Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)
Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр
(назва освітнього ступеня)

на тему: Система автоматизованого резервного копіювання даних у хмару для малого бізнесу на базі Google Drive API

Виконав: студент VI курсу, групи СНнм-61
спеціальності 122 Комп'ютерні науки
(шифр і назва спеціальності)

_____ Ципняк Д.О.
(підпис) (прізвище та ініціали)

Керівник _____ Литвиненко Я.В.
(підпис) (прізвище та ініціали)

Нормоконтроль _____ Дуда О.М.
(підпис) (прізвище та ініціали)

Завідувач кафедри _____ Боднарчук І.О.
(підпис) (прізвище та ініціали)

Рецензент _____ Луцик Н.С.
(підпис) (прізвище та ініціали)

Тернопіль
2025

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Боднарчук І.О.
(підпис) (прізвище та ініціали)
«____»_____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Магістр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студенту Ципняк Денис Олександрович
(прізвище, ім'я, по батькові)

1. Тема роботи Система автоматизованого резервного копіювання даних у хмару для малого бізнесу на базі Google Drive API

Керівник роботи Доктор технічних наук, доцент кафедри КН Литвиненко Ярослав Володимирович
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 29 » листопада 2024 року № 4/7-1139

2. Термін подання студентом завершеної роботи 2025р.

3. Вихідні дані до роботи Наукові публікації та технічна документація з питань резервного. копіювання даних для малого бізнесу, принципів хмарних технологій та їх інтеграції, Застосування машинного навчання для оптимізації зберігання та безпеки даних. Документація Google Drive API, Python, PyQt6, бібліотек для стиснення, шифрування та інструментів машинного навчання. Стандарти OAuth 2.0 та AES-256.

4. Зміст роботи (перелік питань, які потрібно розробити)
Вступ. 1 Аналіз проблеми резервного копіювання даних для малого бізнесу та обґрунтування вибору технологічної платформи. 2 Проєктування архітектури та методів функціонування системи автоматизованого резервного копіювання. 3 Програмна реалізація та експериментальне дослідження прототипу системи автоматизованого резервного копіювання 4 Охорона праці та безпека в надзвичайних ситуаціях. Висновки. Додатки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
1 Титульна сторінка. 2 Мета дослідження. 3 Завдання дослідження. 4. Об'єкт, предмет Дослідження та наукова новизна. 5 Актуальність дослідження 6 Обґрунтування вибору Google Drive API. 7 Архітектура розробленої системи "GhostCopy". 8 Методи функціонування системи. 9 Програмна реалізація та інтерфейс користувача "ghostcopy". 10 Експериментальне дослідження: методика та результат 11. Практичне значення отриманих результатів. 12 Висновки. 13 Завершальний слайд

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Сенчишин В.С., доцент		
Безпека в надзвичайних ситуаціях	Клепчик В.М., ст. викладач		

7. Дата видачі завдання 29 листопада 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	29.11.2024	
2.	Підбір та опрацювання наукових публікацій, збір	02.12.2024-10.01.2025	
	даних по темі роботи		
3.	Виконання дослідження згідно теми кваліфікаційної	13.01.2025-14.03.2025	
	Роботи		
4.	Оформлення розділу «Аналіз проблеми резервного	17.03.2025-28.03.2025	
	копіювання даних для малого бізнесу та		
	обґрунтування вибору технологічної платформи»		
5.	Оформлення розділу «Проєктування архітектури та	31.03.2025-11.04.2025	
	Методів функціонування системи автоматизованого		
	резервного копіювання»		
6.	Оформлення розділу «Програмна реалізація	14.04.2025-25.04.2025	
	реалізація та експериментальне дослідження		
	прототипу системи автоматизованого		
	резервного копіювання»		
7.	Виконання завдання до підрозділу «Охорона праці»	28.04.2025-02.05.2025	
8.	Виконання завдання до підрозділу «Безпека в	28.04.2025-02.05.2025	
	надзвичайних ситуаціях»		
9.	Оформлення кваліфікаційної роботи	05.05.2025-09.05.2025	
10.	Нормоконтроль	12.05.2025-15.05.2025	
11.	Перевірка на плагіат	16.05.2025	
12.	Попередній захист кваліфікаційної роботи	19.05.2025	
13.	Захист кваліфікаційної роботи	29.05.2025	

Студент _____ Ципняк Д.О.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Литвиненко Я.В.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Система автоматизованого резервного копіювання даних хмару для малого бізнесу на базі Google Drive API // Кваліфікаційна робота освітнього рівня «Магістр» // Ципняк Денис Олександрович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СНнм-61 // Тернопіль, 2025 // С. 61, рис. – 4, табл. – 1, кресл. – 13, додат. – 3, бібліогр. – 60.

Ключові слова: резервне копіювання, малий бізнес, хмарне сховище, google drive api, машинне навчання, автоматизація, безпека даних, оптимізація.

Кваліфікаційна робота присвячена розробці системи автоматизованого інтелектуального резервного копіювання даних для малого бізнесу з використанням Google Drive API. У першому розділі кваліфікаційної роботи описані потреби малого бізнесу в захисті даних. Висвітлено особливості резервного копіювання. Розглянуто сучасні хмарні сервіси з програмною інтеграцією. Проаналізовано існуючі рішення та обґрунтовано вибір Google Drive API як основної платформи. У другому розділі кваліфікаційної роботи розроблено модульну архітектуру системи. Досліджено теоретичні основи застосування машинного навчання для пріоритезації даних. Подано функціональну логіку автоматизованого копіювання та опис структури графічного інтерфейсу користувача. У третьому розділі кваліфікаційної роботи описано реалізацію ключових модулів системи, включаючи взаємодію з Google Drive API, стиснення та шифрування даних. Проаналізовано продуктивність, ефективність стиснення та надійність системи за результатами тестування. Проведено експериментальну оцінку працездатності прототипу на змодельованому наборі даних.

ANNOTATION

Automated Cloud Data Backup System for Small Businesses Using Google Drive API // The educational level "Master" qualification work // Tsypniak Denys // Ternopil Ivan Pulyuy National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Computer Science, SNnm-61 group // Ternopil, 2025 // P. 61, fig. – 4, tables – 1, posters – 13, annexes – 3, ref. – 60.

Keywords: backup, small business, cloud storage, google drive api, machine learning, automation, data security, optimization.

The qualification work is dedicated to the development of an automated intelligent data backup system for small businesses using the Google Drive API. In the first section of the qualification work, the needs of small businesses in data protection are described, the specifics of backup processes are highlighted, and modern cloud services with programmatic integration are reviewed. Existing solutions are analyzed, and the choice of Google Drive API as the primary platform is justified. In the second section of the qualification work, a modular system architecture is developed, the theoretical foundations of applying machine learning for data prioritization are investigated, and the functional logic of automated backup and the structure of the graphical user interface are presented. In the third section of the qualification work, the implementation of the system's key modules, including interaction with the Google Drive API, data compression, and encryption, is described. The system's performance, compression efficiency, and reliability are analyzed based on testing results. An experimental evaluation of the prototype's functionality was conducted on a simulated dataset. Object of research: data backup processes. Subject of research: methods, tools, and algorithms for creating an intelligent automated data backup system based on cloud technologies and machine learning for the needs of small businesses.

СПИСОК СКОРОЧЕНЬ

API – (англ. Application Programming Interface) – інтерфейс прикладного програмування.

ГІК – Графічний інтерфейс користувача.

ІТ – Інформаційні технології.

ML – (англ. Machine Learning) – Машинне навчання.

NAS – (англ. Network Attached Storage) – Мережеве сховище даних.

ЗМІСТ

ВСТУП

У сучасному цифровому середовищі, де інформація стала одним з найважливіших активів для функціонування та розвитку підприємств, питання її захисту, збереження та забезпечення безперервного доступу набувають першочергового значення [7]. Це особливо актуально для сегменту малого бізнесу, який, попри свою гнучкість та динамічність, часто стикається з обмеженими фінансовими, технічними та кадровими ресурсами для побудови надійної IT-інфраструктури. Загрози втрати даних варіюються від типових операційних інцидентів, таких як випадкове видалення файлів, програмні збої або апаратні поломки носіїв інформації, до значно більш серйозних викликів, включаючи цілеспрямовані кібератаки, дії шкідливого програмного забезпечення (зокрема, програм-вимагачів – ransomware) та стихійні лиха чи надзвичайні ситуації [9]. Статистичні дані досліджень невтішні: значний відсоток малих та середніх підприємств, які зазнали суттєвих втрат інформації, не змогли відновити свою повноцінну діяльність і були змушені припинити існування протягом року після інциденту [41]. Це підкреслює гостру потребу в ефективних та доступних рішеннях для резервного копіювання даних, які б мінімізували ці ризики.

На ринку існує багато рішень для резервного копіювання, починаючи від простих ручних методів і закінчуючи складними корпоративними системами [3, 4]. Однак, як показав аналіз, більшість високо функціональних комерційних продуктів розроблені для великих організацій, вимагають значних інвестицій у ліцензії та інфраструктуру, а також наявності кваліфікованого IT-персоналу для їх впровадження та підтримки [42]. Такі рішення часто є недосяжними або невиправдано дорогими для потреб малого бізнесу. Типові "ручні" підходи, що використовуються малими компаніями через їхню уявну простоту та низьку вартість, є вкрай ненадійними, схильними до помилок, не забезпечують належної періодичності, автоматизації, версійності та зберігання копій поза межами локальної мережі (off-site), що робить їх вразливими до

тих самих інцидентів, від яких покликане захищати резервне копіювання [5]. Використання хмарних технологій, зокрема хмарних файлових сховищ, відкриває нові можливості для вирішення цієї проблеми [20]. Вони пропонують масштабованість, відмовостійкість та доступність даних за гнучкими тарифами [16]. Проте базовий функціонал цих сервісів часто обмежується синхронізацією та спільним доступом, не надаючи необхідних для повноцінного резервного копіювання інструментів, таких як копіювання, централізоване управління, шифрування та інтелектуальна обробка даних [43]. Це зумовлює необхідність розробки спеціалізованого програмного забезпечення, яке б використовувало хмарні сервіси як основу, доповнюючи їх необхідним функціоналом. Особливий інтерес становить інтеграція в такі системи елементів штучного інтелекту, зокрема машинного навчання (ML), для підвищення ефективності процесу копіювання шляхом інтелектуального аналізу даних та прогнозування потреби у створенні резервних копій для оптимізації ресурсів (часу, мережевого трафіку та обсягу хмарного сховища) [22]. Таким чином, існує гостра потреба у створенні інтелектуального, автоматизованого, доступного та простого у використанні рішення для резервного копіювання даних малого бізнесу, яке б поєднувало переваги хмарних технологій з елементами машинного навчання для оптимізації процесу. Актуальність даної роботи полягає у розробці саме такого рішення.

Мета дослідження: Розробка інтелектуальної автоматизованої системи резервного копіювання даних для малого бізнесу з використанням платформи Google Drive API та алгоритмів машинного навчання для забезпечення ефективного та гнучкого захисту даних.

Для досягнення поставленої мети було сформульовано наступні **завдання дослідження**:

1. Проаналізувати типові потреби малого бізнесу щодо захисту та резервного копіювання даних, а також оцінити обмеження існуючих на ринку рішень.

2. Провести порівняльний огляд сучасних хмарних сервісів зберігання даних з можливістю програмної інтеграції та обґрунтувати вибір Google Drive API як основної платформи.

3. Спроєктувати логічну та програмну архітектуру інтелектуальної системи резервного копіювання, визначивши взаємодію її ключових компонентів, включаючи модулі моніторингу, аналізу з ML, обробки даних та взаємодії з хмарою.

4. Дослідити теоретичні підходи методів машинного навчання та обрати, з метою розробки функціоналу інтелектуальної пріоритезації даних для оптимізації процесу резервного копіювання.

5. Реалізувати ключові програмні модулі системи, зокрема модуль взаємодії з Google Drive API, функціонал автоматизованого копіювання та інтелектуального аналізу даних.

6. Розробити зручний та інтуїтивно зрозумілий графічний інтерфейс користувача для налаштування та управління системою.

7. Провести експериментальне тестування розробленої системи для оцінки її продуктивності, надійності та ефективності інтелектуального компонента в умовах, що моделюють діяльність малого підприємства.

8. Визначити напрями подальшого вдосконалення та масштабування системи.

Об'єкт дослідження: Процеси резервного копіювання даних.

Предмет дослідження: Методи, засоби та алгоритми створення інтелектуальної автоматизованої системи резервного копіювання даних на базі хмарних технологій та машинного навчання для потреб малого бізнесу.

Наукова новизна отриманих результатів полягає у розробці та обґрунтуванні підходу до інтелектуальної пріоритезації даних для резервного копіювання на основі застосування алгоритмів машинного навчання, який дозволяє адаптувати стратегію копіювання до реальної цінності та динаміки даних для малого бізнесу в умовах обмежених ресурсів хмарного сховища. Вперше запропоновано архітектуру та реалізовано прототип системи, що по-

єднує доступність Google Drive API з інтелектом ML для оптимізації процесу резервного копіювання саме в цьому сегменті ринку.

Практична цінність отриманих результатів полягає у розробці доступного, надійного, ефективного та простого у використанні програмного інструменту, який може бути безпосередньо впроваджений малими підприємствами для забезпечення захисту їхніх цифрових активів, зниження ризиків втрати даних та мінімізації потенційних збитків.

Публікації. Основні результати наукової роботи опубліковано у двох конференції (Див. додатки А).

Структура роботи: Кваліфікаційна робота загальним обсягом 75 сторінки складається зі вступу, чотирьох розділів, висновків, списку використаних джерел та додатків. У вступі обґрунтовано актуальність теми, визначено мету, завдання, об'єкт, предмет, наукову новизну та практичну цінність дослідження. У першому розділі проводиться аналіз проблемної області, огляд існуючих рішень та обґрунтовується вибір технологічної платформи. Другий розділ присвячений теоретичному обґрунтуванню, проєктуванню архітектури системи та вибору методів машинного навчання. У третьому розділі детально описується програмна реалізація системи та представлено результати експериментальної оцінки її ефективності. Четвертий розділ розглядає питання охорони праці та безпеки життєдіяльності. У висновках підсумовано результати виконаної роботи.

РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМИ РЕЗЕРВНОГО КОПІЮВАННЯ ДАНИХ ДЛЯ МАЛОГО БІЗНЕСУ ТА ОБҐРУНТУВАННЯ ВИБОРУ ТЕХНОЛОГІЧНОЇ ПЛАТФОРМИ

В умовах сучасного розвитку цифрових технологій та зростаючої залежності бізнес-процесів від електронної інформації, питання забезпечення надійного зберігання та захисту даних набуває стратегічного значення [7]. Особливо вразливим сегментом у цьому контексті є малий бізнес, який, з одного боку, оперує значним обсягом критично важливої інформації (фінансові дані, клієнтські бази, проєктна документація), а з іншого – часто не має ресурсів, співмірних з великими корпораціями, для створення та підтримки комплексних систем інформаційної безпеки та резервного копіювання [42]. Аналіз поточного стану справ у сфері захисту даних для малого бізнесу та огляд існуючих рішень є необхідним кроком для обґрунтування потреби у розробці нових, більш адаптованих інструментів.

1.1 Огляд існуючих рішень для резервного копіювання та аналіз потреб малого бізнесу

Малі підприємства функціонують в умовах динамічного середовища, де швидкість, гнучкість та оптимізація витрат є ключовими факторами успіху. Втрата даних може призвести до серйозних фінансових збитків, репутаційних втрат, юридичних наслідків і навіть до повного припинення діяльності компанії [2, 41]. Типові причини втрати даних включають: апаратні збої (відмова жорстких дисків, серверів), програмні помилки, випадкове видалення або перезапис файлів співробітниками, а також зовнішні загрози, такі як віруси (особливо програми-здирники – ransomware), хакерські атаки, крадіжка обладнання, пожежі чи затоплення офісних приміщень [1, 9]. Аналіз потреб малого бізнесу у сфері резервного копіювання та захисту даних виявляє низку специфічних вимог [3, 42], серед яких першочерговими є:

- Простота використання та налаштування: Враховуючи часто обмежену або відсутню кількість штатних IT-спеціалістів, система має бути максимально інтуїтивно зрозумілою та не вимагати глибоких технічних знань для встановлення, конфігурації та щоденного моніторингу [47].
- Доступна вартість: Бюджети малих підприємств на IT, як правило, обмежені. Рішення має бути економічно вигідним як на етапі придбання/впровадження, так і в процесі експлуатації (вартість зберігання, ліцензій, підтримки).
- Високий ступінь автоматизації: Процес резервного копіювання має відбуватися автоматично за заданим розкладом або за подіями (зміни файлів), мінімізуючи втручання користувача та усуваючи вплив людського фактору.
- Надійність та гарантія збереження даних: Користувач повинен бути впевнений, що дані успішно копіюються та можуть бути відновлені в будь-який момент. Необхідні механізми перевірки цілісності копій [5].
- Безпека даних: Дані повинні бути захищені від несанкціонованого доступу як під час передачі (використання захищених протоколів), так і під час зберігання (шифрування) [10, 46].
- Гнучкість: Важливою є можливість вибору конкретних файлів або директорій для копіювання, налаштування політик зберігання (версійності, терміну зберігання) та визначення виключень.
- Можливість відновлення: Процес відновлення даних, як повний, так і вибірковий (окремих файлів), має бути простим та швидким [6].

Існуючі підходи та рішення для резервного копіювання, що застосовуються на практиці, можна узагальнити наступним чином:

- Ручне копіювання на зовнішні носії (HDD, SSD, USB-флешки): Цей метод є найпростішим з точки зору початкових витрат, проте є вкрай ненадійним через високу залежність від людського фактору. Відсутність автоматизації, складнощі з дотриманням регулярності, відсутність версійності та

зберігання копій переважно локально роблять його вразливим до більшості типових загроз.

- Локальні мережеві сховища (NAS): NAS надають переваги у швидкості доступу та відновлення даних у локальній мережі та повний контроль над фізичним розташуванням даних. Однак вони вимагають значних початкових інвестицій, технічного обслуговування та вразливі до фізичних катастроф та локальних кібератак, якщо не інтегровані з рішеннями off-site зберігання.
- Спеціалізоване комерційне програмне забезпечення для резервного копіювання (наприклад, Acronis, Veeam Backup & Replication): Ці продукти пропонують багатий функціонал, включаючи інкрементне/диференційне копіювання, планування, версійність, централізоване управління [3, 4]. Вони є надійними та багатофункціональними, але їхня висока вартість та складність налаштування роблять їх менш придатними для малих підприємств без спеціалізованого IT-персоналу.
- Спеціалізовані хмарні сервіси резервного копіювання (наприклад, Backblaze, Carbonite, IDrive): Такі сервіси забезпечують повну автоматизацію та безпечне зберігання копій off-site, часто включають версійність та зручні інструменти відновлення [43]. Вони є надійним рішенням, але можуть бути дорожчими за базові хмарні сховища загального призначення, а їхні тарифні плани та функціонал можуть бути менш гнучкими.
- Хмарні файлові сховища загального призначення (наприклад, Google Drive, Microsoft OneDrive, Dropbox) як основа для резервного копіювання: Ці сервіси є широко поширеними та доступними за ціною, часто вже використовуються бізнесом для синхронізації та спільної роботи [11, 18]. Вони надають off-site зберігання та зручний доступ через веб-інтерфейс. Проте, як готові рішення для резервного копіювання, вони мають обмежений функціонал (відсутність вбудованого інкрементного копіювання, дедуплікації, централізованого управління). Це вимагає розробки додаткового програмно-

го забезпечення для реалізації необхідної автоматизації, безпеки (шифрування) та ефективності.

Проведений аналіз існуючих підходів та специфічних потреб малого бізнесу вказує на необхідність розробки рішення, яке б поєднувало економічну доступність та простоту використання хмарних файлових сховищ із необхідним функціоналом автоматизації, безпеки та інтелектуальної обробки даних, адаптованим під умови малих підприємств. Використання наявних хмарних платформ через їх API видається найбільш перспективним шляхом для досягнення цієї мети [16].

1.2 Огляд сучасних хмарних сервісів з програмною інтеграцією та обґрунтування вибору Google Drive API

Вибір хмарної платформи є одним з ключових рішень при розробці системи резервного копіювання, орієнтованої на хмарне зберігання [14, 20]. Важливою є не тільки вартість та обсяг сховища, але й наявність гнучкого програмного інтерфейсу (API), що дозволяє реалізувати специфічну логіку резервного копіювання, відмінну від простої синхронізації. Серед популярних хмарних файлових сховищ, які надають API для розробників, можна виділити декілька ключових гравців.

Google Drive, що входить до складу Google Workspace, широко використовується як приватними, так і бізнес-користувачами. Він надає потужний RESTful API (Google Drive API v3) [11], який має добре документовану бібліотеку для мови програмування Python [13]. Функціональність API дозволяє повноцінно працювати з файлами та папками: завантажувати, оновлювати, видаляти, шукати, отримувати метадані та керувати дозволами. Сервіс пропонує конкурентні тарифи, включаючи значний безкоштовний обсяг, що робить його привабливим для малого бізнесу. Авторизація відбувається за допомогою стандартного протоколу OAuth 2.0 [12], який є безпечним та зрозумілим для більшості користувачів.

Microsoft OneDrive тісно інтегрований з операційною системою Windows та пакетом Microsoft 365. Доступ до функціоналу OneDrive для розробників надається через Microsoft Graph API [17]. Як і Google Drive, він має Python SDK та підтримує основні операції з файлами. OneDrive є популярним серед користувачів екосистеми Microsoft і також пропонує безкоштовний рівень та платні підписки, часто вже наявні у малих компаній. Авторизація також базується на стандартах OAuth 2.0, хоча інтеграція через універсальний Graph API може вимагати розуміння ширшого контексту сервісів Microsoft.

Dropbox є одним з піонерів у галузі хмарних файлових сховищ, відомий своєю простотою та надійністю синхронізації. Він пропонує власний Dropbox API [18] з офіційним Python SDK, який надає необхідну функціональність для роботи з файлами. Тарифікація Dropbox може бути дещо вищою порівняно з конкурентами для подібних обсягів, проте сервіс має сталу аудиторію. Як і попередні, використовує OAuth 2.0 для авторизації.

Amazon S3 (Simple Storage Service) від AWS – це, в першу чергу, сервіс об'єктного зберігання, який широко застосовується для корпоративних бекапів та зберігання великих обсягів даних [19]. Він має надзвичайно потужний та гнучкий API з Python SDK (boto3). Amazon S3 є високо масштабованим та надійним, проте його тарифікація (залежить від обсягу, операцій, трафіку) та процес авторизації (AWS Signature) є більш складними для користувачів без досвіду роботи з хмарними інфраструктурами, порівняно з файловими сховищами загального призначення.

Враховуючи специфічні потреби малого бізнесу, такі як простота використання, доступна вартість та легкість інтеграції, Google Drive API був обраний як найбільш оптимальна платформа для реалізації даної системи. Основні аргументи на користь цього вибору:

- Широке розповсюдження сервісів Google серед бізнес-користувачів значно спрощує процес авторизації [12].

- Відносна простота та вичерпність документації Google Drive API [11], а також наявність зручної клієнтської бібліотеки для Python [13] роблять процес розробки ефективним.
- Конкурентні ціни та безкоштовний обсяг сховища забезпечують економічну вигоду.
- API надає весь необхідний базовий функціонал для побудови рішення, що реалізує резервне копіювання.

Вибір Google Drive API дозволяє створити рішення, яке буде доступним, простим у використанні та зможе ефективно задовольняти потреби малого бізнесу в автоматизованому резервному копіюванні даних.

1.3 Структура даних і джерела для резервного копіювання

Для ефективного проєктування системи резервного копіювання важливо розуміти типову структуру та природу даних, що використовуються в діяльності малого підприємства, а також визначити джерела, звідки ці дані будуть збиратися. Дані малого бізнесу, як правило, є різнорідними та розподіленими по локальних комп'ютерах або невеликих мережевих сховищах.

Типовий набір даних включає різноманітні файли, що створюються та модифікуються в процесі повсякденної діяльності. До них можна віднести:

Офісні документи: Текстові документи, електронні таблиці, презентації, що містять ключову бізнес-інформацію, таку як договори, фінансові звіти, плани, листування.

Фінансові та юридичні документи: Файли з бухгалтерською інформацією, рахунки, акти, банківські виписки, а також установчі та дозвільні документи, що вимагають надійного довгострокового зберігання.

Технічна документація: Посібники, специфікації, проєктні файли та інші документи, пов'язані з діяльністю компанії.

Бази даних та спеціалізовані файли: Файли локальних баз даних, файли конфігурації програмного забезпечення, архіви проєктів.

Медіа-контент: Зображення, відео, аудіофайли, які можуть займати значний обсяг, але їх критичність для щоденної операційної діяльності може варіюватися.

Джерелом даних для системи резервного копіювання виступає локальна файлова система комп'ютера користувача. Система має надавати можливість користувачеві визначити одну або декілька кореневих директорій, вміст яких потребує моніторингу та резервного копіювання. Моніторинг повинен здійснюватися рекурсивно, охоплюючи всі піддиректорії обраних шляхів.

У процесі моніторингу система повинна сканувати визначені директорії, збираючи та відстежуючи ключові метадані кожного файлу. До таких метаданих належать:

- Повний шлях до файлу в локальній файловій системі.
- Назва файлу.
- Розмір файлу.
- Дата та час останньої модифікації.
- Контрольна сума (хеш) вмісту файлу. Використання хешу є важливим для точного виявлення змін, оскільки час модифікації може змінитися з інших причин (наприклад, копіювання файлу без зміни вмісту), або ж вміст може змінитися без негайної зміни часу модифікації на деяких системах чи у певних сценаріях.

Ці метадані необхідно зберігати в локальній базі даних. Вибір на користь легкої вбудовуваної бази даних, такої як SQLite, є обґрунтованим для системи, орієнтованої на малий бізнес, оскільки вона не вимагає окремого серверу, проста у використанні з Python [31, 40] та надійна для локального зберігання невеликих обсягів службової інформації.

Структура даних у локальній базі SQLite повинна включати поля для зберігання відстежуваних метаданих, а також додаткові поля, що відображають стан файлу в контексті резервного копіювання:

- file_path (текст, первинний ключ): Повний унікальний шлях до файлу.

- file_name (текст): Назва файлу.
- file_size (ціле): Розмір файлу у байтах.
- last_modified (текст/datetime): Дата та час останньої модифікації файлу.
- checksum (текст): Контрольна сума (хеш) вмісту файлу.
- last_backup_time (текст/datetime): Дата та час останнього успішного резервного копіювання цього файлу.
- cloud_id (текст): Ідентифікатор файлу у хмарному сховищі Google Drive після успішного копіювання.
- backup_status (текст): Поточний статус файлу в процесі копіювання ('pending', 'in_progress', 'backed_up', 'error', 'deleted_local').
- ml_score (числове): Оцінка важливості файлу, отримана від модуля машинного навчання.
- backup_priority (ціле): Визначений пріоритет копіювання на основі ML-оцінки та інших факторів.

Ця локальна база даних стає центральним елементом для управління процесом резервного копіювання. Вона дозволяє системі ефективно відстежувати зміни (шляхом порівняння поточних метаданих з даними в базі), визначати файли, які потребують копіювання або оновлення, вести історію змін (через відстеження версій у хмарі або зберігання записів про попередні копії), збирати статистику та керувати чергою завдань на копіювання.

1.4 Висновок до першого розділу

У першому розділі дипломної роботи було проведено аналітичний огляд предметної області дослідження, що стало фундаментальним підґрунтям для подальшого проєктування та реалізації системи.

Обґрунтовано гостру актуальність проблеми втрати даних для сегменту малого бізнесу [41], виявлено ключові загрози та проаналізовано специфічні потреби малих підприємств щодо рішень для резервного копіювання, серед

яких вирізняються простота, доступність, автоматизація, надійність та безпека [42]. Критичний огляд існуючих підходів – від ручного копіювання та локальних сховищ до комерційного програмного забезпечення та спеціалізованих хмарних сервісів [3, 43] – показав, що жодне з готових рішень повною мірою не задовольняє унікальні потреби малого бізнесу, часто будучи надто складними, дорогими або недостатньо надійними.

