкритті програми вас вітає стартове вікно менеджера паролів (рис 3.1).

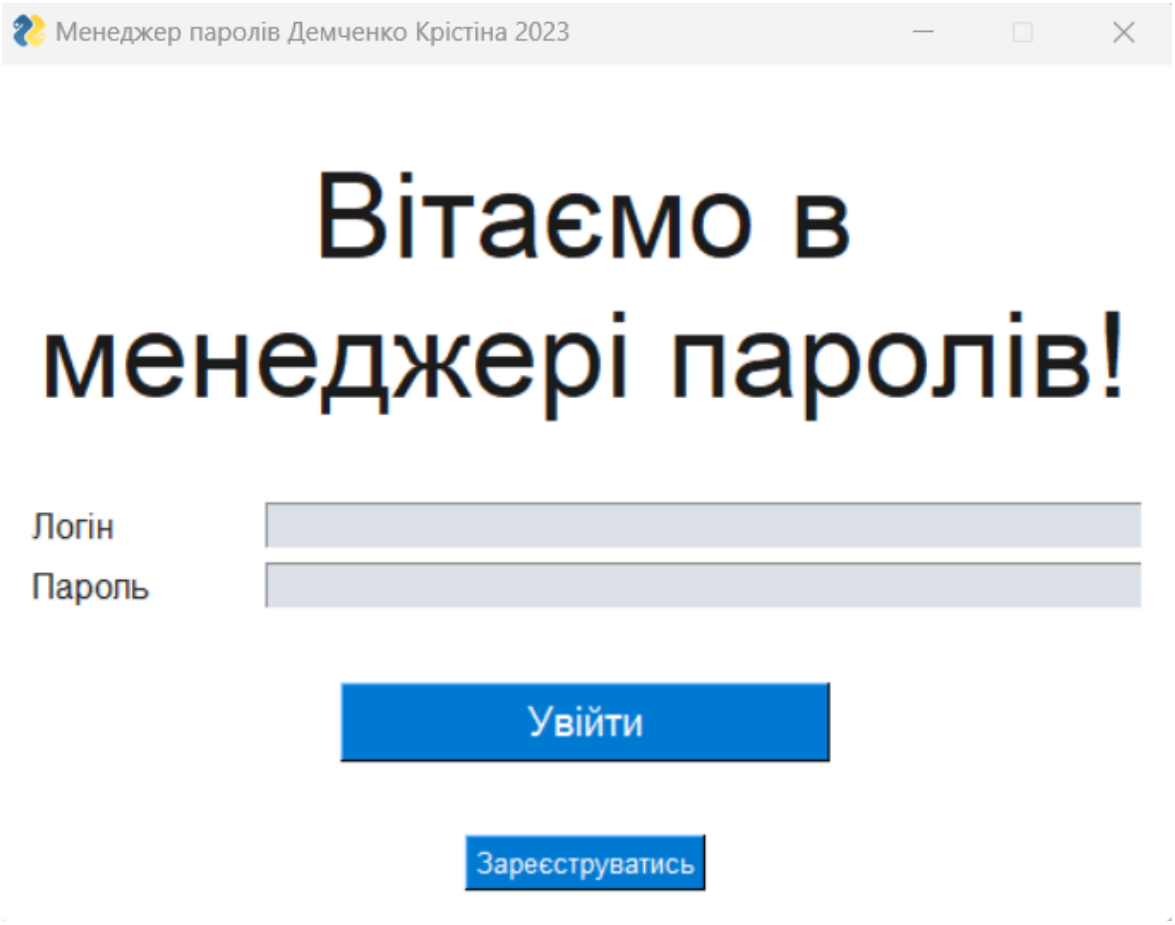


Рис.3.1 Стартове вікно менеджера паролей

Тут ви маєте дві основні опції:

- Увійти. Якщо у вас вже є обліковий запис, ви вводите свій логін та пароль для доступу до ваших збережених паролів.
- Зареєструватись. Якщо у вас ще немає облікового запису, ви можете створити його, вказавши ім'я, логін та пароль.

Користувачам, які у даному менеджері паролей вперше, зі стартовго вікна слід перейти на вікно реєстрації, його вигляд зображено на рисунку 3.2

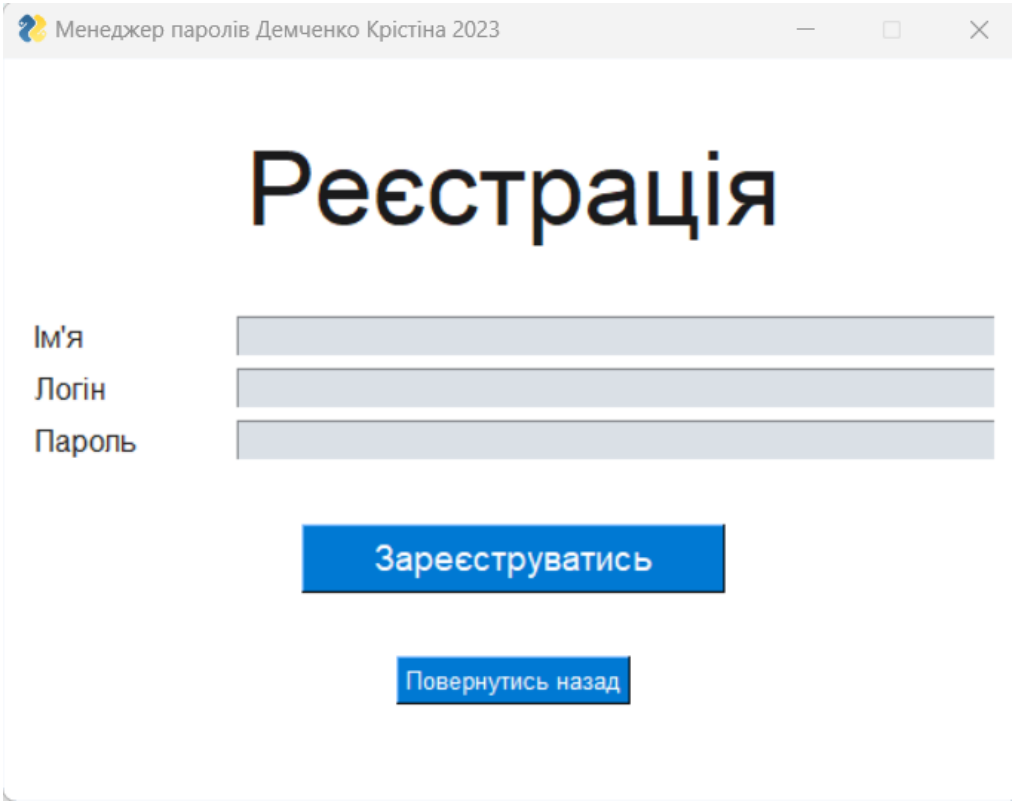


Рис.3.2 Вікно реєстрації

У вікні реєстрації необхідно ввести наступну інформації:

- Ваше ім'я.
- Логін, який ви будете використовувати для входу в систему.
- Пароль для вашого облікового запису.

Варто зазначити що при реєстрації менеджер паролей перевіряє надійність паролю яким ви хочете захистити свій акаунт, якщо пароль не достатньо надійний висвітиться допоміжне вікно у якому вказано чого не вистачає щоб пароль вважався надійним (рис 3.3).

