

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки  
Кафедра інформаційних систем та технологій**

«На правах рукопису»  
УДК 004.42

До захисту допущено:  
Завідувач кафедри  
\_\_\_\_\_ Олександр РОЛІК  
«\_\_» \_\_\_\_\_ 2024 р.

**Магістерська дисертація  
на здобуття ступеня магістра  
за освітньо-професійною програмою «Інтегровані інформаційні  
системи»  
зі спеціальності 126 «Інформаційні системи та технології»  
на тему: «Автоматизована система аналізу файлових сховищ для  
середовища Windows Server»**

Виконав:  
студент 2 курсу, групи ІС-33мп  
Піскун Вадим Сергійович \_\_\_\_\_

Керівник:  
доцент кафедри ІСТ, д.т.н., доцент,  
Цьопа Наталія Володимирівна \_\_\_\_\_

Рецензент:  
завідувач кафедри ІСТ ФІТ  
КНУ ім. Тараса Шевченка,  
д.т.н., професор,  
Дружинін Володимир Анатолійович \_\_\_\_\_

Засвідчую, що у цій магістерській  
дисертації немає запозичень з праць  
інших авторів без відповідних  
посилань.  
Студент \_\_\_\_\_

Київ – 2024 року

**Національний технічний університет України**  
**«Київський політехнічний інститут імені Ігоря Сікорського»**  
**Факультет інформатики та обчислювальної техніки**  
**Кафедра інформаційних систем та технологій**

Рівень вищої освіти – другий (магістерський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Олександр РОЛІК

« \_\_\_\_ » \_\_\_\_\_ 2024 р.

**ЗАВДАННЯ**  
**на магістерську дисертацію студенту**  
**Піскуну Вадиму Сергійовичу**

1. Тема дисертації «Автоматизована система аналізу файлових сховищ для середовища Windows Server», науковий керівник дисертації Цьопа Наталія Володимирівна, доцент кафедри ІСТ, кандидат технічних наук, доцент, затверджені наказом по університету від «08» 11 2024 р. № 5016-с
2. Термін подання студентом дисертації «09» 12 2024 р.
3. Об'єкт дослідження: процес аналізу файлових сховищ у середовищі Windows Server.
4. Вихідні дані: автоматизована система аналізу файлових сховищ для середовища Windows Server, яка повинна забезпечувати швидкий та детальний аналіз файлових систем, створення графіків, а також забезпечувати стабільну роботу серверів, мінімізуючи навантаження на їх ресурси та включати функціонал для оцінки використання ресурсів.
5. Перелік завдань, які потрібно розробити: провести аналіз предметної області та існуючих рішень, визначити вимоги до розроблюваної системи, розробити архітектурне рішення системи, огляд технології розробки, реалізувати серверну та клієнтську частини системи, розробити стартап-проект.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу діаграма сутностей (А3), діаграма варіантів використання (А3), діаграма діяльностей(А3), діаграма сутність-зв'язок(А3), діаграма послідовностей (А3), діаграма класів(А3), схема розгортання (А3), діаграма потоків даних(А3).

7. Дата видачі завдання 02.09.2024 р.

#### Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1.	Аналіз предметної області	01.09.2024 – 15.09.2024	
2.	Огляд існуючих рішень	16.09.2024 – 30.09.2024	
3.	Формування вимог до системи	01.10.2024 – 5.10.2024	
4.	Виокремлення сутностей системи, акторів та створення Use-кейсів на їх основі	06.10.2024 – 10.10.2024	
5.	Огляд архітектурних рішень	11.10.2024 – 13.10.2024	
6.	Створення власного архітектурного рішення	14.10.2024 – 20.10.2024	
7.	Програмна реалізація системи	21.10.2024 – 20.11.2024	
8.	Тестування системи	21.11.2024 – 30.11.2024	
9.	Оформлення документації	01.12.2024 – 08.12.2024	
10.	Подання дисертації на захист	09.12.2024	

Студент

Вадим ПІСКУН

Науковий керівник

Наталія ЦЬОПА

## РЕФЕРАТ

Піскун В. С. Автоматизована система аналізу файлових сховищ для середовища Windows Server. КПІ ім. І. Сікорського, Київ, 2024.

Дисертація містить: 116 с., 33 табл., 16 рис., 10 дод., 30 джерел.

Актуальність теми дослідження обумовлена стрімким зростанням обсягів даних у сучасних інформаційних системах та необхідністю ефективного управління файловими сховищами в середовищі Windows Server. Оптимізація та аналіз файлових ресурсів є критично важливими для забезпечення продуктивності, безпеки та надійності корпоративних систем.

Метою магістерської дисертації є підвищення ефективності управління файловими сховищами в середовищі Windows Server, за рахунок розробки автоматизованої системи для аналізу дискових ресурсів.

Об'єкт дослідження: процес аналізу файлових сховищ у середовищі Windows Server.

Предмет дослідження: методи та засоби реалізації автоматизованого аналізу файлових сховищ, архітектурні рішення та технології, які забезпечать якісний моніторинг та управління дисковим простором.

Практична значимість роботи полягає у можливості використання розробленої системи в корпоративних мережах.

Ключові слова: АВТОМТИЗОВАНА СИСТЕМА АНАЛІЗУ, ФАЙЛОВЕ СХОВИЩЕ, Windows Server, .NET, ОПТИМІЗАЦІЯ РЕСУРСІВ, УПРАВЛІННЯ ДАНИМИ, БЕЗПЕКА.

## ABSTRACT

Piskun V. S. Automated file storage analysis system for Windows Server environmen. Igor Sikorsky Kyiv Polytechnic Institute, Kyiv, 2024.

The dissertation contains: 116 pages, 33 tables, 16 figures, 10 appendices, 30 references.

The relevance of the research topic is determined by the rapid growth of data volumes in modern information systems and the need for effective management of file storage in the Windows Server environment. Optimization and analysis of file resources are critically important for ensuring the productivity, security, and reliability of corporate systems.

The aim of the master's thesis is to improve the efficiency of file storage management in the Windows Server environment through the development of an automated system for analyzing disk resources.

Research object: the process of analyzing file storage in the Windows Server environment.

Research subject: methods and means of implementing automated analysis of file storage, architectural solutions, and technologies that will ensure high-quality monitoring and management of disk space.

The practical significance of the work lies in the possibility of using the developed system in corporate networks.

Keywords: AUTOMATED ANALYSIS SYSTEM, FILE STORAGE, Windows Server, .NET, RESOURCE OPTIMIZATION, DATA MANAGEMENT, SECURITY.

## ЗМІСТ

ПЕРЕЛІК УМНОВНИХ СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ .....	12
ВСТУП.....	13
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ .....	16
1.1 Опис предметного середовища та процесу діяльності .....	18
1.2 Аналіз існуючих рішень та результатів дослідження. ....	20
1.2.1 SolarWinds Storage Resource Monitor .....	20
1.2.2 TreeSize Professional .....	22
1.2.3 ManageEngine FileAudit Plus .....	24
1.3 Постановка задачі .....	26
1.3.1 Призначення та цілі розробки .....	26
1.3.2 Інформаційне забезпечення .....	27
Висновки до розділу 1 .....	28
2 ФОРМУВАННЯ СИСТЕМНИХ ВИМОГ .....	30
2.1 Функціональні вимоги .....	30
2.2 Нефункціональні вимоги .....	32
2.3 Основні сутності системи .....	35
Висновки до розділу 2 .....	38
3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ.....	39
3.1 Розрахунок вартості зайнятого дискового простору.....	39
3.2 Розподіл витрат між відділами .....	40
3.3 Планування бюджетування.....	41
3.4 Аналіз ефективності використання дискового простору.....	42
3.5 Додаткові методи та формули .....	43

	9
3.5.1 Розрахунок середнього зростання даних.....	43
3.5.2 Розрахунок часу до повного заповнення дискового простору.....	44
3.6 Впровадження формул у систему .....	44
Висновок до розділу 3 .....	45
4 ПРОЄКТУВАННЯ СИСТЕМИ .....	47
4.1 Сценарії використання системи .....	47
4.1.1 Сценарій 1: Збір даних про файлові сховища.....	48
4.1.2 Сценарій 2: Перегляд звітів про використання дискового простору .....	49
4.1.3 Сценарій 3: Планування бюджету на зберігання даних .....	49
4.1.4 Сценарій 4: Отримання сповіщень про аномалії у використанні дискового простору.....	50
4.1.5 Сценарій 5: Запит інформації про використання дискового простору .....	51
4.2 Моделювання та проектування системи .....	52
4.2.1 Архітектура .....	53
4.2.2 Серверна частина.....	53
4.2.3 Клієнтська частина .....	55
4.3 Вибір технологій.....	56
4.3.1 Мова програмування .....	56
4.3.2 Фреймворки.....	57
4.3.3 Середовище розробки .....	57
5 ПРОЄКТУВАННЯ БАЗИ ДАНИХ .....	60
5.1 Структура бази даних.....	60
5.2 Опис таблиць бази даних .....	61
5.2.1 Таблиця Servers (Сервери).....	61
5.2.2 Таблиця Folders (Папки) .....	62

	10
5.2.3 Таблиця Departments (Відділи).....	63
5.2.4 Таблиця Users (Користувачі) .....	63
5.2.5 Таблиця DiskUsages (Використання дисків).....	64
5.2.6 Таблиця Notifications (Сповіщення).....	65
Висновки до розділу 5 .....	65
<b>6 ІНТЕРФЕЙС КОРИСТУВАЧА ТА UX/UI ДИЗАЙН .....</b>	<b>67</b>
6.1 Принципи проектування інтерфейсу .....	67
6.1.1 Простота та мінімалізм .....	67
6.1.2 Інтуїтивна навігація.....	69
6.1.3 Візуалізація даних про дисковий простір.....	70
6.1.4 Аналіз використання папок .....	72
6.1.5 Фільтрація та сортування даних.....	74
6.1.6 Можливість персоналізації .....	75
6.2 Адаптивність та кросплатформенність.....	76
6.2.1 Адаптивний дизайн .....	76
6.2.2 Кросплатформенність.....	79
6.2.3 Тестування та оптимізація .....	80
Висновки до розділу 6 .....	80
<b>7 ТЕСТУВАННЯ ТА ОПТИМІЗАЦІЯ СИСТЕМИ.....</b>	<b>81</b>
7.1 Тестування функціональності та навантажувальне тестування.....	81
7.1.1 Тестування функціональності .....	81
7.1.2 Навантажувальне тестування .....	85
7.1.3 Оптимізація продуктивності.....	85
7.2 Тестування безпеки та надійності системи .....	86
7.2.1 Тестування безпеки .....	86

	11
7.2.2 Тестування надійності.....	87
7.2.3 Покращення безпеки та надійності.....	87
Висновки до розділу 7.....	88
8 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ .....	89
8.1 Опис ідеї проекту.....	89
8.2 Технологічний аудит ідеї проекту .....	91
8.3 Аналіз ринкових можливостей запуску стартап-проекту.....	93
8.4 Розроблення ринкової стратегії продукту.....	101
8.5 Розроблення маркетингової програми стартап-проекту.....	108
Висновки до розділу 8.....	112
ВИСНОВКИ.....	114
ПЕРЕЛІК ІНФОРМАЦІЙНИХ ДЖЕРЕЛ.....	117
ДОДАТОК А .....	116
ДОДАТОК Б.....	117
ДОДАТОК В .....	118
ДОДАТОК Г.....	119
ДОДАТОК Д.....	120
ДОДАТОК Е.....	121
ДОДАТОК Ж .....	122
ДОДАТОК И .....	123
ДОДАТОК К .....	124
ДОДАТОК Л .....	125
ДОДАТОК М .....	126

## ПЕРЕЛІК УМНОВНИХ СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

API – Інтерфейс програмування застосунків (Application Programming Interface): набір інструментів та правил, що дозволяють різним програмним системам взаємодіяти між собою.

Blazor – фреймворк для створення вебзастосунків на мові програмування C# та .NET.

GUI – Графічний інтерфейс користувача (Graphical User Interface): тип інтерфейсу, який використовує графічні елементи для взаємодії користувача з комп'ютером.

JSON – JavaScript Object Notation: легкий формат обміну даними, який використовується для передачі структурованої інформації між клієнтом та сервером.

PS – PowerShell: потужна оболонка командного рядка та мова сценаріїв від Microsoft для автоматизації адміністративних задач.

REST – Архітектурний стиль передачі стану представлення (Representational State Transfer): набір принципів для створення веб-сервісів, які є масштабованими та легко інтегруються.

SQL – Мова структурованих запитів (Structured Query Language): стандартна мова програмування для управління та маніпулювання реляційними базами даних.

UI – Інтерфейс користувача (User Interface): засоби взаємодії користувача з програмним забезпеченням, включаючи елементи управління та візуальні компоненти.

UX – Користувацький досвід (User Experience): загальне враження користувача від взаємодії з системою, включаючи зручність використання та задоволеність.

## ВСТУП

На сьогоднішній день інформаційні технології відіграють ключову роль у забезпеченні ефективності та конкурентоспроможності підприємств. Зростання обсягів даних, що зберігаються та обробляються, вимагає впровадження потужних та надійних рішень для управління цими ресурсами. Особливо важливим є управління файловими сховищами на серверних платформах, таких як Windows Server, де велика кількість даних організована у вигляді структурованих папок та дискових розділів.

Збереження, обробка та аналіз файлових даних є критичними завданнями для будь-якої організації. Від ефективного управління дисковим простором залежить не лише продуктивність системи, але й економічна ефективність підприємства. Неефективне використання ресурсів може призвести до зайвих витрат на зберігання даних, зниження швидкодії систем та ускладнення процесів обслуговування. Тому необхідність автоматизованого аналізу файлових сховищ стає все більш актуальною.

Автоматизація процесів збору, обробки та візуалізації даних про файлові сховища дозволяє значно підвищити ефективність управління ІТ-інфраструктурою. Впровадження таких систем дозволяє адміністраторам оперативно отримувати актуальну інформацію про стан дискових ресурсів, виявляти аномалії використання простору, а також планувати розширення або оптимізацію існуючих ресурсів. Це, в свою чергу, сприяє підвищенню загальної продуктивності підприємства та зниженню ризиків, пов'язаних з перевантаженням або нестачею дискового простору.

Сучасна інформаційна ера характеризується стрімким розвитком технологій зберігання даних, що зумовлює необхідність впровадження інноваційних рішень для управління ними. Автоматизовані системи аналізу файлових сховищ стають невід'ємною частиною корпоративних ІТ-інфраструктур, забезпечуючи не лише моніторинг стану ресурсів, але й аналітику їх використання на різних рівнях організації – від загального корпоративного до окремих відділів чи секторів.

Дана магістерська дисертація спрямована на розробку та впровадження комплексного рішення, яке забезпечує автоматизований збір, обробку та візуалізацію інформації про файлові сховища. Система дозволяє ефективно керувати дисковими ресурсами, аналізувати використання простору на рівнях компаній, служб, відділів та секторів, а також виявляти аномалії, такі як нецільове використання простору.

Важливість розробки такої системи обумовлена кількома факторами. По-перше, збільшення обсягів даних вимагає більш ефективних методів їх управління. По-друге, автоматизація процесів зменшує ймовірність людських помилок та підвищує точність аналізу. По-третє, можливість візуалізації даних дозволяє користувачам швидко отримувати необхідну інформацію та приймати обґрунтовані рішення щодо управління ресурсами.

Метою даної роботи є підвищення ефективності управління файловими сховищами в середовищі Windows Server, за рахунок розробки автоматизованої системи для аналізу дискових ресурсів. Запропонований підхід дозволить оптимізувати розподіл місця на дисках, забезпечити прозорий моніторинг файлових сховищ і своєчасно виявляти аномалії у використанні ресурсів.

Для досягнення цієї мети необхідно виконати ряд завдань, серед яких:

- проведення аналізу існуючих рішень для управління файловими сховищами та визначення їх переваг та недоліків.
- розробка архітектури системи, що забезпечить інтеграцію з існуючими серверними платформами та базами даних.
- реалізація компонентів системи, включаючи скрипти для збору даних, арі для обробки інформації та веб-інтерфейс для візуалізації.
- тестування системи для забезпечення її надійності та ефективності у різних умовах експлуатації.
- аналіз отриманих результатів та формулювання рекомендацій щодо подальшого розвитку системи.

Актуальність даного дослідження обумовлена зростаючими вимогами до ефективності управління ІТ-ресурсами в умовах цифрової трансформації

підприємств. Автоматизовані системи аналізу дозволяють суттєво підвищити рівень управління дисковими ресурсами, забезпечуючи їх оптимальне використання та зменшення витрат на зберігання даних. Крім того, впровадження таких систем сприяє підвищенню загальної продуктивності підприємства та забезпечує стабільну роботу IT-інфраструктури.

У даній роботі буде розглянуто основні компоненти системи, зокрема PowerShell-скрипти для збору даних, API на базі ASP.NET Core для обробки інформації та веб-інтерфейс для візуалізації результатів аналізу. Особлива увага буде приділена інтеграції цих компонентів, забезпеченню їх ефективної взаємодії та масштабованості системи для підтримки великої кількості серверів та обробки великих обсягів даних.

Таким чином, розробка автоматизованої системи аналізу файлових сховищ для середовища Windows Server представляє собою важливий внесок у сферу управління IT-ресурсами, сприяючи підвищенню ефективності та надійності роботи корпоративних інформаційних систем. Магістерська дисертація має на меті створення функціонального рішення для автоматизації управління файловими сховищами.