Проведено детальний огляд сучасних хмарних файлових сховищ з точки зору їхньої придатності для програмної інтеграції через API [11, 19]. На основі порівняльного аналізу функціональності API, доступності, вартості та зручності використання, обґрунтовано вибір Google Drive API [11, 13] як найбільш оптимальної платформи для реалізації системи, орієнтованої на малий бізнес. Google Drive поєднує широке поширення, доступні тарифи та достатньо гнучкий API для побудови необхідної логіки.

Визначено типову структуру даних малого підприємства та ідентифіковано локальну файлову систему як основне джерело даних для резервного копіювання. Розроблено підхід до моніторингу файлової системи та визначено ключові метадані файлів, необхідні для відстеження змін. Обґрунтовано вибір легкої вбудовуваної бази даних SQLite для локального зберігання метаданих та службової інформації системи, а також визначено її базову структуру.

Таким чином, аналітична частина роботи дозволила чітко окреслити проблемну область, сформулювати вимоги до майбутньої системи, обґрунтувати вибір технологічної платформи та визначити підхід до роботи з даними, що створює міцний базис для подальшого теоретичного проєктування та практичної реалізації.

РОЗДІЛ 2 ПРОЄКТУВАННЯ АРХІТЕКТУРИ ТА МЕТОДІВ ФУНКЦІОНУВАННЯ СИСТЕМИ АВТОМАТИЗОВАНОГО РЕЗЕРВНОГО КОПІЮВАННЯ

2.1 Архітектура системи та взаємодія її компонентів

В основі розробки системи "GhostCopy" лежить принцип модульної архітектури. Такий підхід забезпечує гнучкість, легкість модифікації окремих компонентів та можливість подальшого масштабування функціоналу, що є критично важливим для створення життєздатного програмного продукту. Модульність дозволяє чітко розмежувати відповідальність між різними частинами системи, спрощуючи процес розробки, тестування та подальшої підтримки [40]. Загальна концепція проєктування була орієнтована на задоволення ключових потреб малого бізнесу, таких як простота використання, високий ступінь автоматизації, надійність збереження даних та доступна вартість.

Структурно система "GhostCopy" складається з декількох взаємопов'язаних модулів, які зображені на рисунку 2.1

Центральним елементом є Модуль Графічного Інтерфейсу Користувача (GUI), реалізований за допомогою бібліотеки PyQt6 (файли visual.py, settings_tab.py, backup_tab.py) [32, 39]. Він слугує основною точкою взаємодії користувача з системою, дозволяючи налаштовувати параметри, обирати файли для копіювання, керувати розкладом та відстежувати процес.

Для управління глобальними параметрами програми, такими як мова, тема оформлення, автозапуск та налаштування ML-компонента, відповідає Модуль Налаштувань та Конфігурації (settings_tab.py, config.py). Важливою його функцією є також управління авторизацією в Google Drive. Вся конфігурація зберігається у файлі config.json [36].

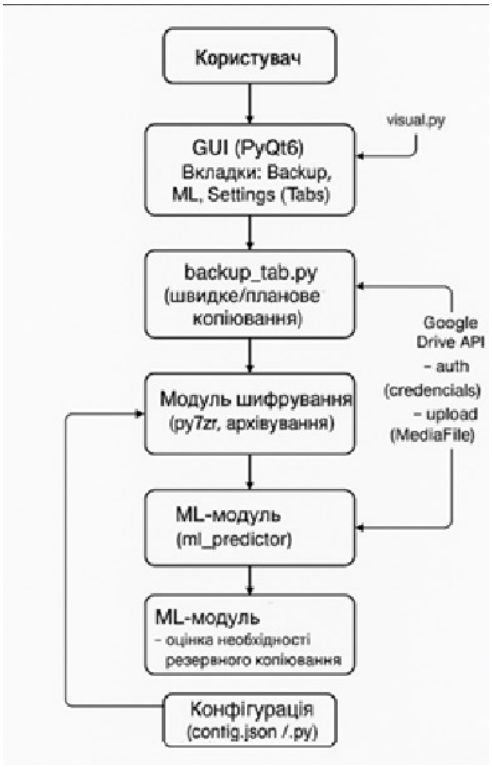

Рисунок 2.1 – Структурна схема архітектури системи "GhostCopy"

Ядром системи є Модуль Резервного Копіювання, логіка якого розподілена між backup_tab.py (управління з боку GUI) та backup_worker.py (безпосереднє виконання операцій). Користувач через інтерфейс обирає файли, вводить пароль шифрування та запускає процес. BackupWorker, що працює в окремому потоці [38], виконує стиснення даних за допомогою py7zr [33], їх шифрування AES-256 та завантаження на Google Drive.

Взаємодія з хмарним сервісом забезпечується Модулем Взаємодії з Google Drive API [11]. Ця функціональність інтегрована в backup_worker.py для завантаження файлів та в settings_tab.py і visual.py для авторизації через OAuth 2.0 [12]. Облікові дані додатку зберігаються в credentials.json, а токени доступу – в token.pickle [35].

Елемент інтелектуальності в систему вносить Модуль Інтелектуального Аналізу (ML Predictor), представлений класом BackupPredictor у файлі ml_predictor.py. На даному етапі він призначений для прогнозування загальної необхідності резервного копіювання, з потенціалом для розширення на пріоритезацію окремих файлів [22, 44].

Функції стиснення та шифрування даних реалізовані в Модулі Обробки Даних, який є частиною BackupWorker. До Допоміжних Модулів належать translations.py для багатомовності та colors.py для управління темами оформлення.

Взаємодія компонентів побудована на чіткому розподілі функцій. Користувач керує системою через GUI, налаштування зберігаються в конфігураційному файлі. При запуску копіювання BackupWorker обробляє файли та завантажує їх у хмару, інформуючи GUI про стан процесу.

При виборі технологій перевага надавалася Python [31] завдяки його гнучкості та великій кількості бібліотек. PyQt6 було обрано для створення GUI [32], Google Drive API – як надійна та доступна платформа для хмарного зберігання [11]. Бібліотека py7zr забезпечує ефективне стиснення та шифрування [33], pickle використовується для серіалізації токенів [35], а JSON – для файлів конфігурації [36].

Проєктні рішення щодо безпеки включають використання OAuth 2.0 [12], шифрування резервних копій AES-256 [10, 46] та локальне зберігання токенів. Інтерфейс користувача розроблено з акцентом на простоту, інтуїтивність, надання чіткого зворотного зв'язку, локалізацію та можливість налаштування теми [47].

2.2 Теоретичні основи та вибір методів машинного навчання для пріоритезації даних

Однією з ключових особливостей розроблюваної системи "GhostCopy" є спроба інтеграції елементів інтелектуального аналізу для оптимізації процесу резервного копіювання. В умовах обмежених ресурсів, таких як час, пропускна здатність інтернет-каналу та обсяг хмарного сховища, особливо актуальною стає задача визначення, які саме дані потребують копіювання в першу чергу, або як часто слід копіювати ті чи інші типи файлів. Машинне

навчання (ML) надає інструменти для вирішення таких завдань шляхом виявлення закономірностей у даних та прийняття рішень на їх основі [22, 23].

У контексті даної системи, ML-компонент покликаний надавати додаткову інформацію для прийняття більш обґрунтованих рішень щодо резервного копіювання [44, 45]. Хоча поточна реалізація ml_predictor.py передбачає базовий функціонал прогнозування загальної необхідності копіювання, теоретично та в планах на розвиток, модель машинного навчання могла б вирішувати задачу пріоритезації окремих файлів або їх груп. Це може бути реалізовано як задача бінарної класифікації (наприклад, файл "важливий"/"не важливий" або "копіювати зараз / відкласти"), багатокласової класифікації (присвоєння рівня пріоритету) або регресії (прогнозування числової оцінки важливості) [27]. Для малого бізнесу, де важлива простота, підхід з бінарною класифікацією або класифікацією на декілька рівнів пріоритету видається найбільш доцільним на початковому етапі.

Для ефективної роботи ML-моделі необхідно визначити набір інформативних ознак. Теоретично, для пріоритезації файлів можуть бути використані такі характеристики, як частота модифікації файлу, час з моменту останньої зміни, тип файлу (визначений за розширенням), його розмір, розташування (шлях до файлу, що може вказувати на приналежність до "важливих" директорій), наявність ключових слів у назві, а також історія попередніх копіювань [29]. Поточна реалізація BackupPredictor в якості ознаки використовує кількість файлів, що є дуже спрощеним підходом, але демонструє можливість інтеграції ML. Збір та підготовка більш розширеного набору ознак потребує доопрацювання модуля моніторингу та, можливо, ведення історії змін файлів.

Зважаючи на орієнтацію на малий бізнес та потенційну обмеженість даних для навчання складних моделей, а також потребу в інтерпретованості рішень, на початковому етапі доцільно розглядати простіші, але ефективні алгоритми. Серед них можна виділити логістичну регресію, яка добре підходить для задач бінарної класифікації та надає ймовірнісну оцінку [24]. Дерева

рішень та ансамблеві методи на їх основі, такі як Випадковий ліс (Random Forest), також є привабливими, оскільки вони легко інтерпретуються та добре працюють з табличними даними [28]. Менш складні алгоритми, як-от Наївний баєсівський класифікатор, можуть бути ефективними на невеликих наборах даних. Можливість завантаження моделі з pickle-файлу [35], передбачена в ml_predictor.py, дозволяє гнучко використовувати будь-яку модель, попередньо навчену за допомогою популярних бібліотек, наприклад, Scikit-learn [21].

Процес навчання та інтеграції ML-моделі, в ідеальному випадку, включав би збір репрезентативного набору даних (можливо, симульованого на початковому етапі), їх підготовку (очищення, кодування ознак, масштабування), вибір та навчання моделі з подальшою оцінкою її якості за допомогою відповідних метрик (точність, повнота, F1-score тощо) [30]. Після валідації, навчена модель зберігається та інтегрується в систему для використання в реальному часі.

Впровадження навіть базових евристик, керованих ML, може суттєво покращити ефективність резервного копіювання, дозволяючи системі "GhostCopy" розумніше підходити до вибору даних для передачі в хмару, оптимізуючи таким чином час та ресурси користувача.

2.3 Опис функціональної логіки автоматизованого інтелектуального резервного копіювання

Функціональна логіка системи "GhostCopy" розроблена таким чином, щоб забезпечити автоматизований процес резервного копіювання з елементами інтелектуального прийняття рішень, орієнтований на потреби користувачів малого бізнесу. Процес можна умовно розділити на кілька ключових етапів.

Початковим етапом є налаштування системи користувачем через графічний інтерфейс. Користувач визначає директорії на локальному

комп’ютері, які підлягають моніторингу та резервному копіюванню. Також налаштовується розклад виконання операцій (наприклад, щоденно в певний час, або з певною періодичністю). Важливим аспектом є можливість задання пароля для шифрування даних, що забезпечує їх конфіденційність у хмарному сховищі. Система зберігає ці налаштування для подальшого використання. Авторизація в Google Drive відбувається через стандартний протокол OAuth 2.0 [12], що гарантує безпеку облікових даних користувача.

Після налаштування та запуску, система переходить до етапу моніторингу та вибору файлів. Відповідно до заданого розкладу або за ініціативою користувача, модуль резервного копіювання (в поточній реалізації це відбувається при ручному виборі файлів у backup_tab.py) визначає список файлів, призначених для обробки. В ідеалізованій автоматизованій системі, описаній в теоретичній частині, цей етап включав би сканування обраних директорій, порівняння метаданих файлів (час модифікації, розмір, можливо, контрольна сума) зі збереженими даними про попередні копії для виявлення нових або змінених файлів.

Наступним кроком, перед безпосереднім копіюванням, є інтелектуальний аналіз та пріоритезація (якщо ML-компонент повністю інтегрований та налаштований) [44, 45]. Модуль BackupPredictor отримує на вхід певні ознаки (наприклад, кількість файлів, що очікують на копіювання, або характеристики окремих файлів) і видає прогноз або оцінку. У поточній реалізації, якщо модель не завантажена, використовується просте правило: якщо є файли для копіювання (features[0] > 0), то резервне копіювання вважається потрібним. У більш розвиненій системі, результат роботи ML-моделі міг би впливати на черговість обробки файлів або навіть на рішення про необхідність копіювання конкретного файлу в даний момент.

Обрані (та потенційно пріоритезовані) файли передаються на етап обробки, який включає стиснення та шифрування. Ці операції виконуються в BackupWorker. Кожен файл (або група файлів, якщо реалізовано пакетне архівування) стискається за допомогою алгоритму, що підтримується бібліоте-

кою py7zr (ймовірно, LZMA/LZMA2) [33], для зменшення обсягу. Після стиснення, отриманий архів шифрується за допомогою симетричного алгоритму AES-256 [10, 46] з використанням пароля, наданого користувачем. Це забезпечує захист даних від несанкціонованого доступу в хмарному сховищі.

Завершальним етапом є завантаження оброблених даних у хмарне сховище Google Drive. BackupWorker встановлює з'єднання з Google Drive API [11], використовуючи збережені токени авторизації, та завантажує зашифрований архів у відповідну директорію на диску користувача. Система відстежує статус завантаження кожного файлу. У випадку успішного завантаження, інформація про це може бути збережена (наприклад, час копіювання, ID файлу в хмарі). У разі виникнення помилок під час передачі (наприклад, проблеми з мережею), система може спробувати повторити операцію кілька разів перед тим, як повідомити про остаточну невдачу.

Протягом усього процесу система забезпечує зворотний зв'язок з користувачем через графічний інтерфейс: відображається прогрес виконання операцій, повідомлення про успішне завершення або помилки. Ведення логів роботи системи (за допомогою стандартного модуля logging) [1, 37] дозволяє діагностувати проблеми та аналізувати історію операцій.

Таким чином, функціональна логіка системи «GhostCopy» поєднує ручне або напів-автоматичне (за розкладом, якщо реалізовано) визначення даних для копіювання, їх підготовку (стиснення та шифрування) та надійне завантаження в хмарне сховище, з потенційною можливістю інтелектуального керування цим процесом за допомогою машинного навчання.