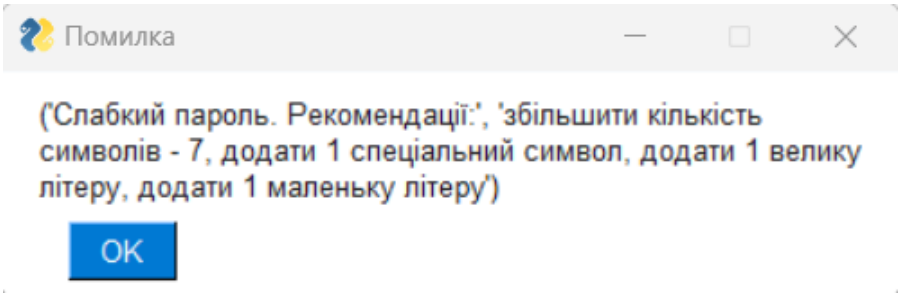


Рис 3.3 Вікно помилки надійності пароля

Для перевірки пароль повинен відповідати таким критеріям:

- Довжина паролю не менше 8 символів

- Не менше 1 цифри
- Не менше 1 маленької літери
- Не менше 1 великої літери

У випадку коли усі норми дотримані користувач отримує повідомлення про те що реєстрація успішна, наприклад зареєструємо користувача з ім’ям Крістіна, логіном Диплом_магістра_2023 та паролем 1Kd*45LwRT92 (рис3.4) та натиснемо на кнопку «Зареєструватись»

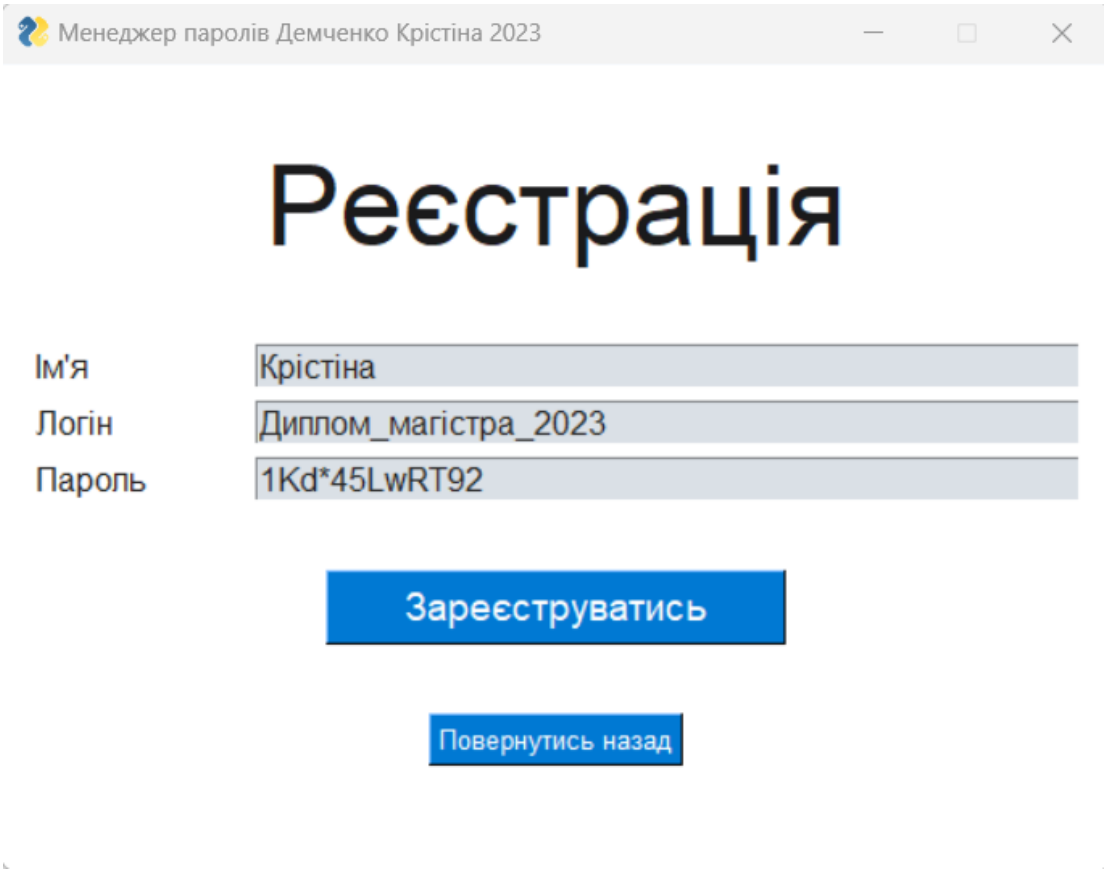


Рис.3.4 Демонстраційне фото реєстрації користувача

Відкриється вікно, яке повідомляє що цього користувача успішно зареєстровано (рис.3.5).

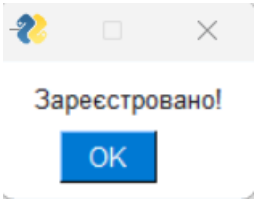


Рис.3.5 Демонстраційне фото успішної реєстрації

При спробі знову зареєструвати користувача з таким самим логіном ми побачимо повідомлення про те що такого користувача не можна реєструвати (рис.3.6).

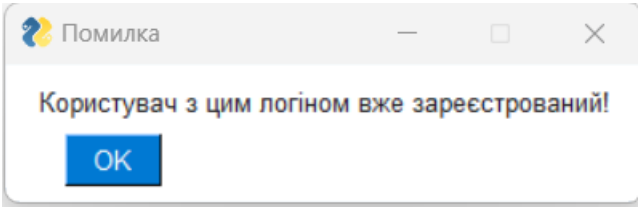


Рис.3.6 Демонстраційне фото помилки повторної реєстрації користувача.

Повернемось у стартове вікно і увійдемо в обліковий запис щойно створеного користувача, для цього вводимо логін і пароль (рис. 3.7) і натискаємо кнопку «Увійти»

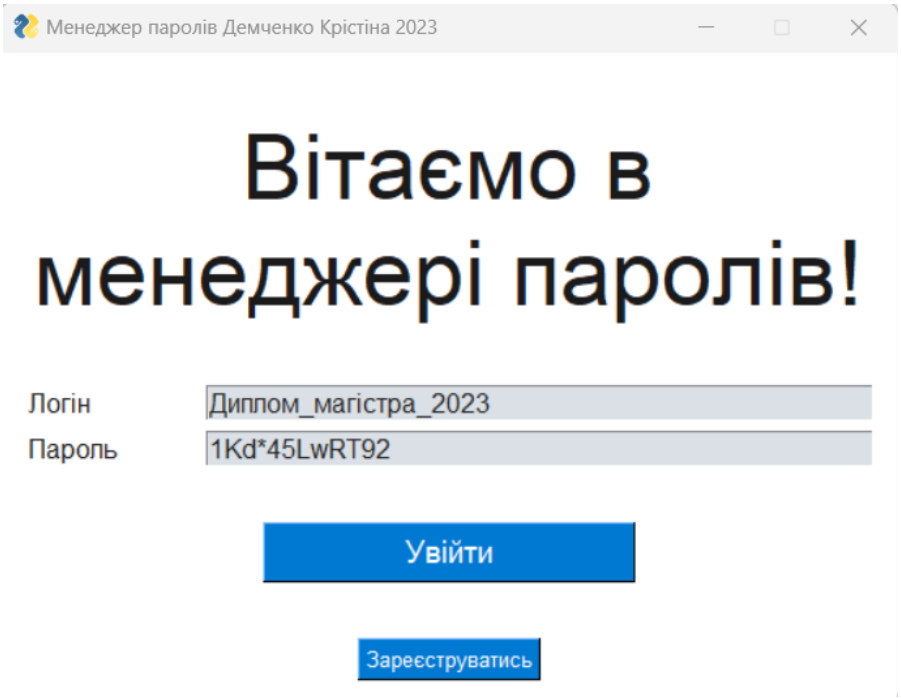


Рис.3.7 Вхід в обліковий запис

Після входу бачимо головне вікно програми (рис 3.8) в якому ми бачимо привітання користувача, а також наступні опції:

- Подивитись пароль, для перегляду збережених паролів з різних сайтів та сервісів.
- Додати новий пароль, надає можливість зберегти новий пароль для сайту або сервісу.