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Автоматизовані системи аналізу файлових сховищ є важливим інструментом для сучасних організацій, які використовують великі обсяги даних. Вони дозволяють ефективно управляти дисковим простором, забезпечуючи оптимальне використання ресурсів та зменшуючи витрати на зберігання даних. У контексті середовища Windows Server, такі системи мають особливе значення, оскільки вони інтегруються з існуючою інфраструктурою та забезпечують централізоване управління.

Основною метою автоматизованої системи аналізу файлових сховищ є збір, обробка та візуалізація інформації про використання дискового простору на серверах. Це дозволяє адміністраторам системи отримувати детальну аналітику щодо використання ресурсів на рівні компаній, служб, відділів та секторів. Такий підхід допомагає виявляти аномалії, наприклад, нецільове використання дискового простору, що може призвести до значних фінансових втрат.

Сучасні рішення в цій галузі включають використання скриптів для збору даних, API для їх обробки та збереження, а також веб-інтерфейсів для візуалізації. PowerShell-скрипти є популярним інструментом для збору даних у середовищі Windows Server, оскільки вони дозволяють автоматизувати процеси та працювати з віддаленими серверами. Використання API на базі ASP.NET Core забезпечує надійну обробку та збереження даних у базі даних Microsoft SQL Server. Веб-інтерфейси, створені за допомогою ASP.NET Core MVC та бібліотеки Chart.js, дозволяють користувачам отримувати зручний доступ до аналітики та візуалізації даних.

Аналіз існуючих рішень показує, що більшість з них орієнтовані на забезпечення масштабованості та гнучкості. Це досягається за рахунок модульної архітектури, яка дозволяє легко додавати нові сервери та вузли, а також адаптувати систему під конкретні потреби бізнесу. Важливим аспектом є також безпека даних, оскільки інформація про файлові сховища може містити конфіденційні дані. Тому сучасні системи забезпечують високий рівень захисту даних на всіх етапах їх обробки та зберігання.

Одним з ключових аспектів автоматизованих систем аналізу файлових сховищ є їх здатність до масштабування. Це означає, що система повинна бути здатна обробляти збільшення обсягів даних без втрати продуктивності. Для досягнення цього використовуються різні технології та підходи, такі як розподілені обчислення, оптимізація запитів до бази даних та кешування. Наприклад, використання Entity Framework Core дозволяє ефективно працювати з великими обсягами даних, забезпечуючи високу продуктивність та надійність.

Крім того, важливим аспектом є інтеграція з існуючими системами та інфраструктурою організації. Це включає підтримку різних версій Windows Server, сумісність з іншими програмними продуктами та можливість інтеграції з системами моніторингу та управління. Використання стандартних протоколів та інтерфейсів, таких як HTTP та REST API, забезпечує легкість інтеграції та взаємодії з іншими системами.

Важливим елементом автоматизованих систем аналізу файлових сховищ є також їх здатність до адаптації під конкретні потреби користувачів. Це включає можливість налаштування параметрів збору даних, фільтрації та візуалізації. Наприклад, користувачі можуть налаштовувати фільтри для відображення даних тільки для певних серверів, компаній або відділів. Це дозволяє отримувати більш точну та релевантну інформацію, що сприяє прийняттю обґрунтованих рішень.

Таким чином, автоматизовані системи аналізу файлових сховищ є невід'ємною частиною сучасної IT-інфраструктури. Вони дозволяють ефективно управляти ресурсами, зменшувати витрати та забезпечувати високий рівень безпеки даних.

Розробка та впровадження таких систем вимагає глибокого розуміння предметної області, аналізу існуючих рішень та інноваційного підходу до вирішення поставлених задач. Враховуючи всі ці аспекти, можна стверджувати, що автоматизовані системи аналізу файлових сховищ мають великий потенціал для подальшого розвитку та вдосконалення.

## 1.1 Опис предметного середовища та процесу діяльності

Предметне середовище автоматизованої системи аналізу файлових сховищ охоплює інфраструктуру серверів Windows, які використовуються в організаціях для зберігання та управління даними. Ці сервери можуть бути частиною корпоративної мережі, що включає різні підрозділи, такі як компанії, служби, відділи та сектори. Основною метою цієї інфраструктури є забезпечення надійного та безпечного зберігання даних, а також ефективного управління дисковим простором.

Процес діяльності в рамках цієї системи включає кілька ключових етапів:

а) збір даних;

1) на першому етапі PowerShell-скрипти запускаються на віддалених серверах Windows. Вони збирають інформацію про файлові системи, включаючи дані про диски, папки та вкладені файли. цей процес включає підрахунок розмірів папок та формування JSON-запитів з отриманими даними.

2) скрипти використовують команду Invoke-Command для виконання завдань на віддалених серверах, що дозволяє централізовано збирати дані з різних вузлів мережі.

б) передача даних:

1) зібрані дані передаються на сервер через http-запити до API. Цей етап включає формування та відправку JSON-запитів, які містять інформацію про файлові системи.

2) API, розроблений на базі ASP.NET Core, приймає ці запити та обробляє їх. Він перевіряє коректність даних, наприклад, довжину шляху та ім'я сервера, щоб забезпечити точність та надійність збереженої інформації.

в) обробка та збереження даних:

1) API обробляє отримані дані та зберігає їх у базі даних Microsoft SQL Server. Цей процес включає оновлення вже існуючих записів або додавання нових, залежно від отриманої інформації.

2) використання Entity Framework Core дозволяє ефективно працювати з базою даних, забезпечуючи високу продуктивність та надійність збереження даних.

г) візуалізація даних:

1) збережені дані відображаються у веб-інтерфейсі, розробленому на базі ASP.NET Core MVC. Користувачі можуть переглядати інформацію у вигляді таблиць та графіків, що дозволяє легко аналізувати використання дискового простору.

2) веб-інтерфейс використовує бібліотеку Chart.js для створення інтерактивних графіків, які допомагають візуалізувати дані та виявляти аномалії.

д) аналіз та прийняття рішень:

1) користувачі системи, такі як системні адміністратори та менеджери, можуть аналізувати зібрані дані для прийняття обґрунтованих рішень щодо управління дисковим простором. Це включає виявлення неефективного використання ресурсів, планування розширення сховищ та оптимізацію витрат на зберігання даних.

2) система дозволяє фільтрувати дані за різними критеріями, такими як сервери, компанії, служби та відділи, що забезпечує гнучкість та точність аналізу.

Таким чином, предметне середовище автоматизованої системи аналізу файлових сховищ охоплює всі аспекти управління дисковим простором у середовищі Windows Server. Процес діяльності включає збір, передачу, обробку, збереження та візуалізацію даних, що дозволяє ефективно управляти ресурсами та приймати обґрунтовані рішення. Впровадження такої системи сприяє підвищенню ефективності роботи організації, зменшенню витрат та забезпеченню високого рівня безпеки даних.

## 1.2 Аналіз існуючих рішень та результатів дослідження.

В дані роботі було розглянуто та проаналізовано три відомі системи для автоматизованого аналізу файлових сховищ у середовищі Windows Server: SolarWinds Storage Resource Monitor, TreeSize Professional та ManageEngine FileAudit Plus. Кожна з цих платформ має свої унікальні функції, переваги та обмеження, що робить їх популярними серед адміністраторів та ІТ-фахівців. Проведений аналіз дозволив оцінити їхню ефективність, технічні можливості та придатність для різних типів організаційних потреб.

### 1.2.1 SolarWinds Storage Resource Monitor

SolarWinds Storage Resource Monitor є потужним інструментом для моніторингу та управління ресурсами зберігання даних у реальному часі. Ця система забезпечує глибокий аналіз використання дискового простору, допомагаючи організаціям оптимізувати свої файлові сховища та уникати перевантажень [1].

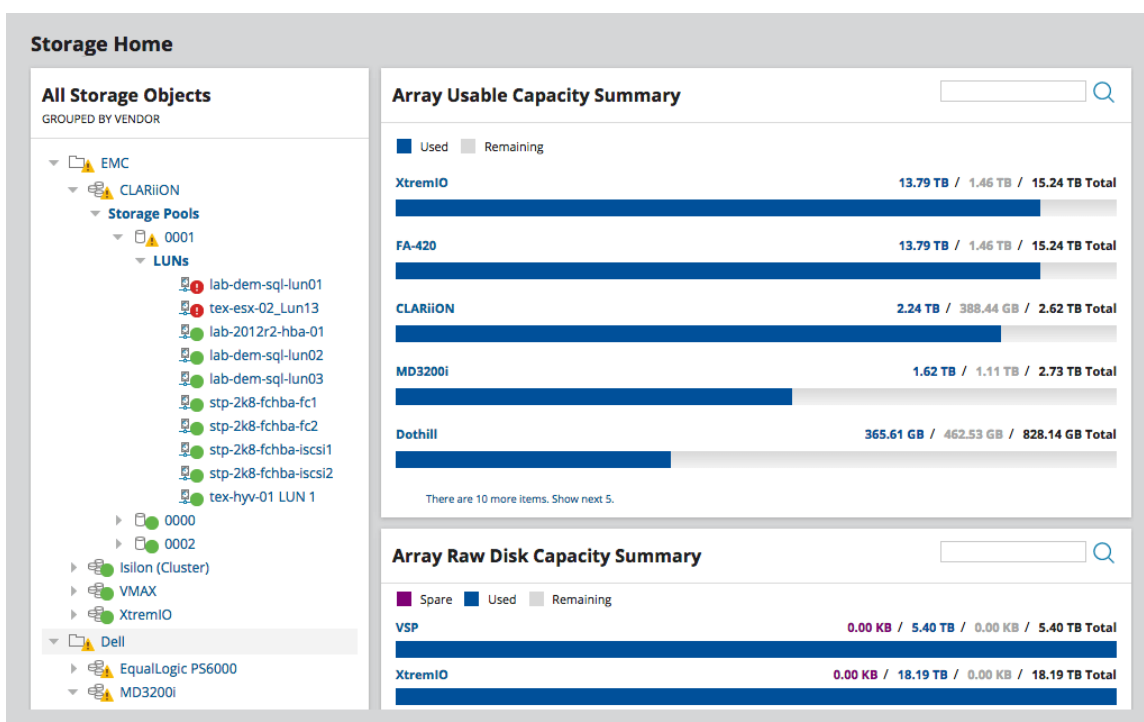


Рисунок 1.1 – Інтерфейс системи SolarWinds Storage Resource Monitor Home

SolarWinds Storage Resource Monitor є потужним та надійним інструментом для моніторингу та управління файловими сховищами в середовищі Windows Server. Його основні переваги полягають у можливості моніторингу в реальному часі, інтуїтивно зрозумілому інтерфейсі та потужній аналітиці, що робить його ідеальним вибором для великих організацій з високими вимогами до управління ресурсами зберігання. Проте, висока вартість ліцензії та складність налаштування можуть стати перешкодою для малих підприємств або проектів з обмеженим бюджетом. В цілому, SolarWinds SRM є ефективним рішенням для організацій, які прагнуть забезпечити оптимальне використання своїх файлових сховищ та підвищити ефективність управління ІТ-ресурсами.

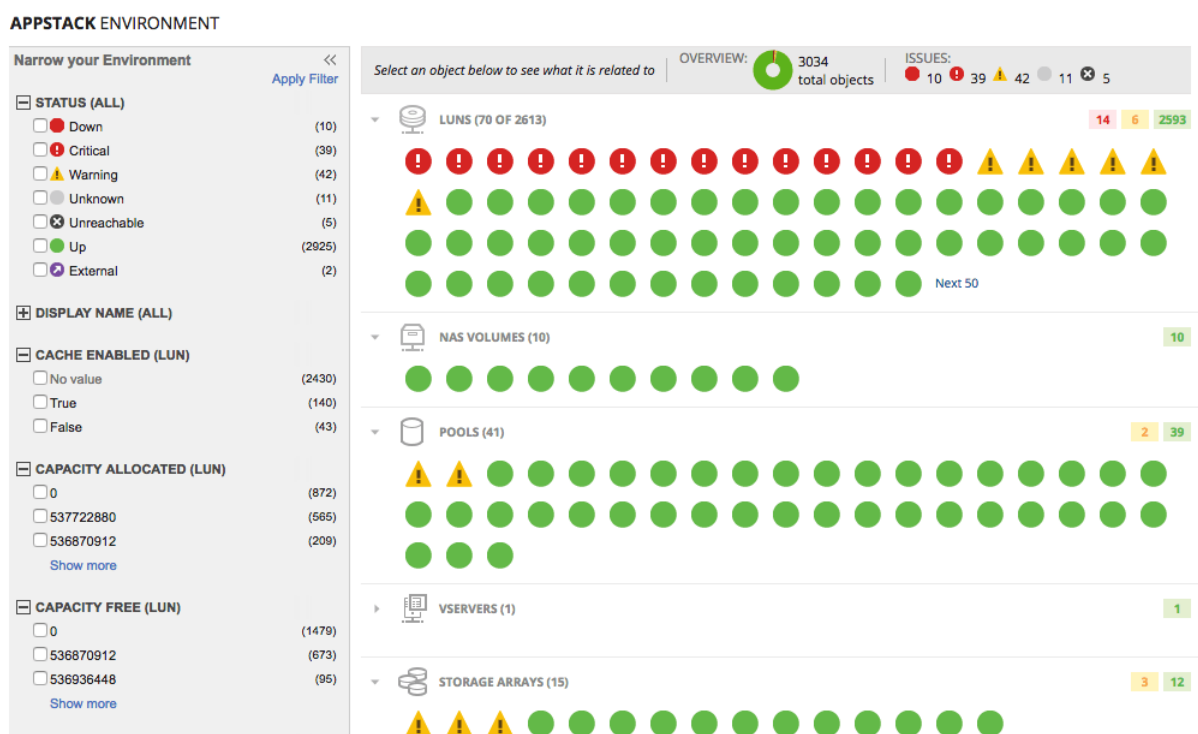


Рисунок 1.2 – Інтерфейс модуля системи SolarWinds Storage Resource Monitor APPSTACK ENVIROMENT

Незважаючи на багато переваг, SolarWinds Storage Resource Monitor також має деякі недоліки та обмеження:

- вартість ліцензії, як комерційний продукт, SolarWinds SRM може бути дорогим варіантом для малих та середніх підприємств з обмеженим бюджетом на ІТ-інфраструктуру.

- складність налаштування, для повноцінного використання всіх функцій системи необхідно мати певний рівень технічних знань та досвіду, що може вимагати додаткового часу на навчання персоналу.
- високі системні вимоги, для забезпечення стабільної роботи SRM може вимагати значних ресурсів серверного обладнання, що може бути недосяжним для деяких організацій.
- обмежена гнучкість в налаштуваннях, хоча система і є потужною, деякі користувачі можуть виявити обмеження у налаштуванні специфічних функцій або інтеграції з нестандартними системами.

Отже, хоча SolarWinds Storage Resource Monitor пропонує широкий спектр функцій для ефективного управління файловими сховищами та моніторингу використання ресурсів, його висока вартість, складність налаштування, вимогливість до системних ресурсів та обмежена гнучкість можуть зробити його менш привабливим для певних категорій користувачів. Для малих та середніх підприємств, які шукають більш доступні та гнучкі рішення, ці недоліки можуть стати вирішальними факторами при виборі альтернативних систем аналізу файлових сховищ.

### 1.2.2 TreeSize Professional

TreeSize Professional є популярним інструментом для аналізу дискового простору, який дозволяє швидко і ефективно оцінювати використання файлових сховищ на Windows Server. Цей інструмент особливо корисний для виявлення великих файлів та папок, що займають значний обсяг дискового простору[2].

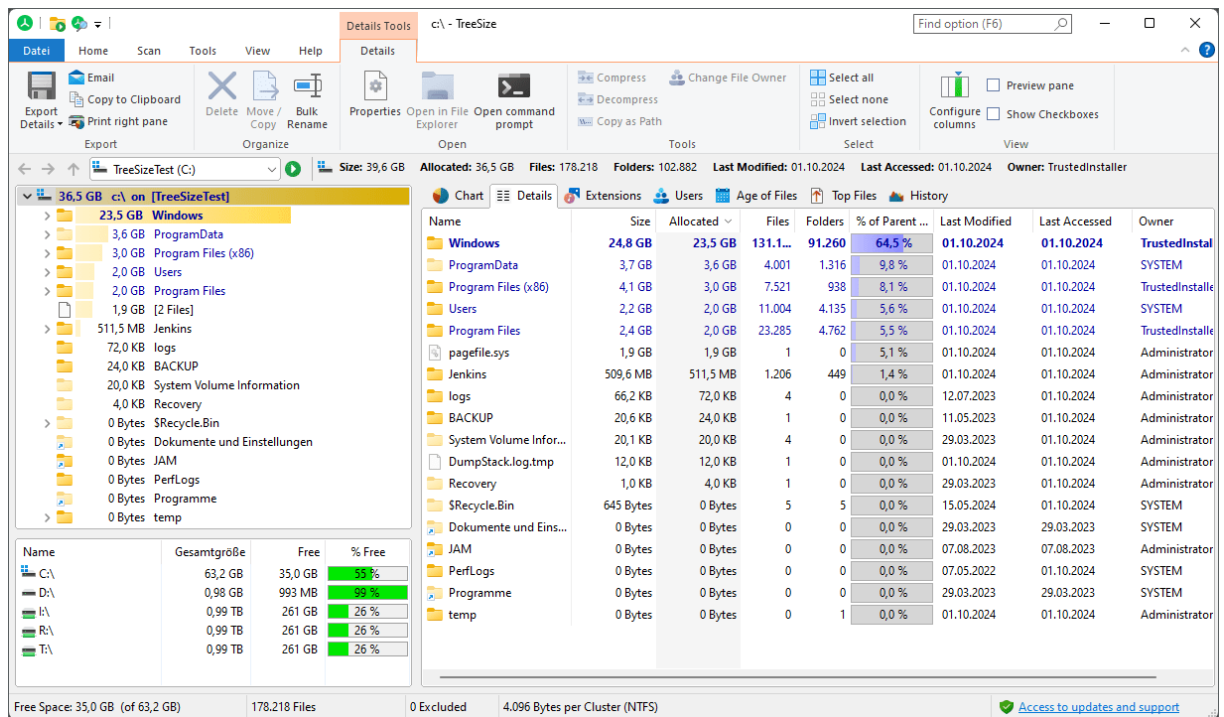


Рисунок 2.3 – Інтерфейс системи TreeSize Professional

До основних недоліків даного рішення можна віднести:

- Обмежені можливості моніторингу в реальному часі: На відміну від SolarWinds SRM, TreeSize Professional не забезпечує постійного моніторингу файлових систем у реальному часі, що може бути недоліком для організацій, які потребують оперативного реагування на зміни.
- Відсутність розширених функцій управління: TreeSize Professional більше зосереджений на аналізі дискового простору і не надає широких можливостей для управління файловими сховищами або інтеграції з іншими ІТ-системами.
- Ліцензійні обмеження: Повна версія TreeSize Professional є платною, хоча існують безкоштовні версії з обмеженим функціоналом, що може бути недостатнім для великих організацій.