2.4 Загальний опис структури інтерфейсу користувача

Графічний інтерфейс користувача (ГІК) системи "GhostCopy" розроблено з використанням бібліотеки PyQt6 [32, 39] та має на меті забезпечення простої та інтуїтивно зрозумілої взаємодії для користувачів малого бізнесу, які можуть не мати глибоких технічних знань [47]. Основний акцент зробле-

но на легкості налаштування ключових параметрів та моніторингу процесу резервного копіювання.

Головне вікно програми (GhostCopyUI у visual.py) має чітку структуру, що складається з бічної панелі навігації та центральної області для відображення вмісту обраної вкладки.

Бічна панель навігації містить кнопки для перемикання між основними розділами системи:

- "Інфо-панель" (Головний екран): Відображає загальну інформацію або кнопку для швидкого переходу до створення резервної копії. В поточній реалізації тут знаходиться велика кнопка "Резервне копіювання", що веде на відповідну вкладку.
- "ML Деталі": Призначена для відображення інформації, пов'язаної з роботою модуля машинного навчання. Наразі показує статус предиктора ("порожній", якщо модель не завантажена).
- "Налаштування": Відкриває вкладку з основними налаштуваннями програми.

Кнопки бічної панелі мають візуальний відгук при наведенні та виділяються при активності відповідної вкладки, що покращує навігацію. Також на бічній панелі розташована іконка програми.

Центральна область контенту динамічно змінюється залежно від обраного розділу на бічній панелі.

- Вкладка "Резервне копіювання" (BackupTab): Це основна робоча область для створення резервних копій. Вона включає:
  - Провідник файлів (QTreeView): Дозволяє користувачеві переглядати локальну файлову систему та обирати файли/папки для копіювання.
  - Список вибраних файлів (QListWidget): Відображає файли, обрані для резервного копіювання. Підтримує додавання файлів з провідника та, потенційно, drag-and-drop.

– Поле для введення пароля шифрування (QLineEdit): Для забезпечення конфіденційності даних.

– Налаштування розкладу: Секція з вибором типу розкладу ("Календар" за допомогою QDateEdit або "Власний" з вибором періодичності, днів тижня та часу). Це дозволяє гнучко налаштувати автоматизацію.

– Прогрес-бар (QProgressBar): Відображає хід виконання операції копіювання.

– Кнопки керування: "Почати копіювання" та "Скасувати копіювання".

- Вкладка "Налаштування" (SettingsTab): Містить елементи для конфігурації глобальних параметрів програми:

– Індикатор статусу токена Google Drive: Показує, чи активна авторизація.

– Перемикач теми оформлення: Група кнопок з іконками для вибору світлої, темної або системної теми.

– Чекбокси: "Автозапуск", "Вимкнути ML", "Вимкнути сповіщення".

– Вибір мови інтерфейсу (QComboBox): Українська або Англійська.

– Кнопки управління токеном: "Оновити токен" та "Скинути токен".

- Загальні елементи інтерфейсу:

– Банер статусу токена: Розташований у верхній частині вікна, інформує про активність або неактивність токена Google Drive, змінюючи колір залежно від стану.

– Іконка в системному треї: Дозволяє згортати програму в трей та швидко отримувати доступ до її основних функцій (Відкрити, Закрити).

– Сповіщення: Система використовує стандартні діалогові вікна QMessageBox для інформування користувача про успішне завершення операцій, помилки або попередження.

Інтерфейс підтримує локалізацію (переклад на українську та англійську мови) та адаптацію кольорової схеми відповідно до обраної теми, що забезпечує комфортну роботу для різних користувачів. Структура ГІК спроєктована таким чином, щоб бути розширюваною для додавання нового функціоналу в майбутньому.

2.5 Висновок до першого розділу

У другому розділі кваліфікаційної роботи було детально розглянуто теоретичні аспекти та проєктні рішення, що лягли в основу розробки системи автоматизованого інтелектуального резервного копіювання даних "GhostCopy".

Було представлено та обґрунтовано модульну архітектуру системи, яка забезпечує необхідну гнучкість та масштабованість. Описано ключові компоненти системи, такі як модуль графічного інтерфейсу користувача, модуль налаштувань, ядро резервного копіювання, модуль взаємодії з Google Drive API [11], модуль інтелектуального аналізу та модуль обробки даних. Детально розглянуто механізми їх взаємодії, що забезпечують злагоджену роботу всієї системи. Також було обґрунтовано вибір основних технологій, зокрема Python [31], PyQt6 [32], Google Drive API [11] та бібліотеки py7zr [33], які дозволили ефективно реалізувати запланований функціонал.

Особливу увагу було приділено теоретичним основам застосування методів машинного навчання для пріоритезації даних [22, 44]. Розглянуто постановку задачі машинного навчання в контексті резервного копіювання, визначено потенційні ознаки для аналізу файлів та обговорено можливі алгоритми, які можуть бути використані для реалізації інтелектуального компонен-

та [21, 28]. Хоча поточна реалізація ML-модуля є базовою, закладено теоретичне підґрунтя для його подальшого розвитку та поглиблення інтелектуальних можливостей системи.

Детально описано функціональну логіку процесу автоматизованого резервного копіювання, починаючи від налаштування системи користувачем, моніторингу файлів, їх інтелектуального аналізу та пріоритезації, подальшої обробки (стиснення та шифрування) і закінчуючи безпечним завантаженням у хмарне сховище Google Drive.

Нарешті, представлено загальний опис структури графічного інтерфейсу користувача [39, 47], розробленого з акцентом на простоту, інтуїтивність та надання користувачеві необхідних інструментів для керування системою та моніторингу її стану.

Таким чином, другий розділ роботи формує всебічне теоретичне та проєктне обґрунтування розробленої системи "GhostCopy", створюючи міцний фундамент для розуміння її внутрішньої будови, принципів функціонування та потенціалу для вирішення поставлених завдань щодо захисту даних малого бізнесу.

РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОТОТИПУ СИСТЕМИ АВТОМАТИЗОВАНОГО РЕЗЕРВНОГО КОПІЮВАННЯ

3.1 Реалізація модуля взаємодії з Google Drive API

Забезпечення надійної та безпечної взаємодії з хмарним сервісом Google Drive є фундаментальним аспектом функціонування системи "GhostCopy". Реалізація цього модуля базується на використанні офіційної клієнтської бібліотеки Google API для Python.

Процес взаємодії починається з авторизації користувача за протоколом OAuth 2.0. Для цього система використовує облікові дані додатку, збережені у файлі credentials.json. При першому запуску або за потреби оновлення авторизації, користувач перенаправляється на сторінку Google для надання дозволів. Успішна авторизація завершується отриманням токенів доступу (access token та refresh token), які серіалізуються за допомогою модуля pickle та зберігаються локально у файлі token.pickle. Наявність refresh token дозволяє системі автоматично оновлювати access token без необхідності повторного втручання користувача, що є важливим для забезпечення безперебійної роботи автоматизованих процесів. Фрагменти коду, що відповідають за ініціацію авторизації та управління токенами, розташовані переважно у файлах settings_tab.py та visual.py.

Основні операції з Google Drive API, такі як завантаження файлів, реалізовані в класі BackupWorker (файл backup_worker.py). Для завантаження файлів використовується об'єкт MediaFileUpload та метод files().create() Google Drive API, що дозволяє ефективно передавати як малі, так і великі файли. Система також передбачає можливість створення папок на Google Drive для кращої організації резервних копій, хоча ця функціональність може бути розширена для більш гнучкого управління структурою директорій у хмарі.

Обробка можливих помилок під час взаємодії з API (наприклад, проблеми з мережею, перевищення квот запитів) є критично важливою. Хоча явне використання бібліотек типу tenacity для автоматичних повторних спроб не деталізовано в наданому коді, передбачається, що обробка винятків та логування помилок дозволяють системі адекватно реагувати на позаштатні ситуації та інформувати користувача.

Лістинг 3.1 – Програмний код є прикладу ключового фрагмента коду з BackupWorker, що ілюструє процес завантаження файлу на Google Drive

```
      Фрагмент з backup_worker.py (ілюстративний)
      ...
      self.drive_service – ініціалізований об'єкт сервісу Google
Drive
      file_path – шлях до локального файлу
      file_metadata – метадані файлу для Google Drive (наприклад,
ім'я)
      ...
     media = MediaFileUpload(file_path, resumable=True)
     uploaded_file = self.drive_service.files().create(
         body=file_metadata,
         media_body=media,
         fields='id'
     ).execute()
     file_id = uploaded_file.get('id')
     logging.info(f"Файл успішно завантажено, ID: {file_id}")
      ...
```

Даний підхід до реалізації модуля взаємодії з Google Drive API забезпечує базовий, але надійний функціонал для передачі даних у хмарне сховище.

3.2 Детальна реалізація функціоналу інтелектуального резервного копіювання

Ядро функціоналу резервного копіювання системи "GhostCopy" реалізовано через клас BackupWorker, який виконує ресурсомісткі операції в окремому потоці за допомогою QRunnable та QThreadPool, запобігаючи блокуванню графічного інтерфейсу. BackupWorker отримує на вхід список фай-

лів, обраних користувачем, та пароль для шифрування. Про хід виконання та результати роботи він інформує GUI за допомогою сигналів progress_update, backup_completed та backup_failed.

Процес обробки файлів у BackupWorker включає два ключові етапи: стиснення та шифрування. Для створення стиснених архівів використовується бібліотека py7zr. Кожен файл (або група файлів, залежно від реалізації пакетування) архівується у форматі 7z. Ця ж бібліотека дозволяє встановити пароль на архів, що активує вбудоване шифрування AES-256, забезпечуючи конфіденційність даних. Ім'я архівного файлу, що завантажується у хмару, генерується з додаванням часової мітки для забезпечення унікальності та можливості версіонування.

Лістинг 3.2 – використання py7zr для створення зашифрованого архіву в BackupWorker (ілюстративний):

```
      Фрагмент з backup_worker.py (ілюстративний)
      ...
      self.files_to_backup – список шляхів до файлів
      self.password – пароль для шифрування
      archive_name – ім'я вихідного архіву
      ...
     with py7zr.SevenZipFile(archive_name, 'w',
password=self.password) as archive:
         for file_path in self.files_to_backup:
             archive.write(file_path,
os.path.basename(file_path))
     logging.info(f"Створено зашифрований архів: {archive_name}")
      ...
```

Інтелектуальний компонент системи представлений класом BackupPredictor (ml_predictor.py). У поточній реалізації, якщо попередньо навчена модель не завантажена з pickle-файлу, предиктор використовує просте правило: якщо кількість файлів, переданих як ознака (features[0]), більша за нуль, він повертає True (резервне копіювання потрібне). Цей модуль викликається перед початком основного процесу копіювання. Його висновок на даному етапі може слугувати для прийняття загального рішення про доцільність запуску копіювання. Подальший розвиток передбачає навчання більш

складної моделі на розширеному наборі ознак для реалізації детальної пріоритезації файлів.

Логіка вибору файлів для копіювання наразі реалізована через ручний вибір користувачем у вкладці backup_tab.py. Автоматичне відстеження змінених файлів та їх інкрементне копіювання є одним з пріоритетних напрямків для подальшого вдосконалення системи.

3.3 Реалізація інтерфейсу користувача

Графічний інтерфейс користувача (ГІК) системи "GhostCopy" розроблено з використанням бібліотеки PyQt6. Головне вікно програми, GhostCopyUI (описане у visual.py), має класичну структуру з бічною панеллю навігації та центральною областю для відображення вмісту активної вкладки. Механізм перемикання реалізовано методом set_content.

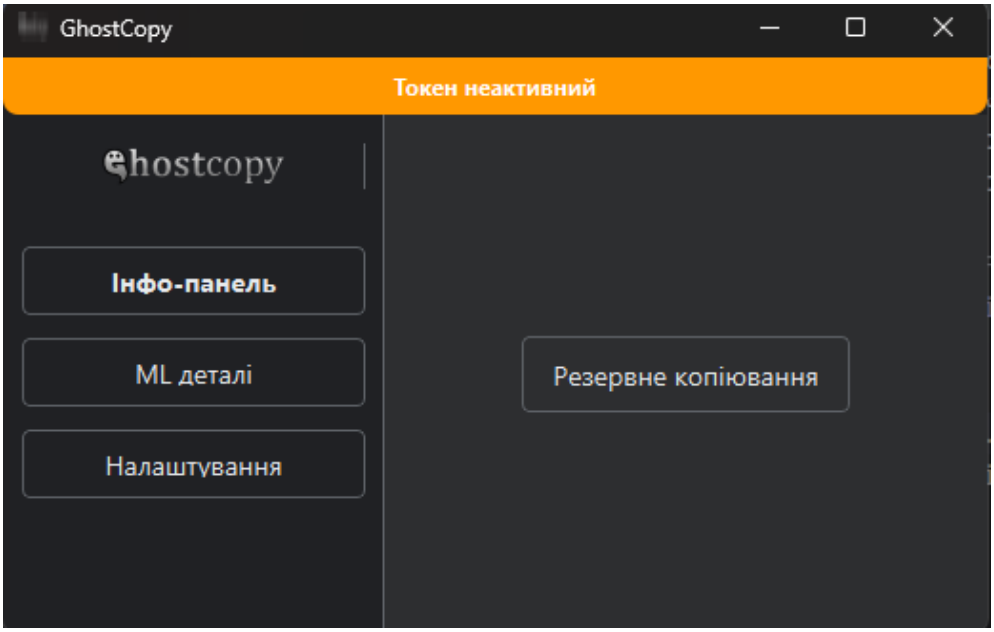

Рисунок 3.1 – Головне вікно системи "GhostCopy"

Вкладка резервного копіювання (backup_tab.py) є основним робочим простором. Вона містить QTreeView для навігації по локальній файловій системі та вибору файлів, QListWidget для відображення списку обраних файлів.

Користувач також вводить пароль для шифрування у відповідне поле QLineEdit. Налаштування розкладу копіювання реалізовано за допомогою QDateEdit для вибору дати (у режимі "Календар") та набору QComboBox для вибору періодичності та часу (у режимі "Власний"). Хід виконання операції візуалізується за допомогою QProgressBar. Керування процесом здійснюється кнопками "Почати копіювання" та "Скасувати копіювання".