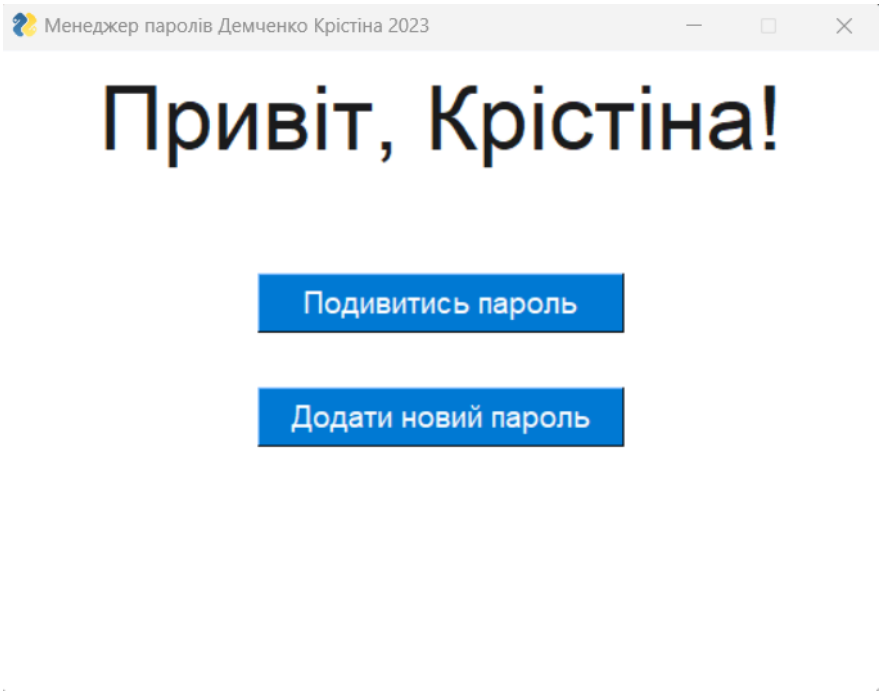


Рис.3.8 Головне вікно менеджера паролей

Якщо ви натискаєте «Додати новий пароль», вам потрібно ввести у вікні яке відкриється назву сайту або сервісу і пароль, який хочете зберегти. У випадку якщо один і той самий користувач двічі введе пароль для одного сайту або сервісу при спробі переглянути пароль він побачить лише той, що був введений останнім. Додамо паролі для кількох сервісів (рис.3.9-3.11)

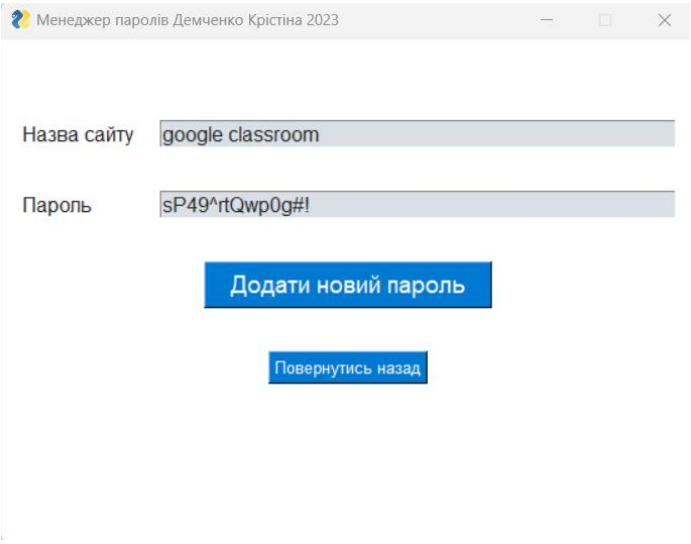


Рис.3.9 Додавання паролю для google classroom

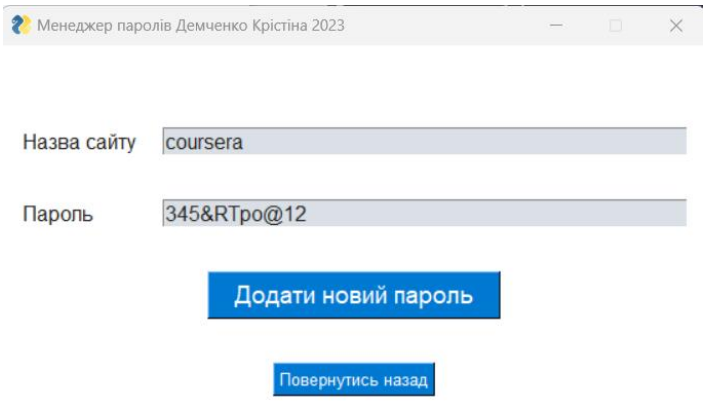


Рис.3.10 Додавання паролю для coursera

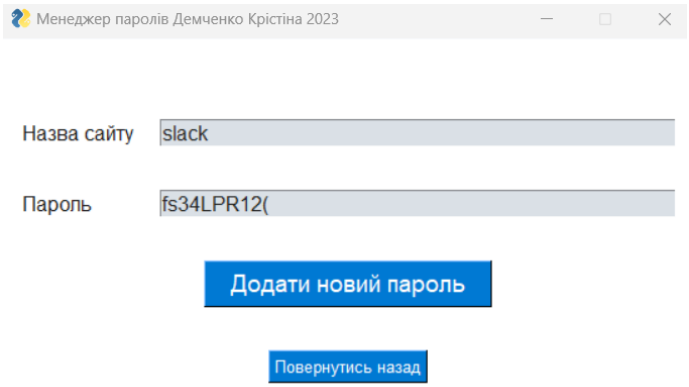


Рис.3.11 Додавання паролю для slack

Повернемось назад і переглянемо усі паролі. Обравши «Подивитись пароль», відкриється вікно зі списком всіх сайтів та сервісів, для яких ви зберегли паролі (рис.3.12).

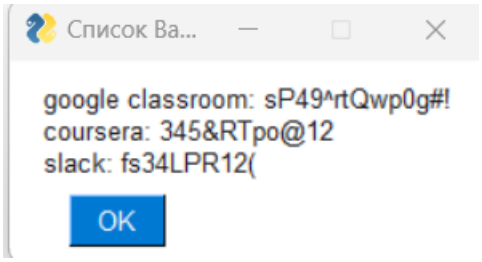


Рис.3.12 Перегляд паролей користувача

Щоб вийти з програми, потрібно просто закрити вікно менеджера паролів. І усі ваші дані залишаться безпечно збережені для наступного входу в систему.

3.3 Оцінка ефективності та порівняння з існуючими системами.

Для оцінки ефективності розробленого додатку можна розглянути такі параметри:

- Функціональність. Додаток пропонує основні функції управління паролями та взаємодії з блокчейном.
- Користувацький інтерфейс. Графічний інтерфейс на базі PySimpleGUI забезпечує простоту використання і наочність, що покращує взаємодію користувача з програмою.
- Продуктивність. Час відгуку додатка на запити користувача та швидкість обробки транзакцій у блокчейні є практично миттєвим, не більше 100 мілісекунд, щоб користувач відчував, що система реагує негайно. Проте варто зазначити, що зі збільшенням кількості користувачів час на обробку транзакцій збільшиться, що призведе до збільшення часу на обробку запитів користувачів.
- Масштабованість. Нажаль в ході даної роботи неможливо перевірити здатність даної системи до масштабування, проте спираючись на теоретичну базу можна стверджувати що дана система має гарний потенціал до масштабування, однак є кілька аспектів, які потрібно враховувати:

1. Архітектура блокчейна. Система використовує блокчейн, який теоретично є масштабованим, але практичне масштабування залежить від реалізації консенсусного алгоритму та обробки транзакцій.

2. Використання ресурсів. Масштабування може вимагати збільшення обчислювальних та пам'ятних ресурсів, особливо якщо кількість користувачів та транзакцій зростає.

3. Відмовостійкість. Необхідно забезпечити відмовостійкість системи при масштабуванні, що може включати додаткові механізми синхронізації та бекапування.

4. Продуктивність інтерфейсу. Графічний інтерфейс повинен залишатися відгуковим незалежно від обсягу даних.

- Безпека. Додаток використовує хешування та шифрування для зберігання паролів, що сприяє підвищенню безпеки збережених даних.
- Оновлюваність і підтримка. Легкість внесення змін і оновлень також є ключовим аспектом для довгострокової ефективності програми. Запропонований додаток наразі виконаний з використанням мінімальних затрат і вимог, а одже не має безлічі додаткових функцій, які б захаращували код та робили його нечитабельним, що значно спрощує процес оновлення. Проте варто зазначити, що блокчейн технологія не любить змін, тож варто бути обережним при введені оновлень.
- Сумісність і залежності. Наскільки легко застосунок може бути інтегрований з іншими системами та які зовнішні залежності він має. Даний застосунок має працювати на усіх найпопулярніших операційних системах для комп’ютерів, таких як Windows, macOS та Linux.

Для повноцінної оцінки слід провести тестування застосунку з реальними користувачами, проаналізувати зворотний зв'язок або виконати стрес-тестування системи.