TreeSize Professional є ефективним та простим у використанні інструментом для аналізу дискового простору, що робить його ідеальним вибором для організацій, які потребують швидкого та точного аналізу використання файлових сховищ. Його основні переваги полягають у швидкому скануванні дисків, гнучких можливостях

фільтрації та інтеграції з Windows Explorer. Проте, відсутність можливостей моніторингу в реальному часі та обмежені функції управління можуть стати перешкодою для організацій з більш складними вимогами до управління файловими сховищами. У цілому, TreeSize Professional є чудовим вибором для малих та середніх підприємств, які прагнуть ефективно управляти своїми дисковими ресурсами без необхідності значних інвестицій у складні системи моніторингу.

### 1.2.3 ManageEngine FileAudit Plus

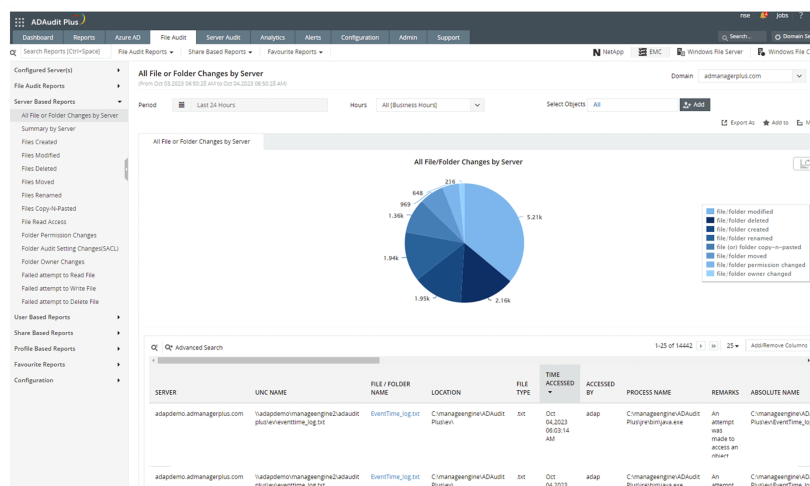


Рисунок 2.3 – Інтерфейс системи ManageEngine FileAudit Plus

ManageEngine FileAudit Plus є комплексним рішенням для моніторингу та аудиту доступу до файлів та папок у середовищі Windows Server. Цей інструмент дозволяє адміністраторам відстежувати та аналізувати дії користувачів щодо файлів, забезпечуючи високий рівень безпеки та відповідності нормативним вимогам [3].

Недоліки:

- вартість ліцензії: ManageEngine FileAudit Plus є комерційним продуктом, що може бути недоступним для малих підприємств з обмеженим бюджетом.

– складність налаштування: Для повного використання всіх функцій системи необхідно здійснити ретельне налаштування, що може вимагати додаткових технічних ресурсів та часу.

– обмежені можливості аналізу дискового простору: Хоча FileAudit Plus забезпечує потужні можливості для аудиту доступу до файлів, він не пропонує таких ж інструментів для аналізу використання дискового простору, як TreeSize Professional.

Отже, ManageEngine FileAudit Plus є потужним інструментом для аудиту доступу до файлів та папок, що робить його ідеальним вибором для організацій, які прагнуть підвищити рівень безпеки своїх файлових сховищ та відповідати нормативним вимогам. Його основні переваги полягають у комплексному аудиті доступу, детальній звітності та автоматизації сповіщень, що дозволяє адміністраторам ефективно управляти безпекою даних. Проте, висока вартість ліцензії та обмежені можливості аналізу дискового простору можуть бути недоліками для деяких організацій. У цілому, ManageEngine FileAudit Plus є чудовим вибором для середніх та великих підприємств, які потребують потужних інструментів для аудиту та моніторингу доступу до файлових ресурсів.

Проведений огляд існуючих рішень для автоматизованого аналізу файлових сховищ у середовищі Windows Server показує, що на ринку представлено різноманітні інструменти з різними функціональними можливостями та ціновими категоріями. SolarWinds Storage Resource Monitor виділяється своєю потужною системою моніторингу в реальному часі та інтегрованими інструментами звітності, проте є дорогим та вимагає високого рівня технічних знань для налаштування. TreeSize Professional є відмінним вибором для швидкого та ефективного аналізу дискового простору з простим інтерфейсом, але не надає розширених можливостей моніторингу та управління. ManageEngine FileAudit Plus забезпечує комплексний аудит доступу до файлів та високий рівень безпеки, але також має високі ліцензійні витрати та обмежені можливості аналізу дискового простору.

Вибір найбільш підходящої системи залежить від конкретних потреб організації, її бюджету та технічних ресурсів. Для проекту "Автоматизована

система аналізу файлових сховищ для середовища Windows Server" важливо врахувати як функціональні можливості існуючих рішень, так і їхню сумісність з цільовою архітектурою системи. Розробка власного рішення дозволить адаптувати функціонал до специфічних вимог проекту, забезпечуючи гнучкість, масштабованість та інтеграцію з іншими компонентами IT-інфраструктури.

Таким чином, аналіз існуючих рішень підкреслює необхідність створення індивідуалізованої системи, яка поєднує переваги різних інструментів та враховує унікальні вимоги користувачів у середовищі Windows Server. Це забезпечить ефективне управління файловими сховищами, оптимізацію використання дискового простору та підвищення загальної продуктивності IT-інфраструктури підприємства.

### 1.3 Постановка задачі

Розробка автоматизованої системи аналізу файлових сховищ для середовища Windows Server вимагає чіткого визначення задач, які необхідно вирішити для досягнення поставлених цілей. Постановка задачі включає визначення призначення та цілей розробки, а також інформаційного забезпечення системи.

#### 1.3.1 Призначення та цілі розробки

Основним призначенням автоматизованої системи аналізу файлових сховищ є забезпечення ефективного управління дисковим простором на серверах Windows. Це досягається шляхом автоматичного збору, обробки та візуалізації даних про файлові системи. Система повинна надавати користувачам можливість отримувати детальну аналітику щодо використання дискового простору на рівні компаній, служб, відділів та секторів.

Цілі розробки системи включають:

- а) автоматизація збору даних;

1) розробка PowerShell-скриптів для автоматичного збору інформації про диски та папки на віддалених серверах Windows;

2) забезпечення можливості централізованого збору даних з різних серверів;

б) обробка та збереження даних;

1) створення API на базі ASP.NET Core для прийому та обробки даних, зібраних PowerShell-скриптами;

2) використання Microsoft SQL Server для надійного збереження даних про файлові системи;

в) візуалізація даних;

1) розробка веб-інтерфейсу для відображення зібраних даних у вигляді таблиць та графіків;

2) використання бібліотеки Chart.js для створення інтерактивних графіків, що дозволяють візуалізувати використання дискового простору;

г) аналіз та оптимізація;

1) надання користувачам можливості аналізувати дані для виявлення неефективного використання дискового простору;

2) розробка інструментів для оптимізації використання ресурсів та зменшення витрат на зберігання даних.

### 1.3.2 Інформаційне забезпечення

Інформаційне забезпечення системи включає всі необхідні дані та ресурси для її функціонування. Це включає:

а) дані про файлові системи;

1) інформація про диски (назва диска, загальний розмір, вільне та використане місце);

2) інформація про папки (повний шлях до папки, розмір папки, рівень: компанія, служба, відділ, сектор);

б) технологічні ресурси;

- 1) PowerShell-скрипти для збору даних;
  - 2) API на базі ASP.NET Core для обробки та збереження даних;
  - 3) база даних Microsoft SQL Server для зберігання інформації;
  - 4) веб-інтерфейс на базі ASP.NET Core MVC для візуалізації даних;
- в) інфраструктурні ресурси;
- 1) сервери Windows, на яких запускаються PowerShell-скрипти;
  - 2) сервери для розміщення API та бази даних;
  - 3) веб-сервери для розміщення веб-інтерфейсу;
- г) безпека даних;
- 1) забезпечення захисту даних на всіх етапах їх обробки та зберігання;
  - 2) використання механізмів аутентифікації та авторизації для контролю доступу до системи.

Таким чином, постановка задачі включає визначення призначення та цілей розробки автоматизованої системи аналізу файлових сховищ, а також інформаційного забезпечення, необхідного для її функціонування. Це дозволяє чітко окреслити обсяг робіт та забезпечити успішну реалізацію проекту.

## Висновки до розділу 1

У першому розділі було проведено детальний аналіз предметної області автоматизованих систем аналізу файлових сховищ для середовища Windows Server. Було визначено, що такі системи є важливим інструментом для ефективного управління дисковим простором, забезпечуючи оптимальне використання ресурсів та зменшуючи витрати на зберігання даних. Основною метою роботи системи є автоматичний збір, обробка та візуалізація інформації про файлові системи, що дозволяє адміністраторам отримувати детальну аналітику щодо використання ресурсів на рівні компаній, служб, відділів та секторів.

Аналіз існуючих рішень показав, що більшість з них орієнтовані на забезпечення масштабованості та гнучкості, що досягається за рахунок модульної архітектури та використання сучасних технологій, таких як PowerShell [4] для

збору даних, ASP.NET Core для розробки API та Microsoft SQL Server для збереження даних. Важливим аспектом є також безпека даних, оскільки інформація про файлові сховища може містити конфіденційні дані, тому сучасні системи забезпечують високий рівень захисту на всіх етапах обробки та зберігання даних.

Постановка задачі включала визначення призначення та цілей розробки системи, а також інформаційного забезпечення, необхідного для її функціонування. Основними цілями розробки є автоматизація збору даних, обробка та збереження інформації, візуалізація даних та аналіз для оптимізації використання ресурсів. Інформаційне забезпечення включає дані про файлові системи, технологічні та інфраструктурні ресурси, а також заходи безпеки для захисту даних.

Таким чином, проведений аналіз предметної області та постановка задачі дозволяють чітко окреслити обсяг робіт та забезпечити успішну реалізацію проекту автоматизованої системи аналізу файлових сховищ для середовища Windows Server.

## 2 ФОРМУВАННЯ СИСТЕМНИХ ВИМОГ

Формування системних вимог є критичним етапом у процесі розробки програмного забезпечення. На цьому етапі визначаються основні функції та характеристики системи, які забезпечують відповідність потребам користувачів та бізнес-цілям організації. Чітко сформульовані вимоги дозволяють встановити спільне розуміння між замовником та командою розробників щодо очікуваної поведінки системи та її характеристик.

### 2.1 Функціональні вимоги

Функціональні вимоги визначають, як система повинна виконувати певні функції та які завдання вона повинна виконувати для задоволення потреб користувачів чи бізнес-вимог. У випадку розроблюваної системи аналізу файлових сховищ для середовища Windows Server, функціональні вимоги зосереджені на забезпеченні ефективного збору, обробки та візуалізації даних про файлові сховища, а також на управлінні користувачами та їх доступом до інформації (таблиця 2.1).

Таблиця 2.1. - Функціональні вимоги до системи аналізу файлових сховищ

№	Функціональна вимога	Опис
1	Збір даних про файлові сховища	<p>- Автоматичний збір інформації: Система повинна автоматично збирати дані про дискові ресурси, включаючи розміри дисків, використане та вільне місце, а також структуру папок та файлів на сервері Windows.</p> <p>- Підключення до віддалених серверів: Забезпечити можливість підключення до кількох віддалених серверів для збору інформації з різних джерел.</p>

№	Функціональна вимога	Опис
2	Обробка та аналіз даних	<ul style="list-style-type: none"> <li>- Обробка зібраних даних: Система повинна аналізувати зібрані дані для виявлення трендів використання дискового простору та ідентифікації аномалій, таких як нецільове використання ресурсів.</li> <li>- Фільтрація нерелевантних даних: Забезпечити механізми фільтрації непотрібної інформації для покращення точності аналізу.</li> </ul>
3	Візуалізація даних	<ul style="list-style-type: none"> <li>- Графічне представлення: Система повинна надавати зручні графічні інтерфейси, такі як діаграми та графіки, для візуалізації використання дискового простору на різних рівнях (компанія, служба, відділ, сектор).</li> <li>- Динамічне оновлення: Забезпечити можливість динамічного оновлення візуалізацій в режимі реального часу.</li> </ul>
4	Управління користувачами	<ul style="list-style-type: none"> <li>- Ролі та права доступу: Система повинна підтримувати різні рівні доступу для користувачів, включаючи адміністратора, менеджера та звичайного користувача.</li> <li>- Аутентифікація та авторизація: Забезпечити безпечні методи аутентифікації та авторизації для захисту доступу до системи.</li> </ul>
5	Інтеграція з іншими системами	<ul style="list-style-type: none"> <li>- API для взаємодії: Надати API для інтеграції з іншими IT-системами та інструментами управління.</li> <li>- Імпорт та експорт даних: Забезпечити можливість імпорту та експорту даних у різних форматах, таких як JSON та CSV.</li> </ul>

№	Функціональна вимога	Опис
6	Звіти та аналітика	<ul style="list-style-type: none"> <li>- Генерація звітів: Система повинна генерувати детальні звіти щодо використання дисків, включаючи інформацію про найбільші файли та папки, тренди використання простору та інші ключові метрики.</li> <li>- Планування звітів: Забезпечити можливість планування автоматичного створення та розсилки звітів у визначені терміни</li> </ul>

## 2.2 Нефункціональні вимоги

Нефункціональні вимоги визначають атрибути та властивості системи, які не стосуються конкретних функцій чи задач, але впливають на її продуктивність, ефективність та якість. У контексті автоматизованої системи аналізу файлових сховищ для середовища Windows Server нефункціональні вимоги включають аспекти надійності, продуктивності, масштабованості, безпеки та зручності використання (таблиця 2.2).

Таблиця 2.2. - Нефункціональні вимоги до системи аналізу файлових сховищ

№	Нефункціональна вимога	Опис
1	Продуктивність	<ul style="list-style-type: none"> <li>- Швидкість обробки даних: Система повинна обробляти великі обсяги даних про файлові сховища з мінімальними затримками.</li> <li>- Час відгуку: Час відгуку системи на запити користувачів не повинен перевищувати 2 секунд.</li> </ul>

№	Нефункціональна вимога	Опис
2	Масштабованість	<p>- Горизонтальне та вертикальне масштабування: Система повинна підтримувати можливість масштабування як за горизонталлю (додавання нових серверів), так і за вертикаллю (збільшення ресурсів існуючих серверів) для забезпечення обробки зростаючих обсягів даних.</p> <p>- Висока доступність: Система повинна бути доступною 99.9% часу, з мінімальними перервами на обслуговування.</p>
3	Безпека	<p>- Захист даних: Система повинна забезпечувати шифрування даних як при зберіганні, так і при передачі.</p> <p>- Управління доступом: Використання ролей та прав доступу для контролю доступу користувачів до різних функцій та даних системи.</p> <p>- Виявлення та запобігання загрозам: Вбудовані механізми для виявлення потенційних загроз безпеці та автоматичне реагування на них.</p>
4	Надійність	<p>- Відновлення після збоїв: Система повинна мати механізми для швидкого відновлення після збоїв або втрати даних.</p> <p>- Резервне копіювання: Регулярне резервне копіювання даних для забезпечення їхнього збереження у випадку аварійних ситуацій.</p>

№	Нефункціональна вимога	Опис
5	Зручність використання (Usability)	<ul style="list-style-type: none"> <li>- Інтуїтивно зрозумілий інтерфейс: Інтерфейс користувача повинен бути простим та зрозумілим, забезпечуючи легкий доступ до всіх функцій системи.</li> <li>- Документація та підтримка: Наявність детальної документації та доступ до технічної підтримки для користувачів та адміністраторів.</li> </ul>
6	Сумісність	<ul style="list-style-type: none"> <li>- Підтримка різних платформ: Система повинна бути сумісною з різними версіями Windows Server та іншими суміжними системами.</li> <li>- Інтеграція з існуючими ІТ-системами: Забезпечення безшовної інтеграції з поточними ІТ-системами організації, такими як бази даних, системи моніторингу та інші корпоративні застосунки.</li> </ul>
7	Відмовостійкість	<ul style="list-style-type: none"> <li>- Резервні компоненти: Використання резервних серверів та компонентів для забезпечення безперервної роботи системи у випадку відмови основних елементів.</li> <li>- Моніторинг та логування: Впровадження систем моніторингу та логування для постійного контролю стану системи та швидкого виявлення відмов.</li> </ul>
8	Економічна ефективність	<ul style="list-style-type: none"> <li>- Оптимізація витрат: Система повинна бути оптимізована для мінімізації витрат на експлуатацію та обслуговування, включаючи витрати на обладнання та ліцензії.</li> <li>- Енергоефективність: Забезпечення енергоефективної роботи серверів та інших компонентів системи для зменшення загальних витрат на енергоспоживання.</li> </ul>

### 2.3 Основні сутності системи

Основні сутності системи – це основні об'єкти, які взаємодіють між собою в рамках розробленої структури. Вони є базовими елементами, необхідними для забезпечення ефективної роботи системи та досягнення її цілей. Ці компоненти виконують конкретні завдання, обмінюються даними і сприяють виконанню загальних функцій системи.