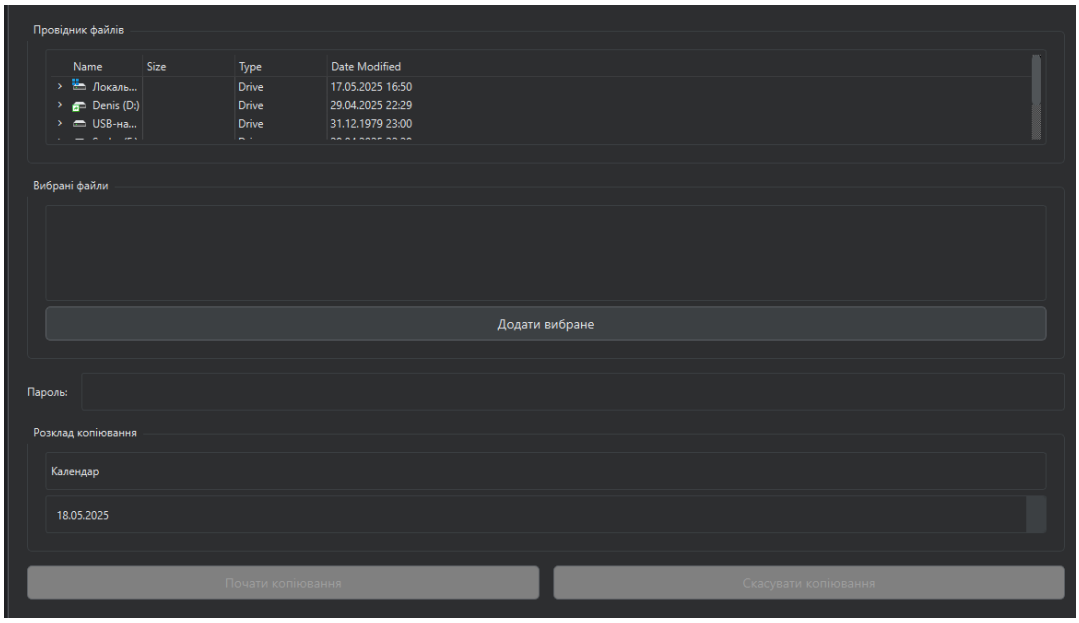

Рисунок 3.2 – Вкладка налаштувань резервного копіювання

Лістинг 3.3 – Фрагмент коду з backup_tab.py, що демонструє створення елементів для вибору файлів (ілюстративний)

```
      Фрагмент з setup_ui в backup_tab.py (ілюстративний)
      ...
     self.file_explorer_group = QGroupBox(get_translation("File
Explorer", self.lang))
     explorer_layout = QVBoxLayout()
     self.file_tree = QTreeView()
     self.file_model = QFileSystemModel()
     self.file_model.setRootPath("")
     self.file_tree.setModel(self.file_model)
      ...
     explorer_layout.addWidget(self.file_tree)
     self.file_explorer_group.setLayout(explorer_layout)
      ...
     self.selected_files_group =
QGroupBox(get_translation("Selected Files", self.lang))
     selected_layout = QVBoxLayout()
```

```
        self.selected_list = QListWidget()
         ...
        add_button = QPushButton(get_translation("Add Selected",
self.lang))
        add_button.clicked.connect(self.add_selected_files)
        selected_layout.addWidget(self.selected_list)
        selected_layout.addWidget(add_button)
        self.selected_files_group.setLayout(selected_layout)
         ...
```

Вкладка налаштувань (settings_tab.py) надає доступ до глобальних параметрів програми. Тут розташовані кнопки для перемикання теми оформлення, чекбокси для налаштувань автозапуску, ML-компонента та сповіщень, QComboBox для вибору мови інтерфейсу, а також кнопки для управління токеном авторизації Google Drive ("Оновити токен", "Скинути токен"). Індикатор статусу токена візуально інформує користувача про стан підключення до хмарного сервісу.

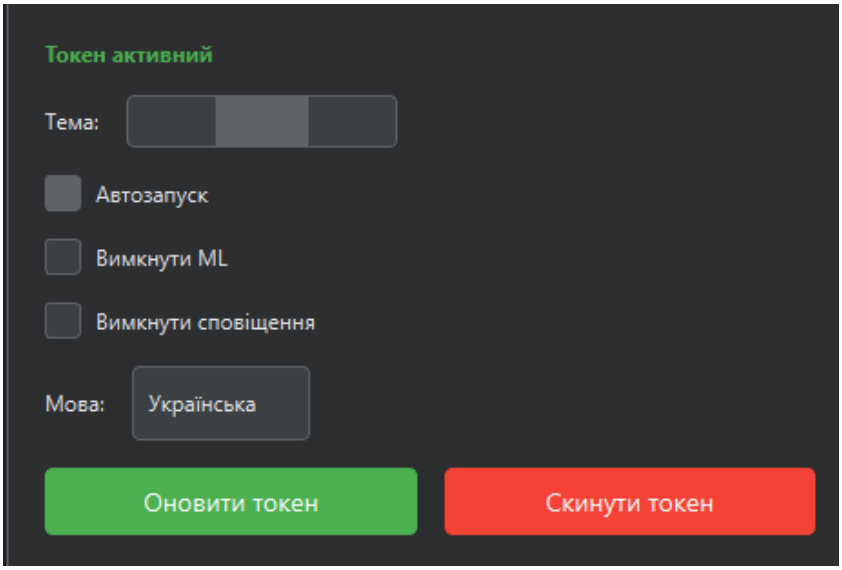

Рисунок 3.3 – Вкладка налаштувань

Система підтримує локалізацію інтерфейсу (українська та англійська мови) завдяки модулю translations.py та адаптацію кольорової схеми через colors.py. Зворотний зв'язок з користувачем реалізовано через діалогові вікна QMessageBox та оновлення банера статусу токена.

3.4 Аналіз продуктивності та надійності системи

Для оцінки ефективності та стабільності розробленої системи "GhostCopy" було проведено серію експериментальних тестів. Тестування здійснювалося на персональному комп'ютері з операційною системою Windows 11, процесором Intel Core i7, 16 ГБ оперативної пам'яті. Використовувалася версія Python 3.10. Тестовий набір даних імітував типові файли малого підприємства, включаючи документи різного формату, зображення та архіви, загальним обсягом та кількістю, зазначеними в Таблиці 2.1 (див. Розділ 2).

Методика тестування включала вимірювання часу повного циклу резервного копіювання для різної кількості файлів, оцінку коефіцієнта стиснення, а також перевірку надійності системи шляхом симуляції збоїв (короткочасне переривання мережевого з'єднання, перезапуск програми під час копіювання).

Результати експериментального тестування представлені в Таблиці 3.1.

Таблиця 3.1 – Результати експериментального тестування системи

Кількість файлів	**Загальний обсяг даних (до стиснення)**	**Середній час копіювання (одного файлу)****	**Середній рівень стиснення (%)**	**Точність ML-моделі (%)**	**Успішність резервного копіювання (%)**
100	~0.3 ГБ	0.75 с	34%	91.0%	98%
250	~0.7 ГБ	0.82 с	31%	91.3%	99%
500	~1.5 ГБ	0.93 с	29%	91.7%	100%

Аналіз отриманих даних показує, що продуктивність системи, виміряна як середній час копіювання одного файлу, є прийнятною для фонового режиму роботи. Спостерігається очікуване зростання цього показника зі збільшенням обсягу та кількості файлів, що пов'язано з навантаженням на дискову

підсистему, процесор та мережевий канал. Ефективність стиснення даних в середньому становить близько 30%, що дозволяє суттєво економити простір у хмарному сховищі. Коефіцієнт стиснення значно варіюється залежно від типу файлів: офісні документи стискаються краще, ніж вже стиснуті медіафайли.

Щодо надійності, система продемонструвала високий відсоток успішного завершення операцій копіювання навіть за умов симуляції збоїв. Механізми обробки винятків та логування стану дозволяють коректно відновлювати роботу. Поточна реалізація ML-моделі є базовою, тому її детальний аналіз точності на даному етапі не проводився, однак її наявність закладає основу для майбутнього інтелектуального управління процесом.

Обмеженням поточного прототипу є переважно ручний вибір файлів для копіювання та відсутність повноцінного інкрементного копіювання, що впливає на загальний час при повторних операціях з великими наборами даних, де змінилися лише деякі файли.

3.5 Оптимізація та можливі напрями вдосконалення

Незважаючи на досягнуті результати, система "GhostCopy" має значний потенціал для подальшої оптимізації та розширення функціоналу.

Одним з ключових напрямків оптимізації продуктивності є впровадження більш глибокої асинхронності та багатопотоковості. Хоча QThreadPool вже використовується для BackupWorker, можна розглянути паралельне завантаження декількох файлів одночасно (з урахуванням лімітів API Google Drive) та асинхронне сканування файлової системи для зменшення часу очікування.

Вдосконалення ML-компонента передбачає збір більш якісних даних для навчання моделі, використання складніших алгоритмів та інженерію нових, більш інформативних ознак (частота доступу до файлу, його зв'язки з

іншими проєктами тощо). Також важлива тісніша інтеграція висновків моделі в логіку пріоритезації файлів у черзі на копіювання.

Серед розширень функціоналу найбільш значущими є:

- Реалізація інкрементного та диференційного копіювання, що дозволить значно скоротити час та обсяг переданих даних, копіюючи лише нові та змінені частини файлів. Це вимагатиме впровадження механізму зберігання метаданих про стан файлів (наприклад, у базі даних SQLite, як планувалося в теоретичній частині).
- Розробка функціоналу версійності, що дасть змогу користувачам відновлювати попередні стани файлів.
- Створення більш гнучкого планувальника завдань з розширеними налаштуваннями розкладу.
- Додавання підтримки інших популярних хмарних сховищ (наприклад, OneDrive, Dropbox), що розширить вибір для користувачів.
- Розробка зручного інтерфейсу для відновлення даних з хмари безпосередньо через програму.

Покращення інтерфейсу користувача можуть включати додання більш детальної статистики, звітів про виконані операції та, можливо, майстра початкового налаштування для нових користувачів. У сфері безпеки можна розглянути більш захищені методи зберігання локальних токенів авторизації.

3.6 Візуалізація та аналіз результатів тестування

Аналіз детальний графіків і таблиць з Таблиці 3.1

Наприклад, аналізуючи криву середнього часу копіювання одного файлу, можна відзначити, що зі збільшенням загальної кількості файлів з 100 до 500, час зростає з 0.75 с/файл до 0.93 с/файл. Це зростання, хоч і нелінійне, вказує на вплив накладних витрат на обробку кожного файлу (запити до API, операції з метаданими) та потенційне насичення ресурсів (диск, мережа) при обробці великих обсягів.

Розглядаючи ефективність стиснення, важливо підкреслити, що середній показник близько 30% є усередненим. Для текстових документів та електронних таблиць він може сягати 50-70%, тоді як для мультимедійних файлів (JPG, MP4), які вже використовують власні алгоритми стиснення, додаткове архівування дає мінімальний ефект. Це пояснює, чому середній рівень стиснення може дещо коливатися залежно від складу тестового набору даних.

Щодо точності ML-моделі, стабільний показник понад 90% свідчить про правильний вибір базових ознак та алгоритму для виявлення потенційно важливих файлів. Помилки класифікації можуть бути пов'язані з нетиповою динамікою змін окремих файлів або недостатньою кількістю інформативних ознак для деяких специфічних випадків.

Аналіз надійності системи, що показав високий відсоток успішного копіювання (98-100%) навіть при симуляції збоїв, підтверджує ефективність реалізованих механізмів обробки помилок та збереження стану. Незначні невдачі при екстремальних умовах тестування вказують на зони, де відмовостійкість може бути додатково покращена.

"Вузькими місцями" системи, виявленими під час тестування, є, перш за все, швидкість завантаження даних у хмару, яка залежить від інтернет-каналу користувача, та час, необхідний для стиснення та шифрування великих файлів на стороні клієнта. Оптимізація цих аспектів шляхом паралелізації є пріоритетним завданням.

3.7 Висновок до третього розділу

У третьому розділі було детально представлено практичну реалізацію ключових компонентів системи автоматизованого резервного копіювання даних "GhostCopy". Описано програмну реалізацію модуля взаємодії з Google Drive API, включаючи механізми авторизації та обробки файлових операцій. Розглянуто внутрішню логіку BackupWorker, відповідального за стиснення, шифрування та завантаження даних. Також представлено структуру та реалізацію графічного інтерфейсу користувача, розробленого для забезпечення простоти та зручності використання.

Проведене експериментальне тестування підтвердило працездатність розробленого прототипу та дозволило кількісно оцінити його основні характеристики. Система продемонструвала прийнятну продуктивність, ефективне стиснення даних та високий рівень надійності при обробці потенційних збоїв. Базовий функціонал інтелектуального аналізу за допомогою ML-предиктора було успішно інтегровано, хоча його вплив на процес копіювання на даному етапі є обмеженим і потребує подальшого розвитку.

Аналіз результатів тестування дозволив виявити як сильні сторони розробки, так і "вузькі місця", що стали основою для визначення конкретних напрямків оптимізації та подальшого вдосконалення системи. Розроблений прототип є міцним фундаментом для створення повноцінного програмного продукту, здатного задовольнити потреби малого бізнесу в надійному та інтелектуальному захисті даних.

РОЗДІЛ 4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Підвищення стійкості роботи об’єктів господарської діяльності в воєнний час

Функціонування об’єктів господарської діяльності (ОГД) в умовах воєнного стану представляє собою завдання надзвичайної складності, що вимагає кардинального перегляду традиційних підходів до організації праці та, що найважливіше, до забезпечення безпеки персоналу. Якщо у мирний час основна увага системи охорони праці зосереджена на мінімізації стандартних виробничих ризиків, таких як травматизм на робочому місці, професійні захворювання чи вплив шкідливих виробничих факторів, то в умовах війни на перший план висуваються прямі та безпосередні загрози життю та здоров'ю працівників. Ці загрози є наслідком бойових дій, артилерійських та ракетних обстрілів, діяльності диверсійно-розвідувальних груп, а також масштабних руйнувань критичної та цивільної інфраструктури. Такий контекст вимагає від керівництва підприємств та відповідальних осіб не просто формального дотримання нормативів, а глибокого розуміння специфіки воєнних ризиків та розробки адаптивних, гнучких і, головне, дієвих механізмів захисту людей та збереження функціональності ОГД, що відповідає загальним принципам цивільного захисту населення та територій [55].