Варто зазначити що наразі немає загальнодоступних менеджерів паролів, які використовували б у своїй реалізації технологію блокчейн, оскільки ця технологія є відносно новою і доволі складною для її імплементація в менеджери паролів. Не так давно почали з’являтись книжки або предмети в університетах які б пояснювали цей складний алгоритм простою і зрозумілою мовою[41]. Саме через це ми проведемо порівняння запропонованого в ході цієї роботи рішення з менеджерами паролів, які були розглянуті у третьому пункті 1 розділу. Для цього знову коротко розглянемо кожен із них та зробимо нову порівняльну таблицю з використанням тих самих критеріїв (таблиця 2.2).

1. Розроблений менеджер паролів на базі блокчейну.

Новаторський підхід з використанням блокчейн технології, забезпечує розподілене зберігання даних і підвищену безпеку через незмінність записів. Відсутність централізованого контролю знижує ризики атак.

2. Enpass.

Enpass пропонує автономне зберігання даних на пристрої користувача, забезпечуючи високий рівень контролю над даними. Він підтримує синхронізацію через хмарні сховища, але не вимагає обов'язкового зберігання даних в хмарі. Окрім паролів також дозволяє зберігати ваші картки або документи.

3. NordPass.

NordPass відомий своїм сучасним інтерфейсом та використанням передових технологій шифрування. Як продукт компанії NordVPN, він також наголошує на сильному захисті приватності та безпеки.

4. Keeper.

Keeper зосереджений на безпеці і забезпечує багатофакторну автентифікацію та захист від кіберзагроз. Це додаток вищого рівня, який використовується в бізнесі та корпоративному середовищі.

5. 1Password.

1Password є одним з найбільш дружніх для користувачів менеджерів паролів, з інтуїтивно зрозумілим інтерфейсом та високим рівнем інтеграції з різними платформами. В розширеній версії окрім паролів також дозволяє зберігати ваші картки або документи. Він також забезпечує потужні можливості для управління паролями в команді або сім'ї.

6. KeePass.

KeePass - це відкритий додаток для управління паролями, який пропонує гнучкість і високий рівень кастомізації. Хоча інтерфейс може здатися менш сучасним, KeePass пропонує сильні опції шифрування і велику свободу вибору налаштувань, а також, даний менеджер паролей має відкритий код, що дозволяє будь-якому спеціалісту зі сфери кібербезпеки перевірити його і переконатись, що застосунок є безпечним і не має зловмисного коду.

До критеріїв оцінки віднесено:

1. Додаткове зберігання карток чи документів. Цей критерій є дуже важливим, оскільки багато користувачів не лише зберігають паролі, але й потребують безпечного місця для зберігання конфіденційних документів або інформації про платіжні картки. Ця функція забезпечує додатковий рівень зручності та безпеки.

2. Прозорість. Розуміння того, як працює менеджер паролей і як він захищає дані користувачів є критично важливим для довіри користувачів. Прозорість може включати в себе відкритий код, чіткі політики конфіденційності та безпеки, а також легкість використання.

3. Вартість повної версії. Цей критерій також є ключовим, оскільки більшість користувачів шукають ефективне співвідношення ціни та якості продукту. Іноді безкоштовні версії пропонують обмежені можливості, тому знання про вартість повної версії дозволяє робити більш обґрунтований вибір.

4. Мультиплатформеність. У сучасному світі люди використовують різні пристрої та операційні системи. Тому важливо, щоб менеджер паролей міг синхронізуватися та працювати на різних платформах. Це забезпечує зручність і гнучкість для користувачів.

Кожен з цих критеріїв допомагає оцінити як функціональність, так і загальну вигоду користувача від використання конкретного менеджера паролей.

Таблиця 2.2

Критерії оцінювання	Розроблений менеджер паролів	Enpass	NordPass	Keeper	1Password	KeePass
Додаткове зберігання карток чи документів	0	5	0	0	5	0
Прозорість	5	3	2	3	2	4

Продовження таблиці 2.2

Вартість повної версії	5	3	4	2	1	5
Мультиплатформеність	3	4	5	5	4	5

Не зовсім доцільно порівнювати новий, невеликий, щойно розроблений продукт з застосунками, які розробляли корпорації і над створенням та тестуванням яких працювали сотні спеціалістів, проте менеджер паролей на основі блокчейн технології має ряд вагомих переваг, які для деяких користувачів привілеюють перед більш зручним інтерфейсом чи більшою кількістю додаткових функцій та можливостей.

До таких переваг можна віднести:

- Розподілене зберігання даних. Використання блокчейну дозволяє зберігати дані на різних вузлах, що зменшує ризик втрати даних через збій одного сервера.
- Підвищена безпека. Блокчейн може забезпечити вищий рівень безпеки завдяки криптографічному шифруванню та неможливості зміни записаних даних.
- Транспарентність. Блокчейн забезпечує прозорість операцій, дозволяючи відстежувати всі транзакції.
- Відсутність централізованого контролю. Блокчейн-базований менеджер паролів не залежить від однієї централізованої системи, що знижує ризик централізованих атак.
- Незмінність даних. Внесені в блокчейн дані не можуть бути змінені або видалені, що забезпечує додатковий рівень безпеки.

3.4. Висновки до третього розділу.

У цьому розділі мною було розроблено покращений авторський алгоритм створення менеджеру паролей, що використовує блокчейн технологію та має підвищений рівень захисту.Загалом даний застосунок , представляє собою інноваційний підхід до зберігання та управління паролями. Використання блокчейну забезпечує безпечне і розподілене зберігання даних, а також незмінність і прозорість усіх записів, що значно знижує ризик централізованих атак і втрати даних.

При оцінці ефективності цього менеджера порівняно з іншими на ринку (такими як Enpass, NordPass, Keeper, 1Password, KeePass) можна сказати, що розроблене рішення не є найоптимальнішим згідно обраним для оцінки критеріям, проте будь який програмний застосунок створюється під свого користувача та має ряд своїх переваг та недоліків, отже важливо врахувати наступні кілька перелічених далі параметрів та визначити чи підходить даний менеджер паролів конкретному користувачу. Незмінність блокчейну забезпечує додатковий рівень безпеки, який може бути відсутній у традиційних менеджерах. Блокчейн технологія дозволяє користувачам відстежувати всі зміни та транзакції, забезпечуючи високу прозорість. Через особливості блокчейну, такий менеджер може мати проблеми з швидкістю обробки та масштабуванням у порівнянні з більш традиційними системами, які використовують сучасні централізовані бази даних. Традиційні менеджери можуть пропонувати більш інтуїтивний інтерфейс та кращу інтеграцію з різними платформами, підтримуватись на мобільних телефонах, пропонувати синхронізацію, тощо.

Отже, цей менеджер паролів може бути ідеальним рішенням для тих, хто цінує безпеку на основі блокчейну і готовий прийняти потенційні компроміси щодо зручності, а при зночному збільшенні користувачів і швидкості обробки транзакцій.

РОЗДІЛ 4. ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

Природне середовище.

Наукове визначення навколишнього природного середовища (або довкілля) описує всі його живі і неживі об'єкти, що існують на Землі або в певній її частині. До складових навколишнього середовища входять природне та соціальне середовище. Основними природними складовими довкілля є:

- Повітря. Повітря є одним з найважливіших продуктів для життя людини. Людина може прожити лише кілька хвилин без доступу до повітря. Забруднення повітря може негативно вплинути на здоров'я людини та навколишнє середовище.
- Вода. Вода є не менш важливою складовою для життя людини. Людина може вижити без води лише декілька днів. Питна вода необхідна для задоволення потреб населення, а також для виробничих процесів та інших цілей. Доступ до безпечної питної води стає все важчим завдяки забрудненню та надмірному використанню.
- Земля. Земля є ключовою складовою для сільського господарства та виробництва їжі. Важливо зберігати родючі ґрунти та дбати про збереження природних ресурсів.
- Природа. Природа включає всі природні екосистеми, які забезпечують різноманітність видів та екологічний баланс на Землі. Руйнування природних середовищ може призвести до вимирання видів та порушити екологічну рівновагу.

Негативні впливи на довкілля включають зміну клімату, забруднення повітря та води, виснаження природних ресурсів, руйнування природних середовищ і зменшення біорізноманіття.

Сучасні екологічні проблеми включають в себе зміну клімату, вимирання видів, забруднення пластиком, вирубку лісів та інші загрози для навколишнього середовища.

Прийняття заходів для збереження та відновлення навколишнього середовища є важливою задачею для забезпечення сталого розвитку та збереження природних ресурсів для майбутніх поколінь.