Зрозуміти ці компоненти є критично важливим етапом на шляху до створення системи, оскільки це дозволяє чітко визначити, як вони взаємодіють і як кожен з них впливає на загальний процес. Тому дуже важливо правильно описати кожен компонент, визначити його функції та взаємозв'язки з іншими частинами системи.[5]

Таблиця 2.3. - Основні сутності системи аналізу файлових сховищ

Сутність	Опис ролі	Атрибути	Відносини з іншими сутностями
Сервер (Server)	Представляє сервер Windows, з якого збираються дані про файлові сховища.	- Ідентифікатор сервера - Ім'я сервера - IP-адреса - Розташування - Опис	- Має багато Дисків - Пов'язаний з Папками та Файлами
Диск (Disk)	Відображає фізичний або логічний диск на сервері, де зберігаються дані.	- Ідентифікатор диска - Літерний ідентифікатор - Загальний обсяг - Використаний обсяг - Вільний обсяг	- Належить до Сервера - Містить багато Папок

Сутність	Опис ролі	Атрибути	Відносини з іншими сутностями
Папка (Folder)	Відображає директорію на диску, яка може містити інші папки або файли.	<ul style="list-style-type: none"> <li>- Ідентифікатор папки</li> <li>- Шлях до папки</li> <li>- Розмір папки</li> <li>- Дата створення</li> <li>- Власник</li> </ul>	<ul style="list-style-type: none"> <li>- Міститься на Диску</li> <li>- Може містити багато Файлів</li> <li>- Пов'язана з Відділом</li> </ul>
Файл (File)	Представляє файл, збережений у системі файлових сховищ.	<ul style="list-style-type: none"> <li>- Ідентифікатор файлу</li> <li>- Назва файлу</li> <li>- Розмір файлу</li> <li>- Розширення</li> <li>- Дата створення/зміни</li> <li>- Власник</li> </ul>	<ul style="list-style-type: none"> <li>- Міститься в Папці</li> <li>- Може бути пов'язаний з Користувачем</li> </ul>
Відділ (Department)	Відображає організаційну одиницю підприємства, яка використовує ресурси файлових сховищ.	<ul style="list-style-type: none"> <li>- Ідентифікатор відділу</li> <li>- Назва відділу</li> <li>- Опис</li> <li>- Менеджер відділу</li> </ul>	<ul style="list-style-type: none"> <li>- Має багато Користувачів</li> <li>- Пов'язаний з Папками</li> </ul>
Користувач (User)	Представляє особу, яка взаємодіє з системою та має певні права доступу.	<ul style="list-style-type: none"> <li>- Ідентифікатор користувача</li> <li>- Ім'я</li> </ul>	<ul style="list-style-type: none"> <li>- Належить до Відділу</li> <li>- Може володіти</li> </ul>

Сутність	Опис ролі	Атрибути	Відносини з іншими сутностями
		користувача - Електронна пошта - Роль (адміністратор, менеджер, користувач) - Статус активності	Файлами - Має Ролі та права доступу
Роль (Role)	Визначає набір прав та привілеїв, які має користувач у системі.	- Ідентифікатор ролі - Назва ролі - Опис прав доступу	Призначається Користувачу - Визначає доступ до Сутностей системи

Повний перелік сутностей, виокремлених під час проектування системи, наведено в Додатку А.

У зв'язку з необхідністю детального опису взаємодії користувачів та системи було вирішено представити ключові сценарії функціонування системи у вигляді діаграм варіантів використання та діаграм діяльності. Це дозволить чітко продемонструвати процеси збору, обробки та візуалізації даних, що є важливими етапами роботи системи. Використання таких діаграм допомагає наочно представити складні процеси, покращити розуміння взаємозв'язків між основними компонентами та підвищити ефективність комунікації серед розробників, адміністраторів та кінцевих користувачів системи. Дані діаграми, представлені в додатках Б та В, сприяють глибшому аналізу та забезпечують прозорість у розумінні функціоналу системи.

## Висновки до розділу 2

Другий розділ надає детальний огляд формування системних вимог для розроблюваної автоматизованої системи аналізу файлових сховищ для середовища Windows Server. У першому підрозділі були розглянуті функціональні та нефункціональні вимоги до системи, що визначають її основні можливості та характеристики роботи. Функціональні вимоги спрямовані на забезпечення ефективного збору, обробки та візуалізації даних, а також на управління користувачами та їх доступом. Нефункціональні вимоги зосереджені на забезпеченні продуктивності, масштабованості, безпеки, надійності та зручності використання системи.

Сценарії використання описують конкретні приклади взаємодії користувачів з системою, що дозволяє краще зрозуміти, як система має працювати в різних ситуаціях. Це допомагає визначити очікувані результати та перевірити відповідність системи поставленим вимогам.

Під час розробки цього розділу, а також під час виокремлення основних сутностей, стало необхідним створення детальних сценаріїв взаємодії з системою. Для наочності та кращого розуміння функціоналу було вирішено представити ці сценарії у вигляді діаграм варіантів використання та діаграм діяльності, які наочно ілюструють процес збору, обробки та візуалізації даних.

Таким чином, другий розділ окреслює етапи формування системних вимог. Здійснення комплексного аналізу вимог, визначення сутностей та розробка сценаріїв використання створюють основу для подальшого проектування та розробки системи. Цей розділ є базою для розуміння потреб і очікувань до системи, що забезпечує комунікацію між командами розробників та замовниками.

### 3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

Математичне забезпечення є критично важливим аспектом розробки автоматизованої системи аналізу файлових сховищ. Воно дозволяє здійснювати точні розрахунки, які необхідні для ефективного планування бюджетування, аналізу використання дискового простору та оптимізації витрат на зберігання даних. У цьому розділі розглянемо основні формули та методи, які використовуються в системі для досягнення цих цілей.[6]

#### 3.1 Розрахунок вартості зайнятого дискового простору

Одним з ключових аспектів системи є можливість розрахунку вартості зайнятого дискового простору для різних відділів за формулою (3.1). Це дозволяє ефективно планувати бюджет та оптимізувати витрати на зберігання даних.

Формула розрахунку загальної вартості дискового простору:

$$C = S \times P \quad (3.1)$$

де:

$C$  — загальна вартість зайнятого дискового простору (гривні);

$S$  — обсяг зайнятого дискового простору (в гігабайтах, ГБ);

$P$  — вартість одного гігабайта дискового простору (гривні за ГБ).

Обсяг дискового простору ( $S$ ) визначається за допомогою збору даних про використання дисків на сервері, що здійснюється PowerShell-скриптами.

Вартість одного гігабайта ( $P$ ) визначається на основі ринкових цін або внутрішніх тарифів компанії на зберігання даних.

Приклад розрахунку наведено у формулі (3.2):

Припустимо, що відділ А використовує 500 ГБ дискового простору, а вартість одного ГБ становить 2 гривні.

$$C_A = 500 \times 2 = 1000 \text{ грн} \quad (3.2)$$

Таким чином, загальна вартість зайнятого дискового простору для відділу А складає 1000 гривень.

### 3.2 Розподіл витрат між відділами

Для справедливого розподілу витрат на дисковий простір між різними відділами організації використовується аналогічна формула, яка враховує обсяг використання кожного відділу, приклад наведено у формулі (3.3).

Формула розподілу витрат:

$$C_i = S_i \times P \quad (3.3)$$

де:

$C_i$  — вартість зайнятого дискового простору для  $i$ -го відділу (гривні);

$S_i$  — обсяг зайнятого дискового простору  $i$ -го відділу (в ГБ);

$P$  — вартість одного гігабайта дискового простору (гривні за ГБ);

Обсяг дискового простору кожного відділу ( $S_i$ ) визначається за допомогою автоматизованого збору даних.

Вартість одного ГБ ( $P$ ) залишається однаковою для всіх відділів, що забезпечує рівномірний розподіл витрат.

Приклад розподілу витрат наведено в формулі (3.4):

Припустимо, що в організації є три відділи:

- Відділ А використовує 500 ГБ.
- Відділ Б використовує 300 ГБ.
- Відділ В використовує 200 ГБ.

Вартість одного ГБ становить 2 гривні.

$$\begin{aligned}
 C_A &= 500 \times 2 = 1000 \text{ грн} \\
 C_B &= 300 \times 2 = 600 \text{ грн} \\
 C_V &= 200 \times 2 = 400 \text{ грн}
 \end{aligned}
 \tag{3.4}$$

Отже, витрати на дисковий простір розподіляються між відділами наступним чином:

- Відділ А — 1000 грн,
- Відділ Б — 600 грн,
- Відділ В — 400 грн.

### 3.3 Планування бюджетування

Для ефективного планування бюджетування на основі прогнозованого зростання обсягу даних використовується формула, яка враховує поточний обсяг дискового простору та очікуване зростання, приклад наведено в формулі (3.5).

Формула планування бюджетування:

$$B = (S_{current} + \Delta S) \times P \tag{3.5}$$

де:

$B$  — прогнозований бюджет на зберігання даних (гривні);

$S_{current}$  — поточний обсяг зайнятого дискового простору (в ГБ);

$\Delta S$  — прогнозоване зростання обсягу даних (в ГБ);

$P$  — вартість одного гігабайта дискового простору (гривні за ГБ);

Поточний обсяг дискового простору ( $S_{current}$ ) визначається за допомогою системи моніторингу.

Прогнозоване зростання ( $\Delta S$ ) базується на історичних даних та прогнозах щодо зростання обсягів даних у майбутньому.

Вартість одного ГБ ( $P$ ) визначається як вартість зберігання одного ГБ на поточний момент часу.

Приклад планування бюджетування наведено в формулі (3.6):

Припустимо, що поточний обсяг дискового простору становить 1000 ГБ, прогнозоване зростання — 200 ГБ, а вартість одного ГБ — 2 гривні.

$$B = (1000 + 200) \times 2 = 1200 \times 2 = 2400 \text{ гривень} \quad (3.6)$$

Отже, прогнозований бюджет на зберігання даних складає 2400 гривень.

### 3.4 Аналіз ефективності використання дискового простору

Для оцінки ефективності використання дискового простору застосовується коефіцієнт ефективності, який дозволяє визначити, наскільки ефективно використовуються доступні ресурси.

Формула коефіцієнта ефективності (3.7):

$$E = \frac{S_{used}}{S_{total}} \quad (3.7)$$

де:

$E$  — коефіцієнт ефективності використання дискового простору (безрозмірний, від 0 до 1 або від 0% до 100%);

$S_{used}$  — обсяг використаного дискового простору (в ГБ);

$S_{total}$  — загальний обсяг доступного дискового простору (в ГБ).

Пояснення:

Обсяг використаного дискового простору ( $S_{used}$ ) визначається за допомогою системи моніторингу.

Загальний обсяг дискового простору ( $S_{total}$ ) — це сума всіх доступних ресурсів дискового простору на сервері або в системі.

Інтерпретація результатів:

$E$  близький до 1 (або 100%) означає, що дисковий простір використовується максимально ефективно, але також може свідчити про ризик перевантаження.

$E$  значно менший за 1 (або 100%) вказує на недоцільне використання ресурсів, що може бути оптимізовано для зменшення витрат.

Приклад розрахунку ефективності наведено в формулі (3.8):

Припустимо, що використано 800 ГБ з загальних 1000 ГБ.

$$E = \frac{800}{1000} = 0.8 \text{ або } 80\% \quad (3.8)$$

Отже, коефіцієнт ефективності використання дискового простору становить 80%, що вказує на досить ефективне використання ресурсів, але з можливістю подальшої оптимізації.

### 3.5 Додаткові методи та формули

Крім основних формул, описаних вище, для більш комплексного аналізу та управління дисковим простором можуть використовуватись додаткові методи та формули:

#### 3.5.1 Розрахунок середнього зростання даних

Для прогнозування майбутніх потреб у дисковому просторі корисно визначити середній ріст обсягу даних за певний період, приклад наведений у формулі (3.9).

$$\Delta S_{avg} = \frac{S_{new} - S_{old}}{N} \quad (3.9)$$

де:

$\Delta S_{avg}$  — середній приріст обсягу даних за один період (в ГБ),

$S_{new}$  — обсяг даних після певного періоду (в ГБ),

$S_{old}$  — обсяг даних до початку періоду (в ГБ),

$N$  — кількість періодів (місяців, років тощо).

### 3.5.2 Розрахунок часу до повного заповнення дискового простору

Для запобігання перевантаженню системи важливо знати, скільки часу залишилося до повного заповнення дискового простору та для розрахунку використовувати формулу (3.10).

$$T = \frac{S_{total} - S_{used}}{\Delta S_{avg}} \quad (3.10)$$

де:

$T$  — час до повного заповнення дискового простору (у відповідних одиницях часу, наприклад, місяці);

$S_{total}$  — загальний обсяг доступного дискового простору (в ГБ);

$S_{used}$  — поточний обсяг використаного дискового простору (в ГБ);

$\Delta S_{avg}$  — середній приріст обсягу даних за один період (в ГБ).

Приклад розрахунку часу до повного заповнення наведено в формулі (3.11):

Припустимо, що загальний обсяг дискового простору становить 2000 ГБ, використано 1500 ГБ, а середній приріст обсягу даних за місяць — 100 ГБ.

$$T = \frac{2000 - 1500}{100} = \frac{500}{100} = 5 \text{ місяців} \quad (3.11)$$

Отже, за прогнозом, дисковий простір буде повністю заповнений через 5 місяців.

### 3.6 Впровадження формул у систему

Для реалізації математичних розрахунків у системі використовуються наступні кроки:

а) збір даних;

1) використання powershell-скриптів для автоматичного збору інформації про використання дискового простору на сервері;

2) дані передаються через api на серверну частину системи;

б) зберігання даних;

1) використання microsoft sql server для зберігання зібраних даних у структурованих таблицях;

2) забезпечення доступу до даних для подальшого аналізу та розрахунків;

в) обробка даних;

1) використання мови програмування c# та фреймворку asp.net core для реалізації логіки розрахунків;

2) застосування entity framework core для взаємодії з базою даних та виконання запитів;

г) візуалізація результатів;

1) використання веб-інтерфейсу для представлення результатів розрахунків у вигляді таблиць, діаграм та звітів;

2) забезпечення можливості динамічного перегляду даних та їх фільтрації за різними параметрами (відділи, типи дисків тощо);

д) автоматизація та оновлення даних;

1) регулярне оновлення даних про використання дискового простору через автоматичні запуски powershell-скриптів;

2) забезпечення актуальності розрахунків та звітів для прийняття обґрунтованих рішень.

Висновок до розділу 3

Математичне забезпечення автоматизованої системи аналізу файлових сховищ грає ключову роль у забезпеченні ефективного управління дисковими ресурсами. Використання відповідних формул та методів дозволяє точно розраховувати вартість зайнятого дискового простору, справедливо розподіляти

витрати між відділами, планувати бюджетування та аналізувати ефективність використання ресурсів. Це сприяє оптимізації витрат, підвищенню продуктивності та забезпеченню стабільної роботи ІТ-інфраструктури підприємств.

У підсумку, розроблене математичне забезпечення не лише задовольняє поточні вимоги до системи аналізу файлових сховищ, але й закладає основу для подальшого розвитку та вдосконалення системи. Воно дозволяє інтегрувати додаткові функції, такі як більш складні методи машинного навчання або інтеграція з іншими системами управління ІТ-ресурсами.

Ретельне опрацювання математичних аспектів забезпечує конкурентні переваги системи на ринку, сприяє підвищенню ефективності використання ІТ-ресурсів та підтримує стратегічні цілі організації щодо оптимізації операцій та зниження витрат.