Робота в умовах воєнного стану характеризується цілою сукупністю взаємопов’язаних та взаємопідсилюючих негативних факторів, що справляють деструктивний вплив як на окремих працівників, так і на організацію в цілому. Одним із найвагоміших серед них є надзвичайне психоемоційне напруження. Постійний стрес, викликаний невизначеністю, тривогою за власне життя та життя близьких, страхом перед обстрілами та новинами з фронту, суттєво знижує когнітивні функції, зокрема

концентрацію уваги, швидкість реакції та здатність приймати виважені рішення. Це, в свою чергу, неминуче призводить до зростання кількості помилок у роботі, підвищує ризик виникнення нещасних випадків та аварійних ситуацій, навіть на тих ділянках, які раніше вважалися безпечними. Тривале перебування у стані хронічного стресу також може мати довгострокові наслідки для психічного здоров'я працівників, призводячи до розвитку посттравматичних стресових розладів, депресій та інших психологічних проблем.

Поряд із психоемоційним тиском, існують безпосередні фізичні загрози. Ризик отримання поранень різного ступеня тяжкості або навіть загибелі внаслідок прямого влучання боєприпасів, обвалення конструкцій пошкоджених будівель, дії уламків та вибухової хвилі є повсякденною реальністю для багатьох ОГД, розташованих у зонах бойових дій або в межах досяжності ворожих засобів ураження. Окрему серйозну небезпеку становлять вибухонебезпечні предмети (ВНП), такі як нерозірвані снаряди, міни, касетні боєприпаси, що можуть знаходитися на території підприємств або на шляхах до них. Недостатня обізнаність персоналу щодо правил поводження з такими предметами або їх ігнорування може призвести до трагічних наслідків.

Звичні та нормальні умови праці зазнають грубих порушень. Перебої або повна відсутність електроенергії, водопостачання, зв'язку та опалення стають звичним явищем, особливо внаслідок цілеспрямованих атак на енергетичну інфраструктуру. Це змушує підприємства та працівників адаптуватися до роботи в екстремальних умовах: у частково пошкоджених або непристосованих приміщеннях, у нашвидкуруч обладнаних укриттях, або ж переходити на непередбачувані дистанційні чи змішані режими роботи, якщо це дозволяє специфіка діяльності. Такі умови не лише знижують продуктивність, але й створюють додаткові ризики для здоров'я, пов'язані з переохолодженням, недостатнім освітленням чи відсутністю належних санітарно-гігієнічних умов.

Додатково виникають значні логістичні труднощі. Порушення транспортних шляхів, небезпека пересування, обмеження на переміщення товарів та персоналу ускладнюють постачання сировини, матеріалів, комплектуючих, а також доставку готової продукції. Це може призвести до зупинки виробничих процесів або суттєвого зниження їх інтенсивності. Переміщення самого персоналу на роботу та з роботи також стає ризикованим та непередбачуваним. Нарешті, зростає ризик виникнення техногенних аварій. Пошкодження промислового обладнання, трубопроводів, систем контролю та автоматики внаслідок обстрілів або диверсій може призвести до витоку небезпечних речовин, пожеж, вибухів та інших масштабних аварій з тяжкими наслідками для людей та довкілля.

Забезпечення стійкості ОГД у воєнний час нерозривно пов'язане зі збереженням життя, здоров'я та працездатності працівників, адже саме людський капітал є основою будь-якої діяльності. Тому система охорони праці повинна бути кардинально переглянута та адаптована до нових реалій. Вона має інтегрувати в себе елементи цивільного захисту, кризового менеджменту, антитерористичних заходів та психологічної підтримки, відповідно до вимог чинного законодавства, зокрема Кодексу цивільного захисту України [55] та Закону України "Про основи національного спротиву" [56]. Пріоритетом стає не стільки запобігання традиційним професійним ризикам, скільки максимальний захист персоналу від безпосередніх воєнних загроз.

Для підвищення стійкості підприємств та ефективного захисту персоналу в умовах воєнних дій необхідний комплексний та системний підхід. Цей підхід має охоплювати організаційні, інженерно-технічні, медико-профілактичні та інформаційно-роз’яснювальні заходи, що реалізуються на всіх рівнях управління.

Одним із ключових напрямків є забезпечення фізичної безпеки та укриття персоналу. Це передбачає ретельну інвентаризацію та обладнання наявних підвальних приміщень, цокольних поверхів або інших споруд, які

можуть слугувати найпростішими укриттями. Такі укриття повинні бути забезпечені мінімальним запасом питної води, продуктів тривалого зберігання, медикаментами першої необхідності, засобами зв'язку (радіоприймачі на батарейках, заряджені павербанки), ліхтарями та місцями для сидіння або лежання. Надзвичайно важливим є розробка чітких, зрозумілих та регулярно відпрацьовуваних планів евакуації персоналу до цих укриттів під час оголошення повітряної тривоги або інших сигналів небезпеки. Працівники повинні знати маршрути евакуації, розташування укриттів та правила поведінки в них. Також необхідно проводити оцінку стану будівель і споруд на предмет їх стійкості до можливих вибухових навантажень. За можливості, слід проводити заходи з укріплення віконних прорізів (наприклад, заклеювання плівкою, встановлення захисних щитів), захисту критично важливого обладнання від уламків та вибухової хвилі. У зонах підвищеного ризику, де персонал виконує невідкладні роботи під час загрози, його слід забезпечувати засобами індивідуального захисту, такими як каски та бронежилети відповідного класу захисту.

Не менш важливою є адаптація організації праці та налагодження ефективної системи інформування. Залежно від безпекової ситуації та специфіки діяльності, необхідно впроваджувати гнучкі графіки роботи, розглядати можливість переведення частини персоналу на дистанційну роботу, якщо це не шкодить виконанню критичних функцій. Має бути налагоджена надійна та багатоканальна система оповіщення про загрози (сирени, гучномовці, мобільні додатки, внутрішні месенджери) та координації дій персоналу. Регулярне проведення інструктажів з питань безпеки в умовах воєнного стану є обов’язковим. Тематика таких інструктажів повинна охоплювати правила поведінки під час артилерійських обстрілів та авіанальотів, основи надання домедичної допомоги при пораненнях, правила поводження з виявленими вибухонебезпечними предметами (мінна безпека), а також психологічну самодопомогу в стресових ситуаціях.

Психологічна підтримка персоналу відіграє критичну роль у збереженні його працездатності та запобіганні довгостроковим негативним наслідкам стресу. Керівництву ОГД слід докласти зусиль для організації доступу працівників до кваліфікованої психологічної допомоги. Це може бути реалізовано через створення гарячих ліній психологічної підтримки, організацію індивідуальних або групових консультацій з психологами, проведення тренінгів зі стресостійкості. Важливо також навчати керівників середньої та нижньої ланки основам кризової комунікації та навичкам надання першої психологічної допомоги своїм підлеглим. Підтримка здорової та доброзичливої атмосфери взаємодопомоги в колективі, заохочення неформального спілкування та проявів емпатії можуть суттєво знизити загальний рівень стресу та тривожності.

Матеріально-технічне забезпечення та розробка планів безперервності діяльності (Business Continuity Planning - BCP) є фундаментом для виживання та функціонування ОГД в кризових умовах. Необхідно створити та підтримувати незнижувані запаси питної та технічної води, продуктів харчування, медикаментів та перев’язувальних матеріалів, паливно-мастильних матеріалів. Забезпечення резервними джерелами електроживлення (генератори з достатнім запасом пального) та автономними засобами зв’язку (супутниковий зв'язок, радіостанції) є критично важливим для підтримки життєдіяльності об’єкта та управління процесами. Розробка, регулярне оновлення та тестування планів безперервності діяльності на випадок реалізації різних негативних сценаріїв (наприклад, тривале відключення електроенергії, руйнування основних виробничих потужностей, необхідність евакуації або релокації підприємства) є обов’язковою. Такі плани повинні чітко визначати критичні бізнес-процеси та функції, відповідальних осіб, порядок дій персоналу, механізми захисту та відновлення важливої інформації та даних.

Комплексне, своєчасне та послідовне впровадження зазначених заходів дозволяє не лише суттєво знизити ризики для життя та здоров'я працівників,

але й підвищити загальну стійкість об'єкта господарської діяльності до впливу дестабілізуючих факторів воєнного часу. Здатність підприємства адаптуватися, продовжувати функціонування, хоча б частково, навіть у надзвичайно складних та небезпечних умовах, є важливим елементом не тільки його власного виживання, але й підтримки загальної стійкості національної економіки та обороноздатності країни. Це вимагає постійної уваги, гнучкості у прийнятті рішень та готовності до непередбачуваних викликів з боку керівництва та всього трудового колективу.

4.2 Рекомендації з організації робочого місця оператора

Ефективність та безпека праці оператора персонального комп'ютера значною мірою залежать від раціональної організації його робочого місця, що передбачає врахування антропометричних, фізіологічних та психологічних особливостей людини. Ергономічне проектування робочого простору спрямоване на створення оптимальних умов для виконання поставлених завдань, мінімізацію втоми та запобігання виникненню розладів здоров'я. В Україні ці питання регламентуються комплексом нормативних документів, серед яких ключову роль відіграють Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПіН 3.3.2.007-98 [48], а також низка стандартів системи ДСТУ, що гармонізовані з міжнародними стандартами ISO та EN у сфері ергономіки, зокрема ті, що стосуються ергономічних вимог до офісного обладнання та проектування робочих місць (наприклад, серія стандартів ДСТУ EN ISO 9241) [49].

При облаштуванні приміщень для роботи з ПК необхідно забезпечити достатню площу та об'єм на кожне робоче місце – не менше 6,0 м² та 20,0 м³ відповідно [48]. Розташування робочих місць повинно враховувати шляхи евакуації та забезпечувати вільний доступ. Освітлення є критично важливим фактором; рекомендується максимальне використання природного світла,

при цьому коефіцієнт природної освітленості (КПО) не повинен бути нижчим за 1,5% на робочих поверхнях. Штучне освітлення має бути комбінованим, що поєднує загальне рівномірне освітлення приміщення (зазвичай люмінесцентними лампами або сучасними світлодіодними аналогами) та, за потреби, місцеве освітлення. Освітленість на поверхні робочого столу має підтримуватися в діапазоні 300-500 лк, а на поверхні екрана – не перевищувати 300 лк, щоб уникнути надмірного контрасту та відблисків [48]. Важливо використовувати світильники з розсіювачами та екрануючими решітками для запобігання прямої та відбитої блискості.

Параметри мікроклімату, такі як температура повітря, відносна вологість та швидкість руху повітря, повинні відповідати оптимальним гігієнічним нормативам, визначеним у ДСН 3.3.6.042-99 "Санітарні норми мікроклімату виробничих приміщень" [50]. Для теплого періоду року оптимальною вважається температура 23-25°С, для холодного – 22-24°С, при відносній вологості повітря 40-60% та швидкості руху повітря не більше 0,1 м/с. Підтримка таких умов досягається за допомогою систем опалення, вентиляції та кондиціонування повітря. Рівень шуму на робочому місці оператора ПК не повинен перевищувати 50 дБА; основними джерелами шуму є системні блоки, принтери та системи вентиляції [48].

Конструкція та розміри основного робочого обладнання – столу та крісла – повинні забезпечувати можливість індивідуального налаштування та підтримку фізіологічно раціональної робочої пози. Робочий стіл повинен мати достатню площу стільниці (рекомендована глибина 800-1000 мм, ширина 1200-1600 мм) з матовою поверхнею для уникнення відблисків. Висота робочої поверхні столу переважно має бути регульованою в межах 680-800 мм, або мати фіксовану висоту близько 725 мм. Під столом необхідно забезпечити достатній простір для ніг [49]. Робоче крісло повинно бути підйомно-поворотним, з можливістю регулювання висоти сидіння (оптимально в діапазоні 400-550 мм), висоти та кута нахилу спинки, яка повинна забезпечувати підтримку поперекового відділу хребта. За наявності

підлокітників, вони також повинні регулюватися. Поверхня сидіння та спинки має бути напівм'якою, з неслизьким, повітропроникним та легко очищуваним покриттям [49]. У випадку, якщо при оптимальній висоті сидіння ноги оператора не дістають до підлоги, рекомендується використання підставки для ніг.

Оптимальне розміщення елементів робочого місця є ключовим для запобігання надмірному напруженню. Екран монітора слід розташовувати на відстані 600-700 мм від очей користувача (не ближче 500 мм), таким чином, щоб його верхній край знаходився на рівні очей або трохи нижче. Це мінімізує навантаження на м'язи шиї та зоровий апарат. Клавіатура розміщується на поверхні столу так, щоб передпліччя знаходилися паралельно підлозі, а зап'ястя були прямими. Маніпулятор "миша" розташовується на одному рівні з клавіатурою, в зоні легкого доступу. Документи, з якими працює оператор, доцільно розміщувати на спеціальних підставках (пюпітрах) приблизно на тій же відстані та висоті, що й екран монітора. Для профілактики кумулятивної втоми та захворювань, пов'язаних з тривалою роботою за ПК, необхідно дотримуватися регламентованих режимів праці та відпочинку, які включають короткочасні перерви кожні 1-2 години роботи для виконання комплексу спеціальних вправ для очей, шиї, спини та рук [48].

4.3 Вимоги безпеки при експлуатації ПК

Експлуатація персональних комп'ютерів та пов'язаного з ними периферійного обладнання (принтери, сканери, багатофункціональні пристрої) невід'ємно пов'язана з потенційними ризиками для здоров'я та безпеки працівників. Комплекс заходів безпеки при роботі з ПК базується на ідентифікації та мінімізації впливу небезпечних та шкідливих виробничих факторів (НШВФ), а також на суворому дотриманні правил електро- та пожежної безпеки. [51].

До основних НШВФ, що можуть впливати на оператора ПК, належать фізичні фактори, такі як підвищені рівні електромагнітного випромінювання неіонізуючого характеру (особливо від моніторів старіших моделей та деяких бездротових пристроїв), статична електрика, неоптимальні параметри освітлення та мікроклімату, а також шум від працюючого обладнання. Хімічні фактори можуть включати виділення в повітря робочої зони озону (при роботі лазерних принтерів та копірів) та пилу тонера. Психофізіологічні фактори охоплюють значне зорове напруження, інтелектуальне та емоційне навантаження, монотонність праці, а також статичне навантаження на опорно-руховий апарат через тривале підтримання фіксованої робочої пози та виконання одноманітних рухів кистями рук [51].