Державною установою "Вінницький ОЛЦ МОЗ України" проводяться щорічні моніторингові дослідження атмосферного повітря в 149 населених пунктах, включаючи міські та сільські. У 2019 році з 4185 досліджених проб атмосферного повітря 6,9% не відповідали вимогам щодо вмісту забруднюючих речовин. Ця кількість незадовільних проб зросла порівняно з 2018 роком (5,3%) і 2017 роком (2,9%). Проби з перевищенням ГДК в міських поселеннях збільшилися з 5,8% в 2018 році до 8,5% в 2019 році, тоді як в сільських поселеннях показники залишилися стабільними на рівні 4,4% в 2019 році. Велика кількість нестандартних проб повітря була внаслідок перевищеного вмісту різних речовин, таких як пил, оксид вуглецю, азоту діоксиду, фенолу та інших.

Зараз в світі відзначається Всесвітній день навколишнього середовища, під час якого проводяться різноманітні заходи та акції на підтримку чистого середовища. Незважаючи на це, досягнення суттєвих змін щодо збереження довкілля залишається важкою задачею. Для покращення ситуації необхідно посилювати контроль за виконанням природоохоронного законодавства, залучати громадські організації та активістів до спільної роботи та підкреслювати важливість партнерських відносин для забезпечення безпечного та благополучного майбутнього для всіх народів і країн.

Для збереження та сталого розвитку природного середовища необхідно пам’ятати про:

- Звільнення викидів CO2. Кліматичні зміни є однією з найбільших загроз для природного середовища і людства в цілому. Підвищення рівня CO2 та інших парникових газів у атмосфері сприяє глобальному потеплінню, змінам

клімату, підвищенню рівня моря та зміні в погодних умовах. Для зменшення впливу нашої діяльності на клімат, необхідно зменшувати викиди парникових газів, зокрема шляхом переходу до відновлюваних джерел енергії, поліпшення енергоефективності та зберігання вуглецю в екосистемах, таких як ліси та мокрощі.

- Біорізноманітість. Збереження біорізноманітності є ключовим елементом збереження природного середовища. Різноманітність життя на Землі забезпечує стійкість екосистем, поліпшує продуктивність сільськогосподарських угідь і забезпечує харчову безпеку. Зниження рівня біорізноманітності через знищення природних середовищ, втрату місць існування та вимирання видів створює серйозні загрози для екосистем і людського благополуччя.
- Водні ресурси. Вода є життєво важливим ресурсом для всіх форм життя на Землі. Забруднення водних джерел, надмірне використання та зміна клімату можуть призвести до дефіциту питної води та загрозити екосистеми водойм. До збереження водних ресурсів включається поліпшення якості води, збереження водних екосистем та раціональне використання води в сільському та міському господарстві.
- Захист екосистем. Екосистеми, такі як ліси, водойми, пустелі та коралові рифи, грають важливу роль у збереженні біорізноманітності та забезпеченні послуг екосистем для людей. Захист цих екосистем включає в себе збереження природних об'єктів, боротьбу з незаконним вирубуванням лісів та забрудненням водойм, а також створення заповідників і морських резерватів для збереження унікальних середовищ.
- Охорона диких тварин. Вилучення та знищення природних середовищ, контрабанда та незаконний полювання можуть призвести до вимирання багатьох видів тварин. Збереження диких тварин включає в себе запуск програм з охорони та відтворення видів, розробку стратегій збереження та боротьбу з незаконним полюванням.

- Усвідомлення та освіта. Освіта та усвідомлення важливості природного середовища грають критичну роль у збереженні природи. Інформаційні кампанії, екологічна освіта та популяризація сталого споживання можуть сприяти залученню громадськості до збереження природи та раціонального використання ресурсів.
- Сумісне життя з природою. Основною метою є створення гармонійних відносин між людьми і природним середовищем, забезпечення сталого розвитку та збереження природних ресурсів для майбутніх поколінь. Спільні зусилля урядів, громад та бізнесу можуть сприяти створенню більш сталого та екологічно свідомого суспільства.

ВИСНОВКИ

В ході виконання цієї кваліфікаційної роботи, мною було розглянуто ключові аспекти забезпечення безпеки інформації та визначено, що безпека інформації стає дедалі актуальнішою та важливішою в сучасному світі, де загрози та види атак на інформаційні системи постійно зростають. У даній роботі було розглянуто найпопулярніші на даний момент загрози, такі як шкідливі програми, фішинг, DDoS-атаки, оскільки вони можуть завдати серйозної шкоди організаціям і приватним особам.

Було виокремлено чинники, які можуть послабити системи захисту, включно з людським чинником, недостатнім оновленням програмного забезпечення і недостатньою навченістю персоналу. Ці фактори підкреслили важливість не тільки технічних заходів безпеки, а й освіти та навчання користувачів. Також варто зазначити, що навіть з поширеними рекомендаціями щодо створення міцних паролів і оновлення їх регулярно, багато користувачів ігнорують ці заходи. Окрім того мною було помічно, що існує невдоволення щодо існуючих менеджерів паролів, які зберігають інформацію на серверах компаній або в хмарних сховищах, що може викликати побоювання щодо безпеки даних.

На базі отриманої інформації мною було визначено основні критерії, яким має відповідати надійна система захисту інформаційної мережі, включно з конфіденційністю, цілісністю та доступністю даних. Ці критерії допомагають визначити рівень захисту та ефективність заходів безпеки у кожній окремо взятій системі.

Мною було визначено, що блокчейн-технології можуть вирішити деякі проблеми, що існують у цій сфері. Публічні блокчейни пропонують розподілену базу даних, яка дає змогу децентралізувати зберігання та обробку інформації. Це може збільшити рівень довіри користувачів і забезпечити доступ до даних з будь-якого пристрою у світі.

Блокчейн - це технологія, яка в останні десятиліття заслужила визнання як один із найбільш значущих і перспективних напрямів у сфері інформаційних технологій. Її принципи, такі як децентралізація, прозорість і безпека, надають їй видатних можливостей і застосування в різних галузях. Блокчейн варто розвивати і дедалі частіше використовувати та впроваджувати в проєкти через такі переваги:

- Посилена безпека. Блокчейн забезпечує високий рівень безпеки завдяки розподіленій природі зберігання даних і криптографічним методам шифрування. Це робить його ідеальним для сфер, де конфіденційність і цілісність даних є критично важливими.
- Прозорість і відстежуваність. Усі транзакції та записи в блокчейні загальнодоступні та можуть бути перевірені. Це забезпечує високий ступінь прозорості та дає змогу відстежувати походження даних. Логістичні компанії та виробники можуть використовувати цю особливість для надійнішого обліку та відстеження своїх товарів.
- Децентралізація. Блокчейн усуває необхідність центральних установ або посередників, що може знизити залежність від них і надати більш рівні умови для учасників системи.
- Зниження витрат. За рахунок усунення посередників і автоматизації процесів, блокчейн може знизити операційні витрати. Це особливо актуально для бізнесу, який може заощадити на комісіях за транзакції та управлінні даними.
- Покращена взаємодія. Блокчейн дає змогу учасникам системи взаємодіяти безпосередньо і без необхідності довіри один одному. Це відкриває нові можливості для бізнесу та співпраці.
- Інновації в продуктах і послугах. Блокчейн надихає на створення інноваційних продуктів і послуг, таких як смарт-контракти, децентралізовані додатки (DApps) і токенізовані активи. Ці новаторські інструменти можуть створювати абсолютно нові бізнес-моделі.

Тож блокчейн дає можливість переосмислити і поліпшити існуючі процеси в різних галузях. Його використання варто розглядати як одну з важливих стратегій для поліпшення ефективності, безпеки та інновацій у проєктах і бізнесі.

Виходячи з цієї інформації мною було розроблено методику та алгоритм реалізації програмного застосунку менеджера паролів на основі блокчейн технолгій, а також розроблено відповідне програмне забезпечення.

Цей покращений менеджер паролів, що використовує технологію блокчейн є інноваційним підходом до зберігання та управління паролями. Його використання забезпечує високий рівень безпеки, прозорість і незмінність даних, що є важливими перевагами.