## 4 ПРОЄКТУВАННЯ СИСТЕМИ

Проєктування системи є ключовим етапом розробки автоматизованої системи аналізу файлових сховищ для середовища Windows Server. На цьому етапі визначаються основні архітектурні рішення, вибираються технології, які забезпечать ефективну інтеграцію всіх компонентів системи, а також розробляються сценарії використання, які демонструють практичні аспекти роботи системи у реальних умовах. [7]

### 4.1 Сценарії використання системи

Таблиця 4.1 – Сценарії використання системи

№	Назва сценарію	Учасники	Мета
1	Збір даних про файлові сховища	Адміністратор ІТ-системи	Автоматизований збір інформації про використання дискового простору на серверах Windows Server
2	Перегляд звітів про використання дискового простору	Керівник відділу	Аналіз використання дискового простору для планування ресурсів та бюджетування
3	Планування бюджету на зберігання даних	Фінансовий аналітик	Планування бюджету на зберігання даних на основі поточного використання та прогнозованого зростання
4	Отримання сповіщень про аномалії у використанні дискового простору	Адміністратор ІТ-системи	Оперативне виявлення та реагування на аномалії у використанні дискового простору

№	Назва сценарію	Учасники	Мета
5	Запит інформації про використання дискового простору	Користувач (співробітник відділу)	Перегляд та аналіз використання дискового простору у своєму відділі для покращення організації роботи з даними

#### 4.1.1 Сценарій 1: Збір даних про файлові сховища

а) учасники: адміністратор іт-системи;

б) мета: автоматизований збір інформації про використання дискового простору на серверах Windows Server;

в) кроки;

1) запуск PowerShell-скрипту: адміністратор ініціює виконання скрипту на центральному сервері або робочій станції;

2) підключення до віддалених серверів: скрипт використовує Invoke-Command для підключення до цільових серверів;

3) збір та фільтрація даних: скрипт виконує команди для збору інформації про дискові розділи, папки та файли, включаючи їхні розміри та структуру, а також фільтрує нерелевантні дані;

4) формування та передача даних через API: зібрані дані конвертуються у формат JSON та надсилаються на серверну частину системи через API;

5) збереження даних у базі: серверна частина приймає дані, перевіряє їх коректність та зберігає у базі даних Microsoft SQL Server;

г) переваги;

1) зменшення ручної праці та ризику помилок;

2) швидкий та ефективний збір великої кількості даних з численних серверів.

#### 4.1.2 Сценарій 2: Перегляд звітів про використання дискового простору

а) учасники: керівник відділу;

б) мета: аналіз використання дискового простору для планування ресурсів та бюджетування;

в) кроки;

1) авторизація у веб-інтерфейсі: керівник відділу входить у систему, використовуючи свої облікові дані;

2) вибір періоду та відділу: у розділі "звіти" обирається потрібний період та відділ для аналізу;

3) перегляд звітів: система відображає детальні звіти у вигляді таблиць та діаграм (стовпчикові та кругові);

4) аналіз трендів та аномалій: керівник аналізує дані про використання дискового простору, виявляє тенденції зростання або зменшення обсягів даних та визначає потенційні аномалії;

5) прийняття рішень: на основі аналізу даних плануються необхідні дії, такі як очищення дисків, архівування даних або збільшення дискового простору;

г) переваги;

1) зручний доступ до детальних звітів дозволяє керівникам приймати обґрунтовані рішення;

2) візуалізація даних полегшує розуміння тенденцій та аномалій у використанні дискового простору.

#### 4.1.3 Сценарій 3: Планування бюджету на зберігання даних

а) учасники: фінансовий аналітик;

б) мета: планування бюджету на зберігання даних на основі поточного використання дискового простору та прогнозованого зростання;

в) кроки;

1) авторизація у веб-інтерфейсі: фінансовий аналітик входить у систему, використовуючи свої облікові дані;

2) отримання поточних даних: аналітик переходить до розділу "бюджетування" та вибирає опцію "прогнозування бюджету";

3) прогнозування зростання: система надає прогнозоване зростання обсягу даних на основі історичних даних та введених параметрів;

4) розрахунок бюджету: використовуючи формули розділу 3, система автоматично розраховує прогнозований бюджет на зберігання даних;

5) формування звіту: система генерує детальний звіт з розрахунками бюджету, який можна експортувати у форматах PDF або Excel для подальшого аналізу;

г) переваги;

1) автоматичний розрахунок бюджету знижує ймовірність помилок та забезпечує точність фінансових планів;

2) інтеграція з системою дозволяє враховувати актуальні дані та забезпечує актуальність прогнозів.

#### 4.1.4 Сценарій 4: Отримання сповіщень про аномалії у використанні дискового простору

а) учасники: адміністратор іт-системи;

б) мета: автоматичне виявлення та повідомлення про аномалії у використанні дискового простору для оперативного реагування;

в) кроки;

1) моніторинг системи: система постійно моніторить використання дискового простору на серверах з встановленими пороговими значеннями (наприклад, 80% використання диска);

2) виявлення аномалії: якщо обсяг використаного дискового простору перевищує встановлені пороги, система автоматично ідентифікує це як аномалію;

3) формування сповіщення: система формує сповіщення з деталями аномалії, включаючи інформацію про сервер, відділ, тип дискового простору та час виявлення;

4) надсилання сповіщення: сповіщення надсилається адміністратору через електронну пошту або інший налаштований канал комунікації;

5) реагування на сповіщення: адміністратор отримує сповіщення та аналізує його, виконує необхідні дії (очищення диска, збільшення дискового простору або проведення додаткового аналізу для виявлення причини аномалії);

г) переваги:

1) оперативне виявлення проблем дозволяє мінімізувати ризики перевантаження дисків та запобігти потенційним збоєм системи;

2) автоматизація процесу сповіщення забезпечує своєчасну реакцію адміністратора, знижуючи потребу в постійному ручному моніторингу;

#### 4.1.5 Сценарій 5: Запит інформації про використання дискового простору

а) учасники: користувач (співробітник відділу);

б) мета: надання користувачам можливості переглядати інформацію про використання дискового простору у своєму відділі для покращення організації роботи з даними;

в) кроки:

1) авторизація: користувач входить у систему через веб-інтерфейс, використовуючи свої облікові дані з урахуванням ролі та прав доступу;

2) перехід до розділу "мій відділ": користувач обирає свій відділ у меню навігації та переходить до розділу "мій відділ";

3) перегляд інформації: система відображає детальну інформацію про використання дискового простору у вигляді таблиць та діаграм, включаючи розміри папок, кількість файлів та тренди використання за попередній період;

4) аналіз даних: користувач аналізує інформацію для виявлення можливостей для оптимізації роботи з даними, таких як архівування старих файлів або видалення непотрібних даних;

5) завантаження звіту: користувач може завантажити звіт у форматі pdf або excel для подальшого аналізу або обговорення з колегами;

г) переваги;

1) надання користувачам доступу до актуальної інформації дозволяє їм ефективніше управляти своїми файловими ресурсами;

2) зручність перегляду та аналізу даних сприяє покращенню організації роботи з даними та підвищенню продуктивності.

Представлені сценарії використання системи демонструють різні аспекти взаємодії користувачів із автоматизованою системою аналізу файлових сховищ. Вони охоплюють ролі адміністраторів, керівників відділів, фінансових аналітиків та звичайних користувачів, кожен з яких має свої унікальні потреби та цілі. Діаграму сценаріїв можна побачити в Додатку Д.

Завдяки таким сценаріям система стає інструментом, який не лише автоматизує рутинні процеси, але й надає користувачам потужні засоби для прийняття обґрунтованих рішень щодо оптимізації використання ІТ-ресурсів.[8]

#### 4.2 Моделювання та проектування системи

Моделювання та проектування системи є ключовим етапом у розробці автоматизованої системи аналізу файлових сховищ для середовища Windows Server. На цьому етапі визначаються архітектурні рішення, вибір технологій та структурна організація компонентів, що забезпечують ефективність, масштабованість та надійність системи. Детальне проектування дозволяє передбачити можливі проблеми, оптимізувати роботу системи та забезпечити її відповідність вимогам користувачів.[9]

### 4.2.1 Архітектура

Архітектура системи базується на багаторівневому підході, що дозволяє розділити функціональність на логічні рівні та забезпечити модульність та гнучкість системи. Основними компонентами архітектури є:

Клієнтська частина (Front-end): відповідає за взаємодію з користувачем, надаючи зручний та інтуїтивно зрозумілий інтерфейс для доступу до функціональності системи. Реалізована у вигляді веб-застосунку, що працює у браузері та забезпечує відображення даних, отриманих від серверної частини.

Серверна частина (Back-end): виконує роль посередника між клієнтською частиною та базою даних. Вона обробляє запити від клієнтської частини, виконує бізнес-логіку, взаємодіє з базою даних для зберігання та отримання інформації, а також забезпечує безпеку та контроль доступу до даних. Діаграму класів представлено в Додатку Е.[10]

База даних: використовується для надійного зберігання структурованих даних про файлові сховища, результати аналізу, користувачів та інші сутності системи. Вона забезпечує швидкий та ефективний доступ до інформації, необхідної для роботи системи.

Служби збору даних: реалізовані у вигляді PowerShell-скриптів, що виконуються на серверах Windows Server. Вони автоматизують процес збору інформації про файлові системи, дискові простори та інші параметри, необхідні для аналізу та прийняття управлінських рішень.

### 4.2.2 Серверна частина

Серверна частина є центральним компонентом системи, що відповідає за обробку запитів від клієнтів, виконання бізнес-логіки та взаємодію з базою даних. Вона реалізована на платформі .NET Core з використанням мови програмування С#, що забезпечує високу продуктивність, кросплатформеність та підтримку сучасних технологій.[11]

### Основні функції серверної частини:

- обробка HTTP-запитів: Серверна частина приймає запити від клієнтської частини через протокол HTTP/HTTPS, обробляє їх та повертає відповідні відповіді.
- виконання бізнес-логіки: Реалізує основні алгоритми та правила роботи системи, включаючи аналіз даних, виявлення аномалій, генерацію звітів тощо.
- взаємодія з базою даних: Здійснює зберігання та отримання даних з бази даних за допомогою Entity Framework Core. Це спрощує роботу з базою даних та забезпечує ефективне виконання запитів.
- безпека та автентифікація: Реалізує механізми автентифікації користувачів (наприклад, за допомогою JWT-токенів), авторизації доступу до різних ресурсів системи та захисту від поширених атак (SQL-ін'єкції, XSS, CSRF).
- логування та моніторинг: Записує важливі події та помилки для подальшого аналізу та моніторингу роботи системи.

Серверна частина має багатoshарову архітектуру та розділена на такі шари:

- контролери (Controllers): Обробляють HTTP-запити, взаємодіють з сервісами та формують відповіді.
- сервіси (Services): Містять бізнес-логіку та здійснюють основні операції над даними.
- репозиторії (Repositories): Відповідають за взаємодію з базою даних, забезпечуючи абстракцію доступу до даних.
- моделі (Models): Визначають структуру даних, які використовуються в системі.

### Архітектурні особливості серверної частини:

- використання Dependency Injection: Забезпечує гнучкість та легкість тестування шляхом інжекції залежностей між компонентами.[12]
- кешування: Реалізовано механізми кешування даних на рівні серверної частини для підвищення продуктивності та зменшення навантаження на базу даних.

- підтримка асинхронності: Використання асинхронних методів для обробки запитів дозволяє підвищити продуктивність та масштабованість системи.

Взаємодія з базою даних:

- Entity Framework Core: ORM-фреймворк, що забезпечує роботу з базою даних на рівні об'єктів. Це спрощує розробку та підтримку коду, зменшуючи кількість SQL-запитів.

- міграції бази даних: Використання механізму міграцій для автоматичного оновлення структури бази даних відповідно до змін у моделях даних.

- оптимізація запитів: Використання індексів, оптимізованих запитів та інших засобів для забезпечення швидкого доступу до даних.

- безпека серверної частини:

- аутентифікація та авторизація: Використання сучасних методів аутентифікації та ролей для контролю доступу.

- валідація даних: Перевірка вхідних даних для запобігання атакам та помилкам.

- SSL/TLS: Забезпечення шифрування даних при передачі між клієнтом та сервером.

#### 4.2.3 Клієнтська частина

Клієнтська частина системи реалізована у вигляді веб-інтерфейсу, що надає користувачам доступ до функціональності системи через браузер. Вона відповідає за відображення інформації, отриманої від серверної частини, та забезпечує зручний інтерфейс для взаємодії з користувачем.

Для розробки клієнтської частини використовується технологія ASP.NET Core MVC у поєднанні з HTML, CSS та JavaScript. Це дозволяє створювати динамічні та інтерактивні веб-сторінки, що адаптуються до різних пристроїв та роздільних здатностей екранів.

Клієнтська частина реалізує механізми аутентифікації та авторизації, забезпечує можливість перегляду та фільтрації даних, візуалізації інформації у вигляді графіків та діаграм з використанням бібліотеки Chart.js. Інтуїтивно зрозумілий інтерфейс та сучасний дизайн сприяють зручності використання системи та підвищують продуктивність роботи користувачів.[13]

Взаємодія з серверною частиною здійснюється за допомогою HTTP-запитів у форматі JSON, що забезпечує швидкий та ефективний обмін даними. Клієнтська частина також реалізує механізми обробки помилок та повідомлень користувачу, що покращує загальний досвід роботи з системою.

### 4.3 Вибір технологій

Вибір технологій для розробки системи є критичним фактором, що впливає на її продуктивність, масштабованість, безпеку та легкість підтримки. У цьому підрозділі обґрунтовується вибір мови програмування, фреймворків та середовища розробки, які використовуються для реалізації автоматизованої системи аналізу файлових сховищ.

#### 4.3.1 Мова програмування

Для розробки серверної та клієнтської частин системи обрана мова програмування C#. Цей вибір обумовлений кількома факторами:

- висока продуктивність та ефективність: C# є сучасною об'єктно-орієнтованою мовою, яка забезпечує високу швидкість виконання коду та оптимізацію ресурсів.
- кросплатформеність: Завдяки платформі .NET Core, C# дозволяє розробляти застосунки, що можуть виконуватися на різних операційних системах, включаючи Windows, Linux та macOS.
- розвинена екосистема: Велика кількість бібліотек та фреймворків полегшує розробку складних систем, надаючи готові рішення для різних задач.

– підтримка спільноти: Активна спільнота розробників та велика кількість навчальних матеріалів спрощують процес навчання та вирішення проблем. [14]

#### 4.3.2 Фреймворки

Для реалізації серверної частини системи використовується фреймворк ASP.NET Core. Він надає потужні інструменти для створення веб-застосунків та API, забезпечуючи високу продуктивність та безпеку. ASP.NET Core підтримує модульну архітектуру, що дозволяє легко додавати або змінювати функціональність системи. [15]

Для роботи з базою даних обрано Entity Framework Core, який є ORM-фреймворком для .NET. Він спрощує взаємодію з базами даних, дозволяючи працювати з ними на рівні об'єктів. Це зменшує кількість коду, необхідного для виконання операцій з даними, та знижує ризик помилок. Схема розгортання представлена в Додатку Ж.[16]

У клієнтській частині використовується ASP.NET Core MVC для реалізації моделі MVC (Model-View-Controller), що розділяє логіку застосунку на три компоненти, спрощуючи розробку та підтримку. Для візуалізації даних застосовується бібліотека Chart.js, яка дозволяє створювати інтерактивні графіки та діаграми на основі JavaScript.[17]

Для створення адаптивного дизайну використовується Bootstrap, популярний CSS-фреймворк, який забезпечує зручні інструменти для розробки інтерфейсу користувача, що коректно відображається на різних пристроях та екранах.[18]

#### 4.3.3 Середовище розробки

Основним середовищем розробки обрано Microsoft Visual Studio, яке надає потужні інструменти для розробки .NET-застосунків. Visual Studio підтримує інтеграцію з системами контролю версій, такими як Git, має розширені можливості

для налагодження та тестування коду, а також широкий набір плагінів та розширень.

Використання Visual Studio дозволяє підвищити продуктивність розробників завдяки таким функціям, як інтелектуальне автозаповнення коду (IntelliSense), інтегрований відладчик, візуальні дизайнери та багато іншого.

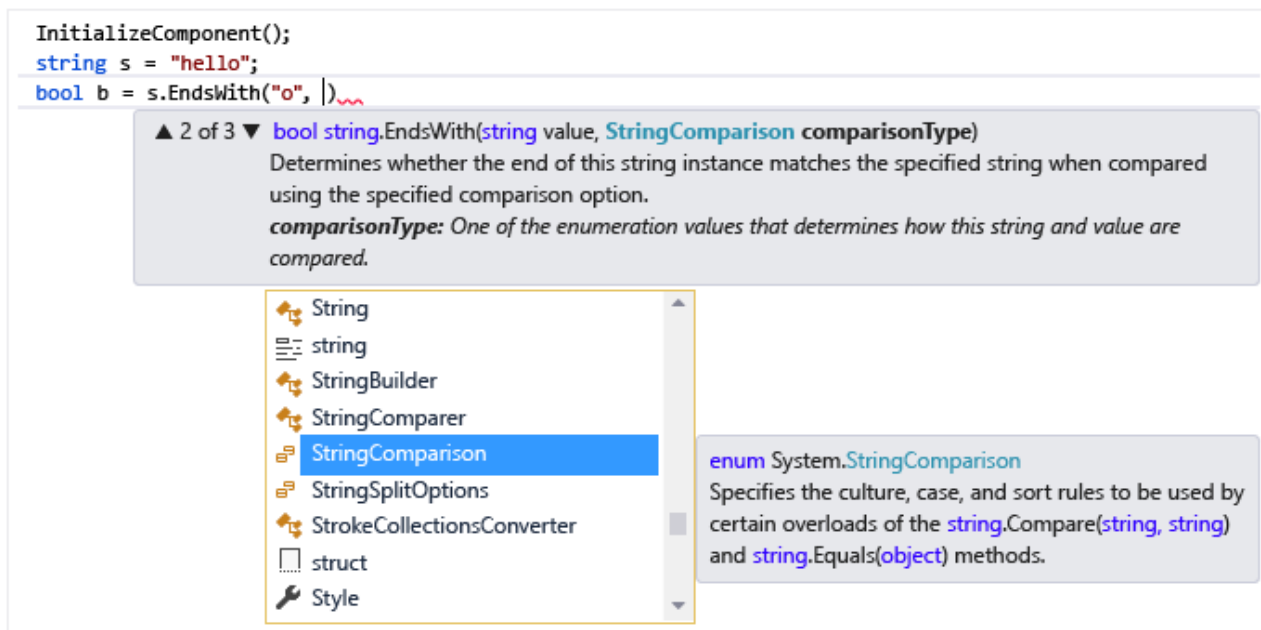


Рисунок 4.1 – Приклад функцій IntelliSense в середовищі Visual Studio 2022

#### Висновок до розділу 4

В розділі 4 було розглянуто основні сценарії використання автоматизованої системи аналізу файлових сховищ, які охоплюють різні етапи взаємодії користувачів із системою. Це включає автоматизований збір даних, моніторинг використання дискового простору, планування бюджету та виявлення аномалій. Така система дозволяє зменшити ручну працю, забезпечує своєчасне прийняття рішень та підвищує ефективність управління ІТ-ресурсами.