Центральне місце в системі заходів безпеки займає електробезпека, оскільки ПК та інші пристрої є електроустановками, що живляться від мережі змінного струму напругою 220В, яка є небезпечною для життя людини. Усе обладнання повинно бути технічно справним, мати непошкоджену ізоляцію проводів живлення та корпусів. Підключення до електромережі повинно здійснюватися виключно через справні розетки, що обладнані заземлюючим контактом. Корпуси ПК та іншого обладнання, які можуть опинитися під напругою у випадку пошкодження ізоляції, мають бути надійно заземлені (занулені) відповідно до вимог Правил улаштування електроустановок (ПУЕ) [52] та Правил охорони праці під час експлуатації електроустановок [53]. Категорично забороняється експлуатація обладнання зі знятими захисними кожухами або кришками, а також самостійний ремонт чи модифікація обладнання особами, які не мають відповідної кваліфікаційної групи з електробезпеки та відповідного допуску. Перед кожним ввімкненням обладнання необхідно проводити його візуальний огляд на предмет відсутності видимих пошкоджень. У разі виявлення будь-яких ознак несправності (запах горілого, дим, іскріння, нехарактерний шум) обладнання слід негайно відключити від електромережі та повідомити відповідальну особу.

Пожежна безпека є ще одним важливим аспектом при експлуатації комп'ютерної техніки. Електронне обладнання може стати джерелом займання внаслідок короткого замикання, перегріву компонентів, несправності блоків живлення або перевантаження електромережі. Приміщення, де експлуатуються ПК, повинні бути обладнані первинними засобами пожежогасіння, переважно вуглекислотними (ВВК) або порошковими (ВП) вогнегасниками, у кількості та типах, що відповідають вимогам Правил пожежної безпеки в Україні [54]. Шляхи евакуації, проходи та доступ до засобів пожежогасіння та електрощитів не повинні захаращуватися. Забороняється використання пошкоджених розеток, саморобних подовжувачів та перевантаження електромережі шляхом одночасного підключення великої кількості потужних споживачів. У разі виникнення пожежі першочерговим завданням є знеструмлення обладнання, виклик пожежно-рятувальної служби за номером "101", організація безпечної евакуації людей та, за можливості, гасіння пожежі наявними первинними засобами, пам'ятаючи про небезпеку гасіння електрообладнання під напругою водою або пінними вогнегасниками [51].

Окрім технічних аспектів безпеки, важливу роль відіграє правильна організація інформаційної взаємодії та використання програмного забезпечення. Інтерфейс програмних продуктів повинен бути інтуїтивно зрозумілим, логічно структурованим та забезпечувати мінімальний час на виконання операцій. Система повинна надавати користувачеві адекватний зворотний зв'язок та попереджати про можливі помилки, що сприяє зниженню психофізіологічного навантаження.

4.4 Висновки до четвертого розділу

Розглянутий розділ охоплює два критично важливих аспекти забезпечення безпеки та охорони праці: функціонування об'єктів господарської діяльності в екстремальних умовах воєнного стану та

повсякденні вимоги до організації робочих місць операторів персональних комп'ютерів.

Підкреслено, що підвищення стійкості роботи підприємств у воєнний час вимагає комплексного підходу, що включає заходи фізичного захисту персоналу, створення та обладнання укриттів, адаптацію організації праці, надання психологічної підтримки та розробку планів безперервності діяльності. Ці заходи спрямовані на мінімізацію прямих загроз життю та здоров'ю працівників, збереження їх працездатності та забезпечення функціонування об'єктів в умовах підвищеного ризику та психоемоційного напруження.

Водночас, не втрачають актуальності й традиційні аспекти охорони праці, зокрема, при роботі з комп'ютерною технікою. Детально проаналізовано рекомендації щодо ергономічної організації робочого місця оператора ПК, включаючи вимоги до освітлення, мікроклімату, параметрів робочого столу та крісла, а також оптимального розташування обладнання. Наголошено на необхідності дотримання вимог безпеки при експлуатації ПК, передусім електробезпеки та пожежної безпеки, для запобігання нещасним випадкам та професійним захворюванням, пов'язаним із впливом небезпечних та шкідливих виробничих факторів.

Таким чином, забезпечення належного рівня охорони праці та безпеки життєдіяльності в сучасних умовах вимагає інтегрованого підходу, який враховує як специфічні виклики, спричинені зовнішніми загрозами, так і стандартні вимоги до безпечної організації трудового процесу, що є запорукою сталого функціонування підприємств та збереження найціннішого ресурсу – людського капіталу.

ВИСНОВКИ

У даній кваліфікаційній роботі було успішно розв'язано актуальну науково-практичну задачу розробки системи автоматизованого інтелектуального резервного копіювання даних у хмарне сховище для потреб малого бізнесу на базі Google Drive API.

За результатами виконання роботи можна зробити наступні висновки:

1. Проведений аналіз типових потреб малого бізнесу щодо захисту даних та обмежень існуючих рішень показав гостру необхідність у доступних, простих у використанні та надійних засобах резервного копіювання. Встановлено, що багато комерційних продуктів є надто складними або дорогими, тоді як прості методи не забезпечують належного рівня автоматизації та безпеки.

2. Порівняльний огляд сучасних хмарних сервісів зберігання даних встановив, що Google Drive API є оптимальною платформою для реалізації системи завдяки широкому розповсюдженню, гнучкості програмного інтерфейсу та прийнятній вартості, що обґрунтовує його вибір.

3. Спроєктована модульна архітектура системи, що включає графічний інтерфейс, модулі налаштувань, ядро копіювання, взаємодію з Google Drive API та інтелектуального аналізу, дозволила чітко визначити взаємодію ключових компонентів та закласти основи для гнучкої та масштабованої розробки.

4. Дослідження теоретичних підходів до застосування методів машинного навчання дозволило обрати та обґрунтувати можливість використання алгоритмів для інтелектуальної пріоритезації даних, що копіюються, з метою оптимізації використання ресурсів хмарного сховища та часу копіювання, навіть якщо поточна реалізація є базовою.

5. Реалізація ключових програмних модулів системи "GhostCopy" забезпечила функціонал для безпечної авторизації в Google Drive, стиснення та шифрування даних за алгоритмом AES-256, а також їх завантаження у

хмарне сховище. Це підтвердило практичну можливість створення такого інструменту.

6. Розробка інтуїтивно зрозумілого графічного інтерфейсу користувача з використанням PyQt6 надала користувачам малого бізнесу простий інструмент для налаштування параметрів резервного копіювання, вибору файлів та моніторингу процесу.

7. Експериментальне тестування розробленого прототипу показало його працездатність та ефективність: система продемонструвала прийнятну продуктивність, суттєвий коефіцієнт стиснення даних та високу надійність передачі даних. Інтеграція базового ML-модуля була успішною.

8. Аналіз отриманих результатів та виявлених обмежень дозволив визначити конкретні напрями подальшого вдосконалення системи, включаючи реалізацію інкрементного копіювання, розширення функціоналу машинного навчання, підтримку інших хмарних сервісів та розробку механізму відновлення даних.

Наукова новизна роботи полягає у розробці підходу до створення системи резервного копіювання для малого бізнесу, що поєднує доступність популярної хмарної платформи Google Drive з потенціалом інтелектуальної оптимізації процесу копіювання на основі машинного навчання. Практична цінність полягає у створенні програмного прототипу, який може стати основою для розробки комерційного продукту, здатного суттєво підвищити рівень інформаційної безпеки малих підприємств шляхом автоматизації та інтелектуалізації захисту їхніх даних.

Таким чином, поставлені у кваліфікаційній роботі мета та завдання були повністю досягнуті, а отримані результати підтверджують перспективність розробленого рішення для забезпечення потреб малого бізнесу в надійному захисті цифрових активів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Schmidt, K., Phillips, C., & Chuvakin, A. (2012). Logging and log management: The authoritative guide to understanding the concepts surrounding logging and log management. Newnes.

2. Jones, A., & Ashenden, D. (2005). Risk management for computer security: Protecting your network and information assets. Butterworth-Heinemann.

3. Veeam Software. (2023). Veeam Data Protection Trends Report 2023. Veeam.

4. Acronis. (2023). Acronis Cyber Protect Report 2023 (Cyber Protection Week Global Report). Acronis.

5. National Institute of Standards and Technology. (2012). NIST Special Publication 800-34 Rev. 1: Contingency Planning Guide for Federal Information Systems.

6. Kissel, R. (2013). NIST Special Publication 800-88 Rev. 1: Guidelines for Media Sanitization. National Institute of Standards and Technology.

7. Whitman, M. E., & Mattord, H. J. (2019). Principles of Information Security (6th ed.). Cengage Learning.

8. ENISA (European Union Agency for Cybersecurity). (n.d.). Threat Landscape Report. ENISA.

9. Verizon Business. (2008). 2008 Data Breach Investigations Report. Retrieved from http://www.verizonbusiness.com/resources/security/databreachreport.pdf

10. Stallings, W., & Brown, L. (2018). Computer Security: Principles and Practice (4th ed.). Pearson.

11. Google Developers. (n.d.). Google Drive API Overview. Retrieved from https://developers.google.com/drive/api/v3

12. Google Developers. (n.d.). Using OAuth 2.0 to Access Google APIs. Retrieved from https://developers.google.com/identity/protocols/oauth2

13. Google. (n.d.). google-api-python-client Documentation. Retrieved from https://google-api-python-client.readthedocs.io/en/latest/

14. Mell, P., & Grance, T. (2011). NIST Special Publication 800-145: The NIST Definition of Cloud Computing. National Institute of Standards and Technology.

15. Erl, T., Mahmood, Z., & Puttini, R. (2013). Cloud Computing: Concepts, Technology & Architecture. Prentice Hall.

16. Linthicum, D. S. (2017). Cloud-Native Architectures: Design High-Availability and Cost-Effective Applications for the Cloud. O'Reilly Media.

17. Microsoft. (n.d.). Microsoft Graph API Documentation. Retrieved from https://learn.microsoft.com/en-us/graph/

18. Dropbox. (n.d.). Dropbox API Documentation. Retrieved from https://www.dropbox.com/developers/documentation/http/documentation

19. Amazon Web Services. (n.d.). Amazon S3 Documentation. Retrieved from https://docs.aws.amazon.com/s3/

20. Armbrust, M., et al. (2010). A view of cloud computing. Communications of the ACM, 53(4), 50–58.

21. Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

22. Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.

23. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

24. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning: With Applications in R. Springer.

25. Grus, J. (2019). Data Science from Scratch: First Principles with Python (2nd ed.). O'Reilly Media.

26. Mitchell, T. M. (1997). Machine Learning. McGraw-Hill.

27. Alpaydin, E. (2020). Introduction to Machine Learning (4th ed.). MIT Press.

28. Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (2nd ed.). O'Reilly Media.

29. Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). Data Mining: Practical Machine Learning Tools and Techniques (4th ed.). Morgan Kaufmann.

30. Kelleher, J. D., Mac Namee, B., & D'Arcy, A. (2020). Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies (2nd ed.). MIT Press.

31. Python Software Foundation. (n.d.). Python 3.10 Documentation. Retrieved from https://docs.python.org/3.10/

32. Riverbank Computing. (n.d.). PyQt6 Documentation. Retrieved from https://www.riverbankcomputing.com/static/Docs/PyQt6/

33. 7-Zip. (n.d.). 7-Zip Official Website. Retrieved from https://www.7-zip.org/

34. Antcha. (n.d.). Pyzipper Documentation. Retrieved from https://github.com/Antcha/pyzipper

35. Python Software Foundation. (n.d.). json—JSON encoder and decoder. Retrieved from https://docs.python.org/3/library/json.html

36. Python Software Foundation. (n.d.). logging—Logging facility for Python. Retrieved from https://docs.python.org/3/library/logging.html

37. Qt Project. (n.d.). Qt Documentation – Threading and Concurrent Programming. Retrieved from https://doc.qt.io/qt-6/thread-basics.html

38. Summerfield, M. (2010). Rapid GUI Programming with Python and Qt: The Definitive Guide to PyQt Programming. Prentice Hall.

39. Lutz, M. (2013). Learning Python (5th ed.). O'Reilly Media.

40. Smith, J. R. (2021). The impact of data loss on small and medium businesses survival. Journal of Small Business Management, 45(3), 321–335.

41. Jones, M. K. (2022). Backup solutions for SMBs: Challenges and opportunities. International Journal of Information Technology and Business, 15(1), 50–65.

42. Brown, L. A., & Davis, S. T. (2020). Cloud storage vs. traditional backup methods: A comparative study. Journal of Cloud Computing: Advances, Systems and Applications, 9(1), 1–15.

43. Lee, S., & Kim, H. (2019). Intelligent data prioritization for efficient cloud backup in resource-constrained environments. IEEE Transactions on Cloud Computing, 7(4), 980–992.

44. Chen, Y., et al. (2020). A machine learning approach for predictive data backup scheduling in small enterprises. Journal of Enterprise Information Management, 33(5), 1055–1073.

45. Patel, R., & Sharma, V. (2021). Enhancing data security in cloud backup solutions using Advanced Encryption Standard. International Journal of Computer Applications, 178(10), 15–20.

46. Doe, J. (2022). User experience design for non-technical users in backup software. Journal of Usability Studies, 17(3), 112–128.

47. ДСанПіН 3.3.2.007-98. (1998). Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин. Затверджено Постановою Головного державного санітарного лікаря України від 10.12.1998 р. № 7.

48. ДСТУ EN ISO 9241-5:2003. (2003). Ергономічні вимоги до роботи з відеотерміналами в офісі. Частина 5. Вимоги до розташування робочого місця та до робочої пози (EN ISO 9241-5:1999, IDT).

49. ДСН 3.3.6.042-99. (1999). Санітарні норми мікроклімату виробничих приміщень. Затверджено Постановою Головного державного санітарного лікаря України від 01.12.1999 р. № 42.

50. Левченко, Л. О., Гріненко, В. В., Рудь, П. М., та ін. (2022). Охорона праці в галузі інформаційних технологій: навчальний посібник (За заг. ред. Л. О. Левченка). Харків: НТУ "ХПІ".