Під час тестування та порівняння розробленого менеджеру паролів з існуючими аналогами було визначено що продукт є цілком конкурентноспроможним, проте, як і будь-який програмний продукт, він має свої сильні та слабкі сторони, і користувачі повинні враховувати такі фактори під час вибору менеджера паролів. До цих факторів можна віднести:

- Безпека. Однією з ключових переваг представленого рішення є безпека завдяки використанню блокчейн технології. Це може привабити користувачів, які цінують високий рівень захисту своїх даних.
- Прозорість і незмінність даних. Блокчейн забезпечує прозорість і можливість відстеження всіх змін і транзакцій, що може бути важливим для користувачів, які потребують доказу цілісності даних.
- Швидкість обробки і масштабованість. Блокчейн може мати обмеження за швидкістю обробки і масштабованості порівняно з традиційними системами. Це може бути недоліком для користувачів, які очікують миттєвих реакцій і обробки великого обсягу даних.
- Інтерфейс і зручність використання. Традиційні менеджери паролів часто пропонують інтуїтивний інтерфейс і хорошу інтеграцію з різними

платформами, включно з мобільними пристроями. Зручність використання може бути важливим фактором для багатьох користувачів.

- Швидкість зростання і масштабування. У разі збільшення кількості користувачів і обсягу даних, розроблений менеджер паролів може зіткнутися з проблемами масштабованості. Необхідно добре продумати план масштабування, щоб забезпечити ефективне обслуговування всіх користувачів.

Отже, даний менеджер паролів може бути ідеальним вибором для тих, хто ставить безпеку на перше місце і готовий прийняти деякі обмеження в зручності використання програми. Він може бути особливо корисним для користувачів, які зберігають чутливі дані та цінують прозорість і незмінність інформації. Однак, важливо також стежити за його масштабованістю і приділяти увагу зворотному зв'язку користувачів для поліпшення інтерфейсу і функціональності.

В ході виконання кваліфікаційної роботи, було:

1) Проведено аналіз існуючих систем та методик забезпечення безпеки паролів користувачів і оцінки ризиків інформаційної безпеки, що дозволило визначити переваги та недоліки існуючих рішень і визнати подальші напрями дослідження;

2) Розроблено алгоритм менеджера паролів на основі блокчейн технологій, що дозволило реалізувати з використанням мови програмування Python та веб-додатку Jupyter Notebook застосунок.

3) Проведено оцінку доцільності отриманих результатів, що дозволило впевнитися в коректності роботи застосунку та виконати порівняння з існуючими системами управління паролями в інформаційних мережах.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Сучасний стан інформаційної безпеки [Електронний ресурс] — URL: http://www.itsmonline.ru/phparticles/show_news_one.php?n_id=294

2. Basic Network Attacks in Computer Network [Електронний ресурс] — URL: https://www.geeksforgeeks.org/basic-network-attacks-in-computer-network/

3. Blockchain Technology Needs to Be Changing Education [Електронний ресурс] / Medium – платформа для соціальної журналістики. – Режим доступу: https://medium.com/age-of-awareness/blockchain-technologyneeds-to-bechanging-education-28324281e2

4. Bitcoin's Largest Competitor Hacked: Over $59 Million "Ethers" Stolen In Ongoing Attack [Електронний ресурс] / Режим доступу: https://www.zerohedge.com/news/2020-06-17/bitcoins-largest-competitorhackedover-59-million-ethers-stolen-ongoing-attack

5. Дереко В.Н. Теоретико-методологічні засади класифікації загроз об’єктам інформаційної безпеки / В. Дереко // Інформаційна безпека людини, суспільства, держави. – 2015. – № 2 (18). – С. 16–22, [Текст]

6. Ben-David A. FairplayMP: a system for secure multi-party computation / A. BenDavid, N. Nisan, B. Pinkas // ACM CCS 2018. – 2018. – P. 257 – 266.

7. Курок Р.О., Національна академія Служби безпеки України, інформаційна безпека в діяльності СБ України: сучасні проблеми та шляхи їх вирішення, [Текст]

8. Camenisch J. Concepts and languages for privacy-preserving attribute-based authentication / J. Camenisch, M. Dubovitskaya, R. Enderlein, A. Lehmann, G. Neven, C. Paquin, F. Preiss // IFIP Working Conference on Policies and Research in Identity Management. – Vol. 19. – 2020. – P. 25 – 44.

9. Cachin C. Blockchain Consensus Protocols in the Wild / C. Cachin, M. Vukolic. // Proceedings of 31th International Symposium on Distributed Computing. – 2021., 505 p.

10. Бурячок В. Л., Толубко В.Б., Хорошко В. О., Толюпа С.В.. «Інформаційна та кібербезпека: соціотехнічний аспект. [Підручник]». - 2015.

11. Горбенко І. Д. Гриненко Т. О. Захист інформації в інформаційнотелекомунікаційних системах: Навч. посібник. Ч.1. Криптографічний захист інформації - Харків: ХНУРЕ, 2004 - 368 с.

12. Singh, P., Basit, A., Kumar, N. C., & Venkaiah, V. C. (2019). Towards a Hybrid Public Key Infrastructure (PKI): A Review. Cryptology ePrint Archive.

13. Інформаційна безпека (соціально-правові аспекти) / [В. Остроухов, В. Петрик, М. Присяжнюк та ін.] ; за ред. Є.Д. Скулиша. – К. : КНТ, 2010. – 776 с., с. 89, [Текст]

14. Yunakovsky, S. E., Kot, M., Pozhar, N., Nabokov, D., Kudinov, M., Guglya, A., ... & Fedorov, A. K. (2021). Towards security recommendations for public-key infrastructures for production environments in the post-quantum era. EPJ Quantum Technology, 8(1), 14.

15.

16. Barton, B.F., Barton, M.S.: User-friendly password methods for computer-mediated information systems. Comput. Secur. 3(3), 186–195 (1984)

17. Florencio, D., Herley, C.: Where do security policies come from? New York, NY, USA (2010).

18. K. Bhargavan and A. Delignat-Lavaud. Web-based attacks on host-proof encrypted storage. In Proc. of WOOT, 2012.

19. D. Akhawe, A. Barth, P. E. Lam, J. Mitchell, and D. Song. Towards a formal foundation of web security. In Proceedings of the 23rd IEEE Computer Security Foundations Symposium, 2010.

20. Критерії оцінки захищеності інформації в комп’ютернихсистемахвіднесанкціонованого доступу НД ТЗІ 2.5-004-99. Режим доступу: https://tzi.ua/assets/files/НД-ТЗІ-2.5-004-99.pdf?fbclid=IwAR16Qka92G63wtjvFfHcK5rALmf2z0iKjdO0rN6f005_fxovlJX3-RtIpqk

21. Гребенніков В.В. - Комплексні системи захисту інформації: проектування,впровадження,супровід. Збірник лекцій. 2013. 161 с. - Режимдоступу:https://mybook.ru/author/vadimgrebennikov-3/kompleksni-sistemi-zahistuinformaciyi-proektuvann/

22. Класифікація автоматизованих систем і стандартні функціональні профілізахищеності оброблюваної інформації від несанкціонованого доступу. Режимдоступу: https://tzi.ua/assets/files/НД-ТЗІ-2.5-005--99.pdf 6.

23. Cyber Security requires strong UX [Електронний ресурс]: - Режим доступу до ресурсу: https://hackernoon.com/cyber-security-requires-an-important-ingredient-s trong-ux-d0727a0c076

24. Нові стандарти для безпарольної аутентифікації [Електронний ресурс]: - Режим доступу до ресурсу: https://habr.com/1cloud/blog/353966/

25. Перспективи розвитку мережі «Блокчейн» [Електронний ресурс]. Режим доступу :http://elibrary.kubg.edu.ua/id/eprint/25667/1/%D0%A1%D0%BF%D0%B0%D1%81%D1%96%D1%82%D1%94%D0%BB%D1%94%D0%B2%D0%B0.pdf

26. . “10 кращих програм для зберігання паролів” [Електронний ресурс] – Режим доступу: https://root-nation.com/ua/soft-ua/ua-10-krashhix-program-dlyazberigannya-paroliv/

27. Фахад Алодьяни. Менеджери паролів - це все про довіру та прозорість. Школа комп'ютерних наук та інформатики, Кардифський університет, Куінс-білдінг, Кардіфф, Великобританія, 2020. № 189 с. 189. https://www.mdpi.com/1999-5903/12/11/189

28. Брюс Шнаейр, Прикладна криптографія. Протоколи, алгоритми, вихідні тексти на мові Сі. Москва, 2002. 610 с.