Архітектура системи базується на багаторівневому підході, де кожен компонент виконує свою чітко визначену роль. Клієнтська частина надає зручний інтерфейс для користувачів, а серверна частина реалізує обробку запитів і взаємодіє з базою даних для зберігання та отримання даних. Така структура дозволяє системі бути гнучкою та масштабованою, що є важливим для ефективного

управління великими обсягами даних. З діаграмою потоків даних можна ознайомитись в Додатку К.

Вибір мови програмування C#, фреймворків ASP.NET Core та Entity Framework Core, а також середовища розробки Microsoft Visual Studio є оптимальним рішенням для реалізації автоматизованої системи аналізу файлових сховищ. Ці технології забезпечують високу продуктивність, безпеку та масштабованість системи, а також спрощують процес розробки та підтримки.

Використання сучасних інструментів та підходів дозволяє створити надійне та ефективне рішення, що відповідає вимогам сучасних організацій щодо управління ІТ-ресурсами.

## 5 ПРОЄКТУВАННЯ БАЗИ ДАНИХ

### 5.1 Структура бази даних

Проєктування бази даних є критичним етапом у розробці автоматизованої системи аналізу файлових сховищ, оскільки від її ефективності залежить швидкість та надійність доступу до даних. База даних повинна забезпечувати зберігання великого обсягу інформації про файлові системи, використання дискового простору, користувачів та інші сутності, пов'язані з роботою системи.[19]

Вибір типу бази даних зупинився на реляційній базі даних Microsoft SQL Server. Цей вибір обумовлений наступними причинами:

- продуктивність та масштабованість: SQL Server здатний ефективно обробляти великі обсяги даних та підтримує масштабування при зростанні навантаження.
- інтеграція з .NET: Тісна інтеграція з платформою .NET та Entity Framework Core спрощує розробку та взаємодію з базою даних.
- безпека: SQL Server надає розширені можливості для забезпечення безпеки даних, включаючи аутентифікацію, шифрування та контроль доступу.
- інструменти адміністрування: Наявність зручних інструментів для управління та моніторингу бази даних.

Моделювання бази даних здійснювалося на основі аналізу вимог системи та включає наступні основні сутності:

- Servers (Сервери) — зберігає інформацію про сервери, з яких збираються дані.
- Folders (Папки) — містить інформацію про папки на серверах, їх розміри та структуру.
- Departments (Відділи) — відображає організаційну структуру підприємства.
- Users (Користувачі) — облікові записи користувачів системи, їх ролі та права доступу.

- DiskUsages (Використання дисків) — історичні дані про використання дискового простору.

- Notifications (Сповіщення) — інформація про сповіщення щодо аномалій або перевищення порогових значень.

- Settings (Налаштування) — конфігураційні параметри системи.

Діаграма сутність-зв'язок (ER-діаграма) відображає взаємозв'язки між таблицями бази даних, наведено у Додатку Г.

Основні взаємозв'язки між таблицями:

- один сервер може мати багато папок (зв'язок один-до-багатьох між Servers та Folders).

- папка може належати до певного відділу (зв'язок багато-до-одного між Folders та Departments).

- користувачі пов'язані з відділами (зв'язок багато-до-одного між Users та Departments).

- користувачі можуть отримувати сповіщення (зв'язок один-до-багатьох між Users та Notifications).

- дані про використання дисків пов'язані з папками та серверами (зв'язки між DiskUsages, Folders та Servers).

## 5.2 Опис таблиць бази даних

У цьому підрозділі детально описуються кожна з таблиць бази даних, їхні поля, типи даних та призначення.

### 5.2.1 Таблиця Servers (Сервери)

Поле	Тип даних	Опис
ServerID	int, PK	Унікальний ідентифікатор сервера
Name	nvarchar(100)	Ім'я сервера

Поле	Тип даних	Опис
IPAddress	nvarchar(50)	IP-адреса сервера
Location	nvarchar(100)	Розташування сервера (опціонально).
Description	nvarchar(255)	Опис або додаткова інформація про сервер

Призначення: зберігає інформацію про сервери, з яких збираються дані.

### 5.2.2 Таблиця Folders (Папки)

Поле	Тип даних	Опис
FolderID	int, PK	Унікальний ідентифікатор папки.
ServerID	int, FK	Ідентифікатор сервера, на якому знаходиться папка.
DepartmentID	int, PK	Ідентифікатор відділу, якому належить папка.
Path	nvarchar(500)	Повний шлях до папки.
Size	bigint	Розмір папки у байтах.
CreatedDate	datetime	Дата створення запису.

Призначення: містить інформацію про папки на серверах, їх розміри та структуру.

Зв'язки:

- зв'язок з таблицею Servers по полю ServerID.
- зв'язок з таблицею Departments по полю DepartmentID.

## 5.2.3 Таблиця Departments (Відділи)

Поле	Тип даних	Опис
DepartmentID	int, PK	Унікальний ідентифікатор відділу.
Name	nvarchar(100)	Назва відділу.
ManagerID	int, FK	Ідентифікатор керівника відділу (посилання на таблицю Users).
Description	nvarchar(255)	Опис відділу.

Призначення: зберігає інформацію про відділи підприємства.

Зв'язки:

- зв'язок з таблицею Users по полю ManagerID.

## 5.2.4 Таблиця Users (Користувачі)

Поле	Тип даних	Опис
UserID	int, PK	Унікальний ідентифікатор користувача.
DepartmentID	int, FK	Ідентифікатор відділу, до якого належить користувач.
Username	nvarchar(50)	Ім'я користувача (логін).
PasswordHash	nvarchar(255)	Хеш пароля.
Email	nvarchar(100)	Електронна пошта користувача.

Поле	Тип даних	Опис
Role	nvarchar(50)	Роль користувача (Admin, Manager, User).
IsActive	bit	Статус активності користувача.

Призначення: містить облікові записи користувачів системи, їхні ролі та права доступу.

Зв'язки:

- зв'язок з таблицею Departments по полю DepartmentID.

#### 5.2.5 Таблиця DiskUsages (Використання дисків)

Поле	Тип даних	Опис
DiskUsageID	int, PK	Унікальний ідентифікатор запису.
ServerID	int, FK	Ідентифікатор сервера.
FolderID	int, FK	Ідентифікатор папки.
CollectedDate	datetime	Дата та час збору даних.
UsedSpace	bigint	Обсяг використаного дискового простору у байтах.
FreeSpace	bigint	Обсяг вільного дискового простору у байтах.
TotalSpace	bigint	Загальний обсяг дискового простору у байтах.

Призначення: зберігає історичні дані про використання дискового простору.

Зв'язки:

- зв'язок з таблицею Servers по полю ServerID.
- зв'язок з таблицею Folders по полю FolderID.

### 5.2.6 Таблиця Notifications (Сповіщення)

Поле	Тип даних	Опис
NotificationID	int, PK	Унікальний ідентифікатор сповіщення.
UserID	int, FK	Ідентифікатор користувача, який отримав сповіщення.
Message	nvarchar(500)	Текст сповіщення.
CreatedDate	datetime	Дата та час створення сповіщення.

Призначення: містить інформацію про сповіщення щодо аномалій або перевищення порогових значень.

Зв'язки:

- зв'язок з таблицею Users по полю UserID.

### Висновки до розділу 5

У цьому розділі було детально розглянуто проєктування бази даних для автоматизованої системи аналізу файлових сховищ. Структура бази даних розроблена таким чином, щоб забезпечити ефективне зберігання та швидкий доступ до інформації, необхідної для роботи системи. Визначені основні таблиці та їхні взаємозв'язки, що дозволяє підтримувати цілісність даних та забезпечувати ефективну взаємодію між компонентами системи.[20]

Ретельно спроектована база даних є фундаментом для надійної та масштабованої системи, що відповідає вимогам щодо продуктивності та безпеки. Вона дозволяє здійснювати складні запити та аналіз даних, необхідних для прийняття обґрунтованих рішень щодо управління дисковими ресурсами в організації. Таким чином, проектування бази даних є важливим кроком на шляху до створення ефективної та надійної системи аналізу файлових сховищ.

## 6 ІНТЕРФЕЙС КОРИСТУВАЧА ТА UX/UI ДИЗАЙН

У сучасних програмних системах інтерфейс користувача відіграє критично важливу роль, оскільки від його зручності та інтуїтивності залежить ефективність роботи користувачів із системою. У цьому розділі розглядаються принципи проектування інтерфейсу для автоматизованої системи аналізу файлових сховищ, а також аспекти адаптивності та кросплатформенності, що забезпечують доступність системи на різних пристроях та платформах.[21]

### 6.1 Принципи проектування інтерфейсу

При розробці інтерфейсу користувача для системи аналізу файлових сховищ було враховано основні принципи UX/UI дизайну, що сприяють створенню зручного та ефективного інтерфейсу:

#### 6.1.1 Простота та мінімалізм

Принципи простоти та мінімалізму є основними в розробці інтерфейсу, оскільки вони дозволяють створити зручний та ефективний досвід для користувача. У розробленій системі ці принципи реалізовано через чистий, не перенасичений елементами інтерфейс, що дозволяє користувачам швидко орієнтуватися на головній сторінці.

На головній сторінці інтерфейсу (рисунок 6.1) програми відсутні зайві елементи або складні анімації, що можуть відволікати увагу користувача. Основна увага приділяється важливим даним про використання дискового простору, таким як графік та статистика, що дозволяє користувачеві без зайвих кроків оцінити поточний стан системи.[22]

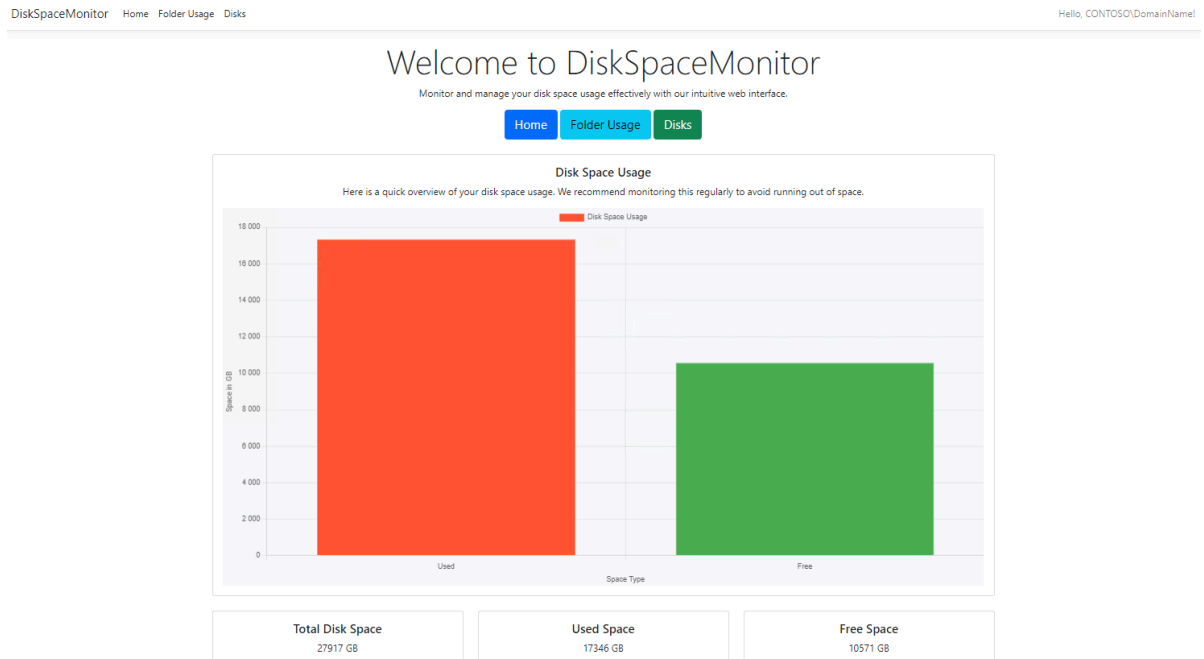


Рисунок 6.1 – Головна сторінка системи

Головна сторінка є центральним елементом інтерфейсу, що надає користувачеві швидкий доступ до найбільш важливих функцій додатку. Вона містить кілька основних елементів:

- заголовок та вступне повідомлення: У верхній частині сторінки розташовано заголовок "Welcome to DiskSpaceMonitor", що чітко вказує на призначення додатку. Під заголовком розміщено короткий опис функціоналу програми, який допомагає користувачеві швидко зрозуміти, що саме він може моніторити через цю систему.

- навігаційні кнопки: Проста та інтуїтивно зрозуміла навігація на сторінці представлена кількома великими кнопками, які дозволяють користувачеві швидко переміщатися між основними розділами програми, такими як "Home", "Folder Usage" і "Disks". Кнопки мають контрастні кольори та достатній розмір, що робить їх помітними та доступними для взаємодії.

- графік використання дискового простору: Один із основних елементів інтерфейсу — це графік, що надає візуалізацію використаного та вільного дискового простору. Столбчаста діаграма надає користувачеві наглядну інформацію про поточний стан дискового простору. Діаграма виконана в

мінімалістичному стилі: без зайвих ліній та елементів, що дозволяє швидко сприймати основну інформацію.

– карточки статистики: Під графіком розташовані три карточки статистики, що відображають загальний обсяг дискового простору, використане та вільне місце. Карточки мають чітку структуру з назвами та значеннями в гігабайтах, що забезпечує легкість у сприйнятті інформації.

### 6.1.2 Інтуїтивна навігація

Інтуїтивна навігація є важливим аспектом у проектуванні інтерфейсу користувача, оскільки вона дозволяє легко та швидко переміщатися між різними розділами програми без необхідності зайвих пошуків. Система реалізує просту та зрозумілу навігацію через кнопки на головній сторінці (рисунок 6.2), що сприяє зручному користуванню та ефективному доступу до важливої інформації.[23]

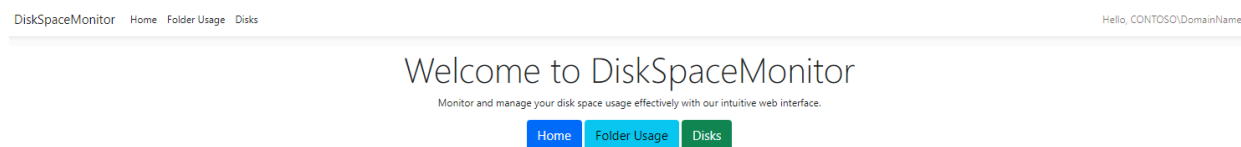


Рисунок 6.2 – Кнопки навігації системи

– головна сторінка (Home): Кнопка "Home" є основним елементом для повернення на стартову сторінку програми. Вона дозволяє користувачу швидко повернутися до головної інформації, якщо він знаходиться в іншому розділі. Це допомагає підтримувати зручність користування та дає змогу легко перейти назад до основних даних, таких як графік та статистика.

– використання папок (Folder Usage): Кнопка "Folder Usage" забезпечує доступ до розділу, де користувач може побачити більш детальну інформацію про використання дискового простору конкретними папками. Ця кнопка дозволяє користувачу зручно перемикатися між різними типами даних, зберігаючи при цьому зрозумілу та логічну навігацію по додатку.

– диски (Disks): Кнопка "Disks" призначена для переходу в розділ, де відображається інформація про стан дисків, включаючи їх загальний обсяг, використане та вільне місце. Це дозволяє користувачам отримати більш детальну картину щодо дискового простору на різних серверах і пристроях, що відображаються в додатку.

Таким чином, інтуїтивна навігація через кнопки на головній сторінці дозволяє користувачам без зусиль переміщатися між різними розділами програми, швидко знаходити потрібну інформацію та ефективно працювати з даними без зайвих зусиль і складнощів.

### 6.1.3 Візуалізація даних про дисковий простір

Система надає користувачу графічне зображення використаного та вільного дискового простору через столбчасту діаграму (рисунок 6.3). Користувачі можуть швидко оцінити стан всіх доступних дисків, переглядаючи графіки, які наочно показують, скільки місця використано і скільки залишилося. Користувач може відображати показники для кожного окремого сервера, використовуючи фільтри (рисунок 6.4) Це дозволяє легко виявляти проблеми з дисковим простором і приймати оперативні рішення.