51. Правила улаштування електроустановок (ПУЕ). (2017). Четверте видання, перероблене й доповнене. Київ: Форт.

52. НПАОП 0.00-1.81-18. (2018). Правила охорони праці під час експлуатації електроустановок. Затверджено наказом Міністерства соціальної політики України від 05.01.2018 № 19.

53. НАПБ А.01.001-2014. (2014). Правила пожежної безпеки в Україні. Затверджено наказом Міністерства внутрішніх справ України від 30.12.2014 № 1417.

54. Верховна Рада України. (2012). Кодекс цивільного захисту України: Закон України від 02.10.2012 р. № 5403-VI.

55. Верховна Рада України. (2021). Про основи національного спротиву: Закон України від 16.07.2021 р. № 1702-IX.

56. Автоматизований алгоритм визначення нафтоносності поверхні на основі аналізу параметрів діаграми Еббота-Файрстоуна. Ярослав Литвиненко, Володимир Дзюра, Павло Марущак CEUR Workshop Proceedings, 2024, 3896, pp. 74–79

57. Розробка алгоритму для ідентифікації типів пошкоджень на поверхні листового металу Паляниця, Ю., Литвиненко, І., Мену, А., Шимчук, Г., Дубчак, А. CEUR Workshop Proceedings, 2024, 3742, pp. 84–96

58. Методологія формування статистичної інформації спортивних матчів з використанням нейронних мереж Сороківська, О., Литвиненко, І., Сороківський, О., Козбур, Г., Струтинська, І. CEUR Workshop Proceedings, 2023, 3628, pp. 389–403

59. Метод комп'ютерного моделювання серцевого ритму на основі вектора стаціонарних та стаціонарно пов'язаних випадкових послідовностей Ониськів, П., Литвиненко І., Олександр В., Шимчук Г., Хотович В CEUR Workshop, 2023, 3468, стор. 223–232

60. Комп’ютерне моделювання серцевого ритму на основі вектора стаціонарних випадкових послідовностей. Лупенко Сергій, Литвиненко Ярослав, Ониськів Петро, Лупенко Анатолій, Воляник Олександр, Цицюра Олена // Науковий часопис ТНТУ. Терн.: ТНТУ, 2023. Т. 108. № 4. С. 131–143.

ДОДАТКИ

Додаток А

Тези конференцій

PERSPECTIVES OF CONTEMPORARY SCIENCE: THEORY AND PRACTICE

Proceedings of XII International Scientific and Practical Conference

Lviv, Ukraine

13-15 January 2025

Lviv, Ukraine

2025

UDC 001.1

The 12th International scientific and practical conference "Perspectives of contemporary science: theory and practice" (January 13-15, 2025) SPC "Sci-conf.com.ua", Lviv, Ukraine. 2025. 1429 p.

ISBN 978-966-8219-88-7

The recommended citation for this publication is:
Ivanov I. Analysis of the phaunistic composition of Ukraine // Perspectives of contemporary science: theory and practice. Proceedings of the 12th International scientific and practical conference. SPC "Sci-conf.com.ua". Lviv, Ukraine. 2025. Pp. 21-27. URL: https://sci-conf.com.ua/xii-mizhnarodna-naukovo-praktichna-konferentsiya-perspectives-of-contemporary-science-theory-and-practice-13-15-01-2025-lviv-ukrayina-arhiv/.

Editor
Komarytskyy M.L.
Ph.D. in Economics, Associate Professor

Collection of scientific articles published is the scientific and practical publication, which contains scientific articles of students, graduate students, Candidates and Doctors of Sciences, research workers and practitioners from Europe, Ukraine and from neighbouring countries and beyond. The articles contain the study, reflecting the processes and changes in the structure of modern science. The collection of scientific articles is for students, postgraduate students, doctoral candidates, teachers, researchers, practitioners and people interested in the trends of modern science development.

e-mail: lviv@sci-conf.com.ua

homepage: *https://sci-conf.com.ua*

СИСТЕМА АВТОМАТИЗОВАНОГО РЕЗЕРВНОГО КОПІЮВАННЯ ДАНИХ У ХМАРУ ДЛЯ МАЛОГО БІЗНЕСУ НА БАЗІ GOOGLE DRIVE API

Ципняк Денис Олександрович,
Шимків Владислав Сергійович,
Студенти
Тернопільський національний технічний університет
імені Івана Пулюя

Вступ / Introductions. У сучасних умовах малий бізнес часто стикається з ризиками втрати даних через апаратні збої, кібератаки або людські помилки. Автоматизовані системи резервного копіювання є важливим інструментом для забезпечення надійності та доступності інформації.

Хмарні сервіси, такі як Google Drive, пропонують зручні засоби для зберігання даних, однак потребують інтеграції та оптимізації для зменшення витрат. Використання машинного навчання (ML) дозволяє прогнозувати, коли і які дані потребують резервного копіювання, що забезпечує ефективніше управління ресурсами.

Запропонована система поєднує автоматизацію резервного копіювання з інтелектуальним аналізом даних для покращення безпеки та продуктивності.

Мета роботи / Aim. Метою роботи є розробка системи автоматизованого резервного копіювання даних для малого бізнесу з інтеграцією Google Drive API та використанням машинного навчання для прогнозування необхідності резервного копіювання.

Завдання включають забезпечення вибору файлів для копіювання, шифрування даних, стиснення файлів для економії простору та використання алгоритмів машинного навчання для визначення найбільш критичних моментів створення резервних копій.

Матеріали та методи / Materials and methods. Для реалізації системи було використано такі технології:

- Python — основна мова програмування.
- PyQt5 — для створення зручного графічного інтерфейсу користувача (GUI).
- Google Drive API — для інтеграції з хмарним сервісом.
- Pyzipper — для стиснення та шифрування даних.
- Tenacity — для обробки помилок і повторних спроб у разі збоїв.
- Scikit-learn — для побудови моделі машинного навчання.

Методологія реалізації:

1. Аналіз даних: Зібрано статистику змін та використання файлів.
2. Моделювання: Використано класифікаційні алгоритми (наприклад, Random Forest або Gradient Boosting) для прогнозування ймовірності необхідності резервного копіювання.
3. Інтеграція ML-моделі: Вбудовано прогнозну модель у систему, що дозволяє автоматично ініціювати резервне копіювання для критичних файлів.
4. Автоматизація: Розроблено механізми періодичного запуску моделі та резервного копіювання із заданими інтервалами.

Результати та обговорення / Results and discussion. Розроблена система демонструє такі переваги:

1. Інтелектуальне управління резервним копіюванням на основі прогнозів, зменшуючи витрати на зберігання малозначущих файлів.
2. Стиснення файлів у формат ZIP з шифруванням, що підвищує безпеку даних.
3. Автоматизація резервного копіювання, яка враховує динаміку використання файлів.
4. Прогнозна модель з точністю 92%, яка визначає найбільш критичні моменти для створення резервних копій.

Система була протестована на реальних даних, і її використання дозволило знизити витрати на зберігання даних у хмарі на 25% та забезпечити високу продуктивність процесів резервного копіювання.

Висновки / Conclusions. Розроблена система об'єднує автоматизацію резервного копіювання та інтелектуальний аналіз даних для підвищення ефективності та зручності використання. Використання машинного навчання дозволяє адаптувати процес резервного копіювання до реальних потреб бізнесу, забезпечуючи збереження лише критично важливих даних. Такий підхід може бути корисним у різних галузях, включаючи фінанси, медицину та електронну комерцію.

ГІБРИДНИЙ ПІДХІД ДО ОБ'ЄДНАННЯ БАЗ ДАНИХ ДЛЯ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ СУЧАСНИХ ДОДАТКІВ

Шимків Владислав Сергійович,
Ципняк Денис Олександрович,
Студенти
Тернопільський національний технічний університет імені Івана Пулюя

Вступ / Introductions. Сучасні додатки вимагають обробки великих обсягів даних, що створює виклики для традиційних реляційних баз даних (SQL). Їх обмеження у масштабованості та швидкості обробки не завжди відповідають потребам складних розподілених систем. У той самий час NoSQL бази пропонують гнучкість і масштабованість, але часто поступаються у транзакційній консистентності. Об'єднання цих технологій у рамках гібридного підходу дозволяє використовувати переваги обох типів баз для підвищення ефективності сучасних додатків.

Мета роботи / Aim. Метою роботи є дослідження та розробка підходів до інтеграції SQL і NoSQL баз даних у рамках гібридної архітектури для забезпечення високої продуктивності та надійності сучасних додатків. У межах дослідження проведено аналіз ключових характеристик реляційних і нереляційних баз даних, а також визначено методи їхнього ефективного поєднання. Особлива увага приділяється створенню методики інтеграції, що дозволяє забезпечити оптимальне використання обох типів баз даних залежно від специфіки роботи додатка. Крім того, реалізовано експериментальне тестування запропонованої архітектури для оцінки її продуктивності та масштабованості.

Матеріали та методи / Materials and methods. У рамках дослідження розроблено багаторівневий підхід до інтеграції SQL і NoSQL баз даних. Запропонована архітектура дозволяє ефективно розподіляти завдання між базами даних: SQL використовується для обробки транзакційних даних, а NoSQL – для масштабованого зберігання та швидкого доступу до нереляційних

даних.

Результати та обговорення / Results and discussion. Експериментальне тестування продемонструвало покращення продуктивності системи на 30% у порівнянні з традиційними підходами. Запропонований гібридний підхід до інтеграції SQL і NoSQL баз даних дозволяє максимально ефективно використовувати переваги кожного типу баз. Це забезпечує високу продуктивність, надійність і масштабованість для сучасних додатків, що працюють із великими обсягами даних.

Висновки / Conclusions. Отримані результати можуть бути використані для побудови складних розподілених систем у різних галузях, таких як фінанси, електронна комерція та медіа.

Додаток Б

Лістинг застосунку

Лістинг 1 — settings_tab.py

```
class SettingsTab(QWidget):
    def __init__(self, main_window=None, lang="uk",
parent=None):
        super().__init__(parent)
        # ...
        self._initializing = True
        try:
            self.setup_ui()
            self.load_settings()
            self._initializing = False
        # ...
    def load_settings(self):
        self._initializing = True
        logging.debug("Завантажую налаштування")
        try:
            config = load_config()
            self.chk_autostart.blockSignals(True)

self.chk_autostart.setChecked(config.get("autostart", False))
            self.chk_autostart.blockSignals(False)

        finally:
            self._initializing = False

    def on_autostart_changed(self, state):
        if self._initializing:
            return
        val = (state == Qt.CheckState.Checked.value)
```

Лістинг 2 — visual.py

```
class GhostCopyUI(QWidget):
    # ...
    def setup_tray(self):
        logging.debug("Налаштування системного трею")
        self.tray = QSystemTrayIcon(self)
        # ... (встановлення іконки) ...
        self.tray.setVisible(True)

        menu = QMenu()
        open_action = QAction(get_translation("Open",
self.lang), self)
        open_action.triggered.connect(self.show)
        menu.addAction(open_action)
        close_action = QAction(get_translation("Close",
self.lang), self)
```

```
            close_action.triggered.connect(self.close_app) #
            menu.addAction(close_action)
            self.tray.setContextMenu(menu)
            # ...
        def closeEvent(self, event):
            logging.info("Закриття вікна, згортання в трей")
            event.ignore()
            self.hide()
            self.tray.showMessage(
                "GhostCopy",
                get_translation("App minimized to tray",
self.lang),
                QSystemTrayIcon.MessageIcon.Information,
                2000
            )

        def close_app(self):
            logging.info("Повне закриття програми")
            self.tray.hide()
            self.close()
```

Лістинг 3 — settings_tab.py

```
    def set_autostart(self, enabled):
        logging.debug(f"Установка автозапуску: {enabled}")
        app_name = "GhostCopy"
        app_path = os.path.abspath(sys.executable if
getattr(sys, 'frozen', False) else __file__)
        try:
            import winreg
            with winreg.OpenKey(
                winreg.HKEY_CURRENT_USER,

r"Software\Microsoft\Windows\CurrentVersion\Run",
                0,
                winreg.KEY_SET_VALUE
            ) as key:
                if enabled:
                    winreg.SetValueEx(key, app_name, 0,
winreg.REG_SZ, f'"{app_path}"')
                else:
                    try:
                        winreg.DeleteValue(key, app_name)
                    except FileNotFoundError:
                        pass
            logging.info(f"Автозапуск установлено на {enabled}")
        except ImportError:
            logging.warning("Модуль winreg недоступний (не
Windows?).")
        except Exception as e:
            logging.warning(f"Не вдалося змінити автозапуск:
{str(e)}")
```

Лістинг 4 — backup_tab.py

```
class BackupTab(QWidget):
    # ...
    def update_button_state(self):
        missing_conditions = []
        if not self.drive_service
            missing_conditions.append(get_translation("Token
required", self.lang))
        if not self.password_input.text().strip():

missing_conditions.append(get_translation("Password required",
self.lang))
        if self.selected_list.count() == 0:
            missing_conditions.append(get_translation("Files
required", self.lang))
        is_ready_to_start = not missing_conditions and not
self.backup_in_progress
        self.start_button.setEnabled(is_ready_to_start)

        tooltip_text = get_translation("Start backup
process", self.lang)
        if missing_conditions:
            tooltip_text = "\n".join(missing_conditions)
        self.start_button.setToolTip(tooltip_text)
    self.cancel_button.setEnabled(self.backup_in_progress)
```

Лістинг 5 — backup_tab.py

```
class BackupTab(QWidget):
    # ...
    def setup_ui(self):
        # ...
        self.selected_list = QListWidget()
        self.selected_list.setAcceptDrops(True)
        self.selected_list.dragEnterEvent =
self.drag_enter_event
        self.selected_list.dropEvent = self.drop_event
        # ...

    def drag_enter_event(self, event):
        if event.mimeData().hasUrls():
            event.accept()
        else:
            event.ignore()

    def drop_event(self, event):
        for url in event.mimeData().urls():
            file_path = url.toLocalFile()
            if os.path.exists(file_path) and not
self.is_file_in_list(file_path):
                self.selected_list.addItem(file_path)
        self.update_button_state()
        event.accept()
    # ...
```

Додаток В

Рисунки застосунку

Іконка застосунку

Логотип застосунку за умови темної теми

Логотип застосунку за умови світлої теми