29. Про Основні засади розвитку інформаційного суспільства [Електронний ресурс]. — [S. l. : s. n.]. — URL: http://zakon4.rada.gov.ua/laws/show/537-16

30. Войтович В.С., Гриник Р.О. Основні безпекові проблеми кіберпростору України. Зб. тез доповідей Міжнародна науково-практична

конференція “Інформаційна безпека в сучасному суспільстві” (м. Львів, 24-25 листопада 2016 р.). Львів : ЛДУБЖД, 2016. С. 23–24.

31. Заник О., Ткачук Р. Вплив людського фактору на системи організації інформаційної безпеки. Зб. тез доповідей V Всеукр. наук.-практ конф. молодих 8 учених, студентів і курсантів “Інформаційна безпека та інформаційні технології” (м. Львів, 26 листопада 2020 р.). Львів : ЛДУБЖД, 2020. С. 21–22.

32. Cybersecurity Best Practices Guide For IIROC Dealer Members - Investment Industry Regulatory Organization of Canada, 2015. - 53 pp.

33. Zhu S.Y. Guide to Security in SDN and NFV: Challenges, Opportunities, and Applications. / Shao Ying Zhu, Sandra Scott-Hayward, Ludovic Jacquin, Richard Hill (Editors). - Springer International Publishing AG 2017, Gewerbestrasse 11, 6330 Cham, Switzerland, 2017. - 331 pp.

34. Blockchain guide for aspiring developers SINGHAL BIKRAMADITYA , DAMEJA GAUTAM , PANDA PRIYANSU SEHAR, BHY APRESS, 2020. - 288pp.

35. Інформаційна безпека людини як споживача телекомунікаційних послуг: Монографія / І.В. Арістова, Д. В. Сулацький ; НДІ інформатики і права НАПрН України. — К. : Право України; Х. : Право, 2013. — 184 с.

36. Aumasson Jean-Philippe. Serious Cryptography_A Practical Introduction to Modern Encryption

37. Aumasson Jean-Philippe. Crypto Dictionary_500 Tasty Tidbits for the Curious Cryptographer

38. Mihailescu M.I., Nita S.L. Pro Cryptography and Cryptanalysis with C++20_Creating and Programming Advanced Algorithms

39. Achary R. Cryptography And Networking Security_An Introduction

40. Ellison, C., & Schneier, B. (2000). Ten risks of PKI: What you're not being told about public key infrastructure. Comput Secur J, 16(1), 1-7.

41. Diaz-Sanchez, D., Marín-Lopez, A., Mendoza, F. A., Cabarcos, P. A., & Sherratt, R. S. (2019). TLS/PKI challenges and certificate pinning techniques for IoT

and M2M secure communications. IEEE Communications Surveys & Tutorials, 21(4), 3502-3531.

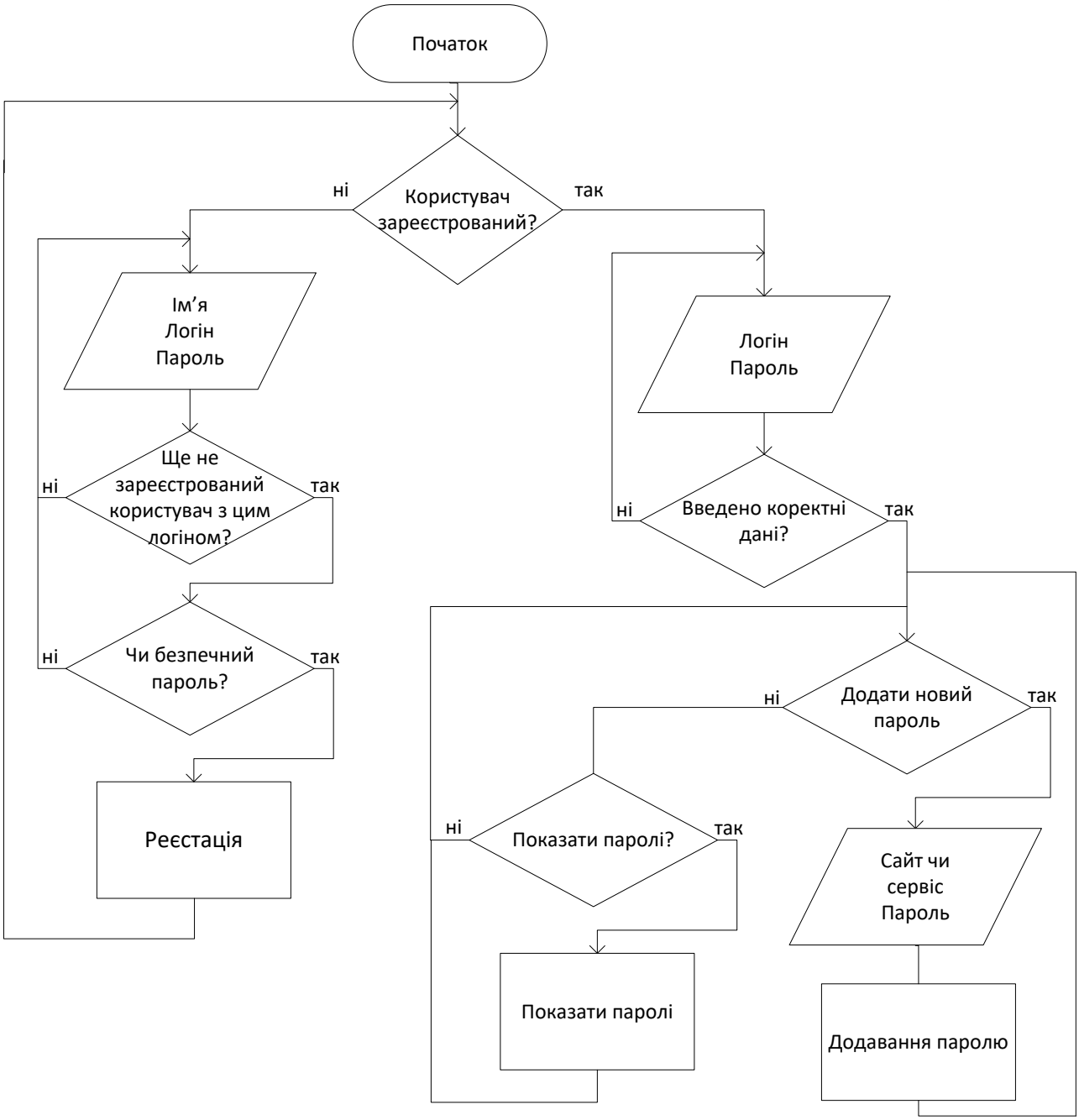
42. Höglund, J. (2023). Public Key Infrastructure and its applications for resource-constrained IoT (Doctoral dissertation, Acta Universitatis Upsaliensis).

43. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. [Електронний ресурс] – Режим доступу: https://bitcoin.org/bitcoin.pdf

44. Gatteschi, V.; Lamberti, F.; Demartini, C.; Pranteda, C.; Santamaría, V. Blockchain and smart contracts for insurance: Is the technology mature enough? Future Internet 2018, 10, 20. [Електронний ресурс] – Режим доступу: https://www.mdpi.com/1999-5903/10/2/20

45. “Менеджер паролів” [Електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/%D0%9C%D0%B5%D0%BD%D0%B5%D0%B4%D0%B6%D0%B5%D1%80_%D0%BF%D0%B0%D1%80%D0%BE%D0%BB%D1%96%D0%B2

Додаток А


Початок
Користувач зареєстрований?
ні
так
Ім’я
Логін
Пароль
Ще не зареєстрований користувач з цим логіном?
ні
так
Чи безпечний пароль?
ні
так
Реєстація
Логін
Пароль
Введено коректні дані?
ні
так
Додати новий пароль
ні
так
Показати паролі?
ні
так
Показати паролі
Сайт чи сервіс
Пароль
Додавання паролю