## Disk Usage (Used and Free Space)



Рисунок 6.3 - Столпчикова діаграма для моніторингу використаного та вільного простору на дисках

Select Server

Filter by Server Group

Disk Name	Total Space (GB)	Free Space (GB)	Used Space (GB)	Server
Disk 1	0	0	0	Server-02

## Disk Usage (Used and Free Space)

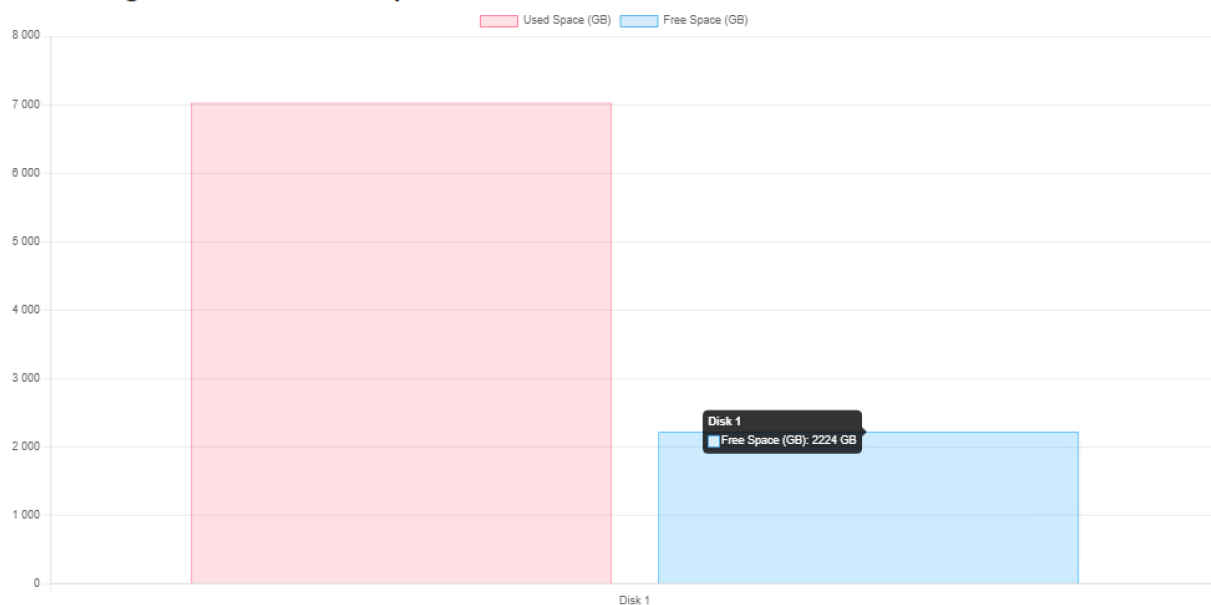
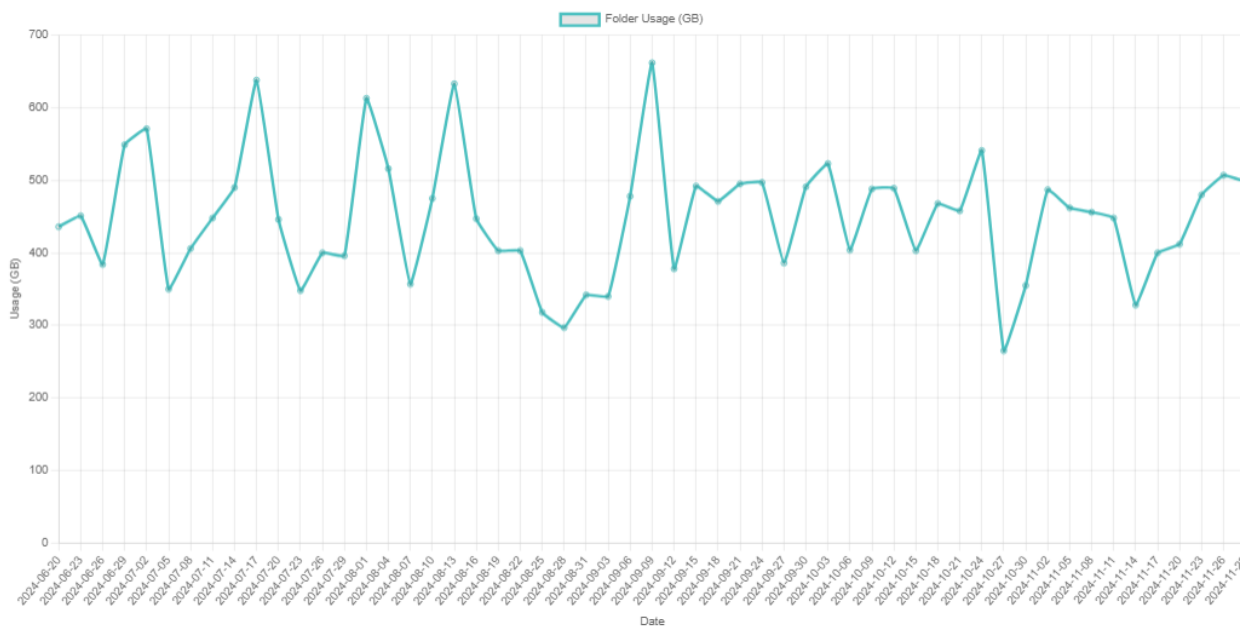


Рисунок 6.4 – Приклад фільтрації та побудови графіків по вибраному серверу

#### 6.1.4 Аналіз використання папок

У системі також передбачена можливість переглядати статистику по конкретних папках, що дозволяє визначити, які саме каталоги займають найбільше місця на диску (рисунок 6.5). Користувач може за допомогою фільтрів проаналізувати файлове сховище відповідно до компанії, відділу, сектору та служби (рисунок 6.6). Ця функція є важливою для користувачів, які хочуть здійснювати більш детальний аналіз і оптимізувати використання дискового простору.

## Folder Usage Over Time



## Folder Usage by Company

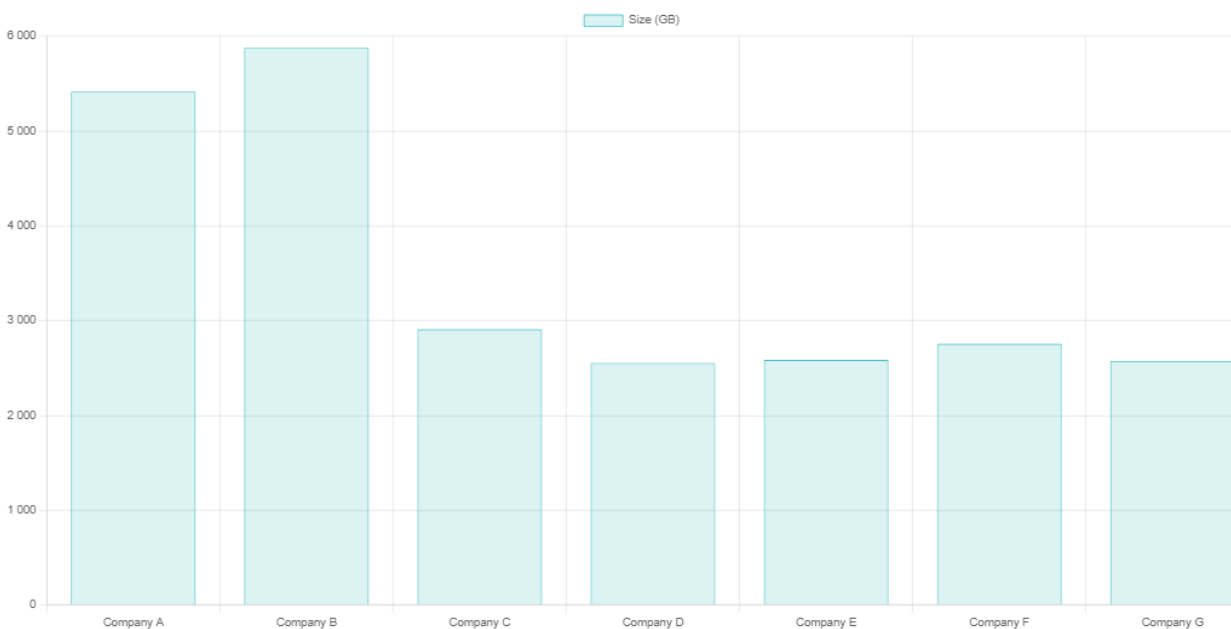
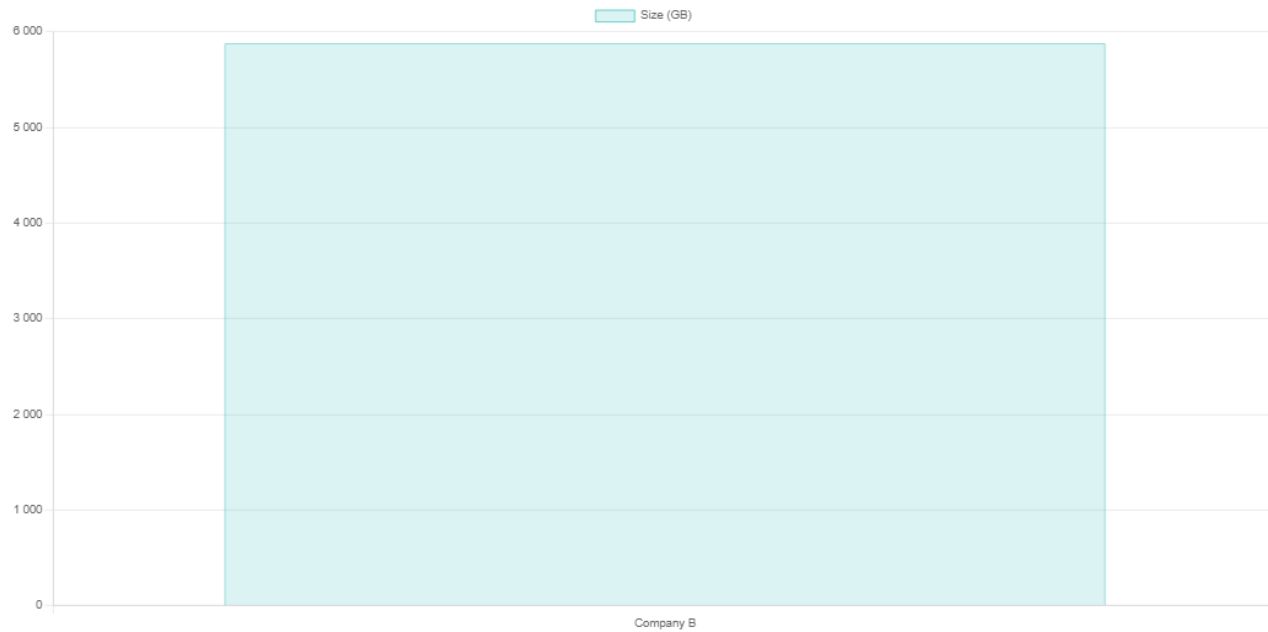


Рисунок 6.5 – Графіки та діаграми використання простору по папках

Folder Usage by Company



Folder Usage by Service



Рисунок 6.5 - Графіки та діаграми використання простору по папках, застосувавши фільтр по компанії

### 6.1.5 Фільтрація та сортування даних

Система також дозволяє фільтрувати та сортувати дані за різними критеріями (наприклад, за сервером, компанією, відділом, сектором чи службою), що робить взаємодію з додатком ще зручнішою (рисунок 6.6). Користувачі можуть вибрати

лише ті сервери або департаменти, які їм необхідно аналізувати, що оптимізує роботу з великими обсягами даних (рисунок 6.7).

## Folder Usage

Filter By:  
Company

Filter Value:  
Company B

Select Time Period:  
-- Select Time Period --  
-- Select Time Period --  
Last 7 Days  
Last 30 Days  
Last 6 Months

Рисунок 6.6 – Фільтр даних по структурі компанії

## Disks

Select Server  
All Servers

All Servers
Server-01
Server-02
Server-03
Server-04
Server-05
Server-06
Server-07
Server-08
Server-09
Server-10

Рисунок 6.7 – Фільтр даних по серверам

### 6.1.6 Можливість персоналізації

Система надає можливість налаштувати інтерфейс відповідно до уподобань користувача.

– налаштування відображення даних: Користувач може вибирати, які дані та у якому форматі відображати.

## 6.2 Адаптивність та кросплатформенність

Сучасні користувачі очікують, що програмні системи будуть доступні на різних пристроях та платформах. У зв'язку з цим при розробці інтерфейсу особлива увага приділялася адаптивності та кросплатформенності.

### 6.2.1 Адаптивний дизайн

Адаптивний дизайн забезпечує коректне відображення інтерфейсу на різних розмірах екранів та пристроях. Це означає, що незалежно від того, чи використовує користувач великий десктопний монітор, планшет або смартфон, інтерфейс програми буде відображатися чітко, функціонально та зручно для користування, приклади наведені на рисунках 6.8, 6.9 та 6.10.

а) використання гнучких макетів;

1) графіки, які використовуються в інтерфейсі, побудовані на основі гнучких макетів. це забезпечує автоматичне підлаштування розмірів графіків та їх компонентів до розміру вікна браузера користувача. зокрема, розміри графіків змінюються, а маркування осей, підписи та інші елементи оновлюються, щоб залишатися читабельними та зрозумілими;

б) медіа-запити css;

1) для забезпечення адаптивності графіків також використовуються медіа-запити css, що дозволяє змінювати стилі залежно від розміру екрану та щільності пікселів. наприклад, на малих екранах текст маркувань може зменшуватися, графіки можуть відображатися вертикально, а не горизонтально, щоб зберегти зручність використання.

Адаптивність забезпечує можливість зручного перегляду даних на різних пристроях: на малих екранах елементи графіку, такі як підписи, змінюють розмір або приховуються для збереження чистоти інтерфейсу. [24]

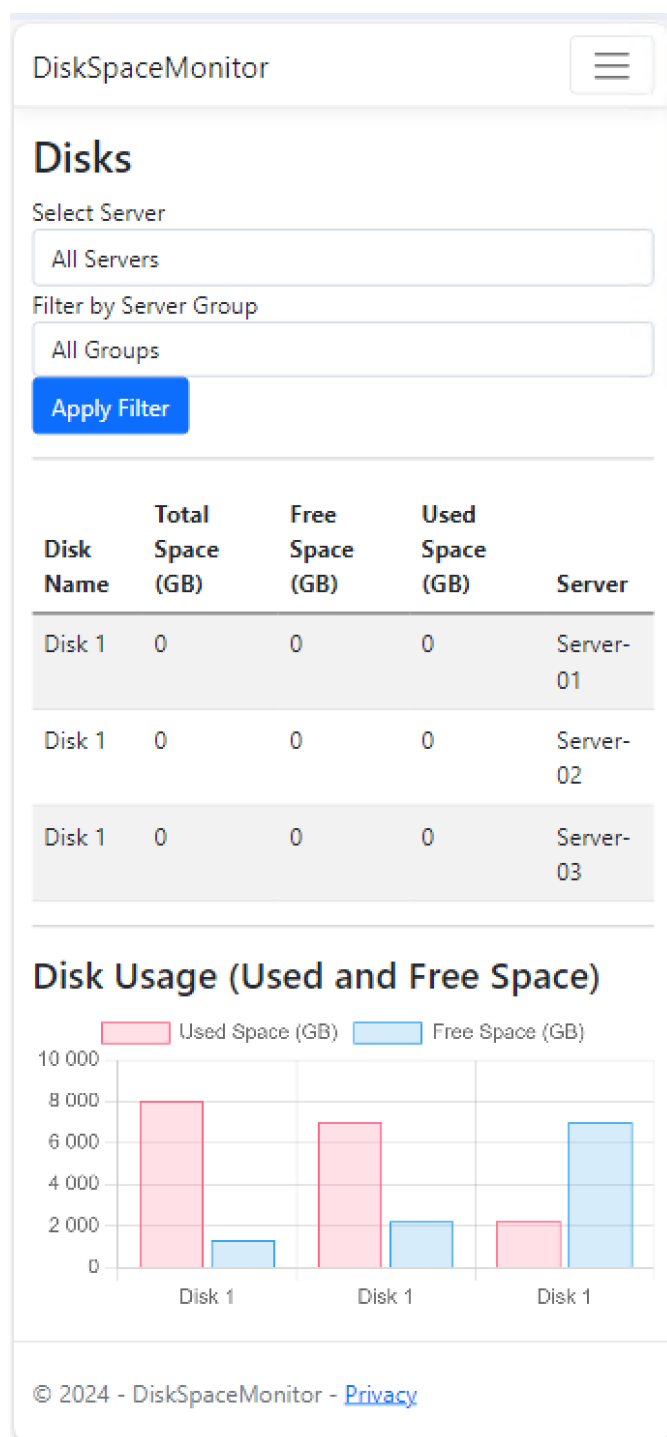


Рисунок 6.8 – Інтерфейс системи на пристрої Apple iPhone 12 Pro Max

Disk Name	Total Space (GB)	Free Space (GB)	Used Space (GB)	Server
Disk 1	0	0	0	Server-01
Disk 1	0	0	0	Server-02
Disk 1	0	0	0	Server-03

### Disk Usage (Used and Free Space)

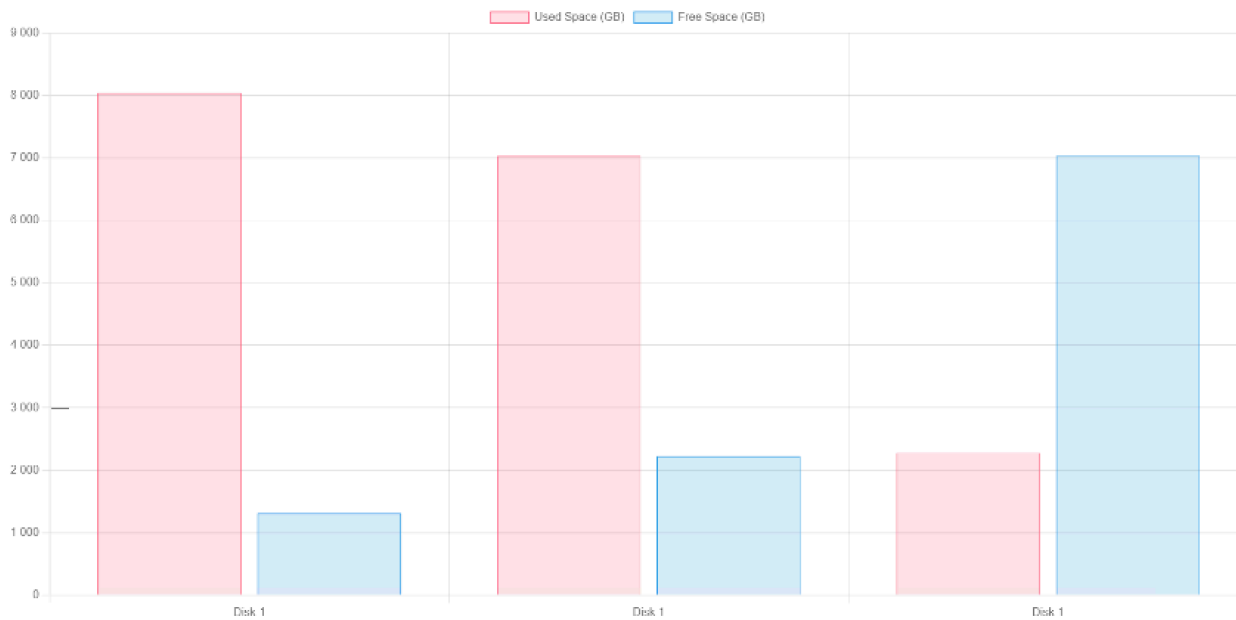


Рисунок 6.9 – Інтерфейс системи на пристрої Asus Zenbook Fold

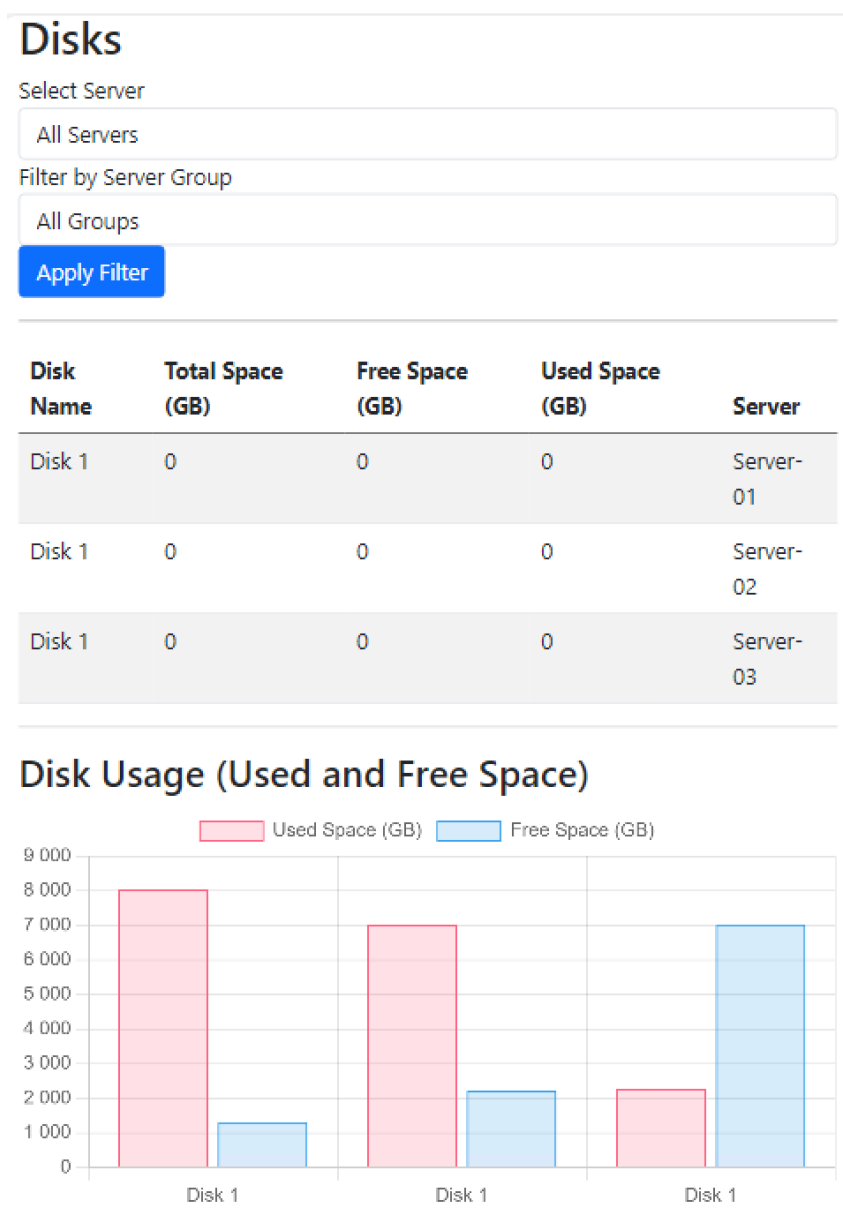


Рисунок 6.10 – Інтерфейс системи на пристрої Apple iPad Air

## 6.2.2 Кросплатформенність

Система розроблена з урахуванням можливості роботи на різних операційних системах та браузерях.

- підтримка основних браузерів: Інтерфейс тестувався у сучасних браузерах, таких як Google Chrome, Mozilla Firefox, Microsoft Edge та Safari.
- використання стандартних веб-технологій: HTML5, CSS3 та JavaScript забезпечують сумісність з різними платформами без потреби у додаткових плагінах.

### 6.2.3 Тестування та оптимізація

Для забезпечення високої якості інтерфейсу було проведено ретельне тестування на різних пристроях та у різних умовах.

- кросбраузерне тестування: виявлення та виправлення проблем сумісності у різних браузерах.

- оптимізація швидкості завантаження: зменшення розміру ресурсів та використання кешування для швидкого завантаження інтерфейсу.

### Висновки до розділу 6

У цьому розділі було розглянуто основні принципи проектування інтерфейсу користувача для автоматизованої системи аналізу файлових сховищ. Дотримання принципів простоти, інтуїтивності та консистентності дозволило створити зручний та ефективний інтерфейс, який відповідає потребам користувачів різних категорій. Особлива увага приділялася адаптивності та кросплатформенності, що забезпечує доступність системи на різних пристроях та платформах.

Завдяки продуманому UX/UI дизайну система не лише виконує свої функціональні завдання, але й забезпечує позитивний досвід користувача, що сприяє підвищенню продуктивності та ефективності роботи з файловими сховищами.

## 7 ТЕСТУВАННЯ ТА ОПТИМІЗАЦІЯ СИСТЕМИ

Розробка програмного забезпечення неможлива без ретельного тестування та оптимізації, що забезпечують високу якість, надійність та продуктивність системи. У цьому розділі розглядаються процеси тестування функціональності, навантажувального тестування, а також тестування безпеки та надійності автоматизованої системи аналізу файлових сховищ.

### 7.1 Тестування функціональності та навантажувальне тестування

#### 7.1.1 Тестування функціональності

Тестування функціональності системи має на меті перевірку коректності роботи всіх її компонентів відповідно до вимог та специфікацій. Основними етапами тестування функціональності є:

- аналіз вимог та розробка тест-кейсів: на основі функціональних вимог було розроблено набір тест-кейсів, що охоплюють всі основні сценарії використання системи, включаючи як позитивні, так і негативні випадки.
- модульне тестування: кожен окремий модуль системи, включаючи серверну та клієнтську частини, було протестовано ізольовано. використовувалися інструменти для модульного тестування, такі як Nunit.
- інтеграційне тестування: після успішного модульного тестування проводилося інтеграційне тестування, яке перевіряло взаємодію між різними компонентами системи. це включало перевірку коректності передачі даних між сервером, клієнтом та базою даних.
- системне тестування: система тестувалася як єдине ціле в середовищі, максимально наближеному до реального. це дозволило виявити можливі проблеми, які не проявилися на попередніх етапах.
- регресійне тестування: після внесення змін або виправлень проводилося регресійне тестування для перевірки того, що нові зміни не вплинули негативно на вже існуючу функціональність.[25]

Результати тестування функціональності, наведені у таблиці 7.1, підтвердили, що система відповідає встановленим вимогам і коректно виконує всі необхідні операції, включаючи збір даних, обробку та аналіз інформації, генерацію звітів та відображення їх у клієнтському інтерфейсі.

Таблиця 7.1 - Тестування функціональності

№	Назва тесту	Опис тесту	Результат виконання
	Тест фільтрації за категоріями	Перевірка коректності фільтрації даних за компанією, службою, відділом та сектором.	ОК
	Тест вибору періоду часу	Перевірка коректності вибору періоду (7, 30, 180 днів) для відображення даних.	ОК
	Тест адаптивності інтерфейсу	Перевірка коректного відображення інтерфейсу на різних розмірах екранів (десктоп, планшет, мобільний).	ОК
	Тест відображення графіків	Перевірка, що графіки (лінійний та стовпчасті) відображаються коректно та адаптивно на всіх пристроях.	ОК
	Тест лінійного графіка	Перевірка, що лінійний графік відображає зміни в використанні папок за вибраний період часу.	ОК
	Тест стовпчастих діаграм	Перевірка, що стовпчасті діаграми показують використання дискового простору за	ОК

№	Назва тесту	Опис тесту	Результат виконання
		компаніями, службами, відділами та секторами.	
	Тест відображення таблиці даних	Перевірка, що таблиця даних коректно відображає всі папки, фільтровані за обраними критеріями.	ОК
	Тест сортування даних у таблиці	Перевірка можливості сортування даних у таблиці за різними параметрами (наприклад, за розміром).	ОК
	Тест взаємодії з формою фільтрів	Перевірка, що фільтри застосовуються коректно після вибору користувачем значення та натискання кнопки "Apply Filters".	ОК
	Тест PowerShell скриптів для збору даних	Перевірка коректної роботи PowerShell скриптів для збору інформації з серверів. Скрипти повинні без помилок отримувати актуальні дані про використання дискового простору.	ОК
	Тест обробки помилок PowerShell скриптів	Перевірка, що система коректно обробляє помилки, які можуть виникнути під час роботи PowerShell скриптів, і надає користувачу відповідну інформацію.	ОК
	Тест оновлення даних у системі	Перевірка, що дані, зібрані PowerShell скриптами, коректно	ОК

№	Назва тесту	Опис тесту	Результат виконання
		оновлюються у системі та відображаються у користувачькому інтерфейсі.	
	Тест продуктивності запитів	Перевірка часу виконання запитів до бази даних для відображення великих обсягів даних без затримок.	ОК
	Тест безпеки доступу до даних	Перевірка, що доступ до конфіденційних даних може отримати лише авторизований користувач.	ОК
	Тест інтеграції з базою даних	Перевірка коректної інтеграції з базою даних та отримання інформації про використання дискового простору.	ОК
	Тест кешування даних	Перевірка, що дані кешуються для зменшення навантаження на сервер та покращення швидкості завантаження.	ОК
	Тест редагування налаштувань	Перевірка можливості редагування налаштувань PowerShell скриптів, зокрема часів виконання та серверів, з яких збирається інформація.	ОК

### 7.1.2 Навантажувальне тестування

Навантажувальне тестування мало на меті оцінити продуктивність та стабільність системи під час роботи з великими обсягами даних та під високим навантаженням користувачів.

Методологія навантажувального тестування включала наступні кроки:

- визначення метрик продуктивності: було встановлено ключові показники, такі як час відгуку системи, пропускна здатність, використання ресурсів (процесора, пам'яті, дискового вводу/виводу).
- підготовка тестового середовища: створено середовище, яке імітує реальні умови експлуатації, включаючи конфігурацію апаратного та програмного забезпечення.
- розробка сценаріїв навантаження: розроблено сценарії, що моделюють поведінку користувачів, включаючи одночасний доступ великої кількості користувачів, масове введення даних та виконання складних запитів.
- виконання тестів: за допомогою інструментів, таких як Apache Jmeter , проводилися тести з поступовим збільшенням навантаження.
- аналіз результатів: зібрані дані аналізувалися для виявлення вузьких місць та потенційних проблем продуктивності.

Результати навантажувального тестування показали, що система здатна обробляти значні обсяги даних та витримувати високі навантаження без суттєвого погіршення продуктивності. Проте було виявлено декілька областей для оптимізації, зокрема в оптимізації запитів до бази даних та використання кешування.

### 7.1.3 Оптимізація продуктивності

На основі результатів навантажувального тестування було виконано ряд заходів для оптимізації продуктивності системи:

- оптимізація запитів до бази даних: переглянуто та оптимізовано sql-запити, додано індекси до часто використовуваних полів, що зменшило час виконання запитів.
- впровадження кешування: реалізовано механізми кешування даних на стороні сервера та клієнта, що знизило навантаження на базу даних та прискорило відгук системи.
- оптимізація коду: проаналізовано та оптимізовано алгоритми обробки даних, що зменшило використання ресурсів та покращило швидкодію системи.
- масштабування: розглянуто можливості горизонтального масштабування серверів додатків для розподілу навантаження при збільшенні кількості користувачів.

## 7.2 Тестування безпеки та надійності системи

### 7.2.1 Тестування безпеки

Безпека є одним з критичних аспектів системи, оскільки вона працює з конфіденційними даними. Тестування безпеки включало:

- перевірка аутентифікації та авторизації: тестування механізмів входу в систему, перевірка правильності реалізації ролей та прав доступу.
- тестування на вразливості: виконано сканування системи на поширені вразливості, такі як SQL-ін'єкції, XSS (міжсайтовий скриптинг), CSRF (підробка міжсайтових запитів), за допомогою інструментів типу OWASP ZAP.
- перевірка шифрування даних: переконанося, що конфіденційні дані, такі як паролі та особиста інформація, зберігаються та передаються у зашифрованому вигляді з використанням SSL/TLS та хешування паролів з сіллю.
- аудит безпеки: аналіз журналів системи для виявлення підозрілої активності та спроб несанкціонованого доступу.[26]

Результати тестування безпеки підтвердили відсутність критичних вразливостей у системі. Виявлені незначні проблеми були оперативно виправлені, що підвищило загальний рівень безпеки.

### 7.2.2 Тестування надійності

Тестування надійності спрямоване на перевірку стабільності роботи системи та її здатності відновлюватися після збоїв.

– тестування відмовостійкості: імітувалися збої компонентів системи (наприклад, відключення бази даних), перевірялася коректність реакції системи та можливість збереження даних.

– тестування на стійкість до помилок: вводилися некоректні або несподівані дані для перевірки того, як система обробляє помилки та чи надає вона коректні повідомлення користувачу.

– тестування резервного копіювання та відновлення: перевірялася працездатність механізмів резервного копіювання даних та їх відновлення у разі втрати або пошкодження.

Результати тестування надійності показали, що система здатна коректно обробляти помилки та відновлюватися після збоїв без втрати даних. Механізми резервного копіювання та відновлення працюють належним чином.

### 7.2.3 Покращення безпеки та надійності

Після проведеного тестування було впроваджено наступні покращення:

– підсилення політик безпеки: введено обмеження на кількість спроб входу, вимоги до складності паролів, регулярне оновлення облікових даних.

– розширене журналювання: додано детальніше логування дій користувачів та системних подій для полегшення моніторингу та аудиту.

– впровадження додаткових механізмів захисту: реалізовано захист від CSRF-атак, використання HTTP-заголовків для підвищення безпеки (Content Security Policy).

– оптимізація резервного копіювання: налаштовано автоматизоване резервне копіювання бази даних з можливістю швидкого відновлення.[27]

## Висновки до розділу 7

Проведене тестування та оптимізація системи підтвердили її готовність до впровадження у реальних умовах. Система відповідає встановленим вимогам, демонструє високу продуктивність та стабільність під навантаженням, має належний рівень безпеки та надійності.

Ретельне тестування функціональності забезпечило коректну роботу всіх компонентів системи та їх взаємодію. Навантажувальне тестування дозволило виявити та усунути потенційні проблеми продуктивності, що підвищило швидкодію та ефективність системи.

Тестування безпеки та надійності виявило можливі вразливості та області для покращення, які були успішно усунені. Впроваджені заходи підвищили рівень захисту даних та стійкість системи до збоїв.

Загалом, система готова до експлуатації та здатна забезпечити ефективне управління файловими сховищами, підтримуючи прийняття обґрунтованих рішень та оптимізацію використання ресурсів в організаціях.

## 8 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ

Стартап — це новий бізнес-проект, що орієнтований на вирішення конкретної проблеми або задоволення певної потреби на ринку. Його основними характеристиками є інноваційність, можливість швидкого масштабування та супутній ризик. Стартапи можуть з'являтися в різних сферах — від технологій і послуг до виробництва і обслуговування.

Багато стартапів стикаються з труднощами через низку типових проблем. Основною з них є недостатня розробленість бізнес-моделі та стратегії. Недостатнє розуміння цільової аудиторії та неясне формулювання ідеї можуть призвести до неефективного використання ресурсів. Крім того, фінансові труднощі, висока конкуренція і неправильне управління можуть стати причиною фінансових втрат.[28]

### 8.1 Опис ідеї проекту

В таблиці 8.1 наведена інформація про зміст проекту, потенційні напрямки застосування та вигоди, які матиме користувач.

Таблиця 8.1 – Опис ідеї стартап-проекту

Параметр	Опис
Назва проекту	Автоматизована система аналізу