Додаток Б

```
import PySimpleGUI as sg
import hashlib
import time
import requests
import cryptocode


all_users = {}
current_user = ''

sg.theme('Reddit')

class Block:
    def __init__(self, index, previous_hash, timestamp, data, hash):
        self.index = index
        self.previous_hash = previous_hash
        self.timestamp = timestamp
        self.data = data
        self.hash = hash


def calculate_hash(index, previous_hash, timestamp, data):
    return
hashlib.sha256(f'{index}{previous_hash}{timestamp}{data}'.encode('utf-
8')).hexdigest()

def create_genesis_block():
    return Block(0, '0', time.time(), 'Genesis Block', calculate_hash(0, '0',
time.time(), 'Genesis Block'))
```

```
def create_new_block(prev_block, data):
    index = prev_block.index + 1
    timestamp = time.time()
    hash = calculate_hash(index, prev_block.hash, timestamp, data)
    return Block(index, prev_block.hash, timestamp, data, hash)


class BlockchainPasswordManager:
    def __init__(self):
        self.blockchain = [create_genesis_block()]
        self.passwords = {}

    def add_password(self, user, site_name, password):
        prev_block = self.blockchain[-1]
        data = f'{user}.{site_name}:{password}'
        new_block = create_new_block(prev_block, data)
        self.blockchain.append(new_block)
        if user not in self.passwords:
            self.passwords[user] = {}
        self.passwords[user][site_name] = cryptocode.encrypt(password, user)

    def get_password(self, user):
        user_passwords = self.passwords.get(user, {})
        return "\n".join([f"{site}: {cryptocode.decrypt(password, user)}" for site,
password in user_passwords.items()])


class Node:
    def __init__(self):
```

```
        self.blockchain = [create_genesis_block()]

    def add_block(self, new_block):
        self.blockchain.append(new_block)

    def get_blockchain(self):
        return self.blockchain

# Функция для добавления блока в блокчейн
def add_block_to_chain(node, user, site_name, password):
    prev_block = node.blockchain[-1]
    data = f'{user}.{site_name}:{password}'
    new_block = create_new_block(prev_block, data)
    node.add_block(new_block)

# Инициализация узлов (пример)
nodes = [Node() for _ in range(3)]  # Создание трех узлов

# Функция для распределения данных между узлами
def distribute_data(user, site_name, password):
    for node in nodes:
        add_block_to_chain(node, user, site_name, password)



def check_password(password):
    digit = '1234567890'
    uppers = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
    lowers = 'abcdefghijklmnopqrstuvwxyz'
```

```
    symbol = '!@#$%^&*()-+'
    acceptable = digit+uppers+lowers+symbol
    rez = ''
    passwd = set(password)
    if any(char not in acceptable for char in passwd):
        rez = 'Помилка. Заборонений символ.'
    else:
        recommendations = []
        if len(password) < 8:
            recommendations.append(f'збільшити кількість символів - {8-
len(password)}')
        for what, message in ((digit, 'цифру'),
                              (symbol, 'спеціальний символ'),
                              (uppers, 'велику літеру'),
                              (lowers, 'маленьку літеру')):
            if all(char not in what for char in passwd):
                recommendations.append(f'додати 1 {message}')

        if recommendations:
            rez = 'Слабкий пароль. Рекомендації:', ', '.join(recommendations)
        else:
            rez = 'Зареєстровано!'
    return rez

def window_signin():
    layout = [
        [sg.Text('')],
        [sg.Text("Вітаємо в\nменеджері паролів!", font=('Arial', 42),
justification='center', expand_x=True)],
```

```
        [sg.Text('')],
        [sg.Text('Логін', justification='left', expand_x=True),
sg.InputText(key='signin_input_login')],
        [sg.Text('Пароль', justification='left', expand_x=True),
sg.InputText(key='signin_input_password')],
        [sg.Text('')],
        [sg.Button('Увійти', size=20, font=("Arial", 14), key='signin')],
        [sg.Text('')],
        [sg.Button('Зареєструватись', font=("Arial", 10),
key='signin_Registration')]
    ]
    return sg.Window('Менеджер паролів Демченко Крістіна 2023', layout,
finalize=True, size=(550, 400), font='Arial 12', element_justification='center')


def window_Registration():
    layout = [
        [sg.Text('')],
        [sg.Text("Реєстрація", font=('Arial', 42), justification='center',
expand_x=True)],
        [sg.Text('')],
        [sg.Text("Ім\'я", justification='left', expand_x=True),
sg.InputText(key='Registration_input_name')],
        [sg.Text('Логін', justification='left', expand_x=True),
sg.InputText(key='Registration_input_login')],
        [sg.Text('Пароль', justification='left', expand_x=True),
sg.InputText(key='Registration_input_password')],
        [sg.Text('')],
```

```
        [sg.Button('Зареєструватись', size=20, font=("Arial", 14),
key='Registration')],
        [sg.Text('')],
        [sg.Button('Повернутись назад', font=("Arial", 10),
key='Registration_signin')]
    ]
    return sg.Window('Менеджер паролів Демченко Крістіна 2023', layout,
finalize=True, size=(550, 400), font='Arial 12', element_justification='center')

def window_Hello(name_for_hello):
    layout = [
        [sg.Text(f'Привіт, {name_for_hello}!', font=('Arial', 42),
justification='center', expand_x=True)],
        [sg.Text('')],
        [sg.Text('')],
        [sg.Button('Подивитись пароль', size=20, font=("Arial", 14),
key='check_pass')],
        [sg.Text('')],
        [sg.Button('Додати новий пароль', size=20, font=("Arial", 14),
key='new_pass')]
    ]
    return sg.Window('Менеджер паролів Демченко Крістіна 2023', layout,
finalize=True, size=(550, 400), font='Arial 12', element_justification='center')



def window_new():
    layout = [
        [sg.Text('')],
```

Продовження додатку Б

```
        [sg.Text('')],
        [sg.Text('Назва сайту', justification='left', expand_x=True),
sg.InputText(key='new_site_name_save')],
        [sg.Text('')],
        [sg.Text('Пароль', justification='left', expand_x=True),
sg.InputText(key='new_pass_save')],
        [sg.Text('')],
        [sg.Button('Додати новий пароль', size=20, font=("Arial", 14),
key='save_new_pass')],
        [sg.Text('')],
        [sg.Button('Повернутись назад', font=("Arial", 10), key='new_to_hello')]
    ]
    return sg.Window('Менеджер паролів Демченко Крістіна 2023', layout,
finalize=True, size=(550, 400), font='Arial 12', element_justification='center')



w1, w2, w3, w4 = window_signin(), None, None, None

manager = BlockchainPasswordManager()

while True:
    window, event, values = sg.read_all_windows()
    if event == sg.WINDOW_CLOSED:
        break
    if window == w1:
        if event == 'signin_Registration':
            w2 = window_Registration()
            w1.hide()
```

Продовження додатку Б

```
        elif event == 'signin':
            if (str(values['signin_input_login']) in all_users) and
(all_users[str(values['signin_input_login'])]['password'] ==
hashlib.sha1(str(values['signin_input_password']).encode()).hexdigest()):
                current_user = str(values['signin_input_login'])
                w3 = window_Hello(all_users[current_user]['name'])
                w1.hide()
            else:
                sg.popup('Введено не вірно! Спробуйте ще!', title='Помилка')
    elif window == w2:
        if event == 'Registration_signin':
            w1 = window_signin()
            w2.hide()
        elif event == 'Registration':
            if str(values['Registration_input_login']) not in all_users.keys():
                if check_password(str(values['Registration_input_password'])) ==
'Зареєстровано!':
                    all_users[str(values['Registration_input_login'])] = {'name':
str(values['Registration_input_name']), 'password':
hashlib.sha1(str(values['Registration_input_password']).encode()).hexdigest()}

sg.popup(check_password(str(values['Registration_input_password'])),
title='Реєстрація успішна')
                else:

sg.popup(check_password(str(values['Registration_input_password'])),
title='Помилка')
            else:
```

Продовження додатку Б

```
                sg.popup('Користувач з цим логіном вже зареєстрований!',
        title='Помилка')
    elif window == w3:
        if event == 'new_pass':
            w4 = window_new()
            w3.hide()
        elif event == 'check_pass':
            sg.popup(manager.get_password(current_user), title='Список Ваших
        паролей')
    elif window == w4:
        if event == 'save_new_pass':
            manager.add_password(current_user,
        str(values['new_site_name_save']), str(values['new_pass_save']))
        elif event == 'new_to_hello':
            w3 = window_Hello(all_users[current_user]['name'])
            w4.hide()



window.close()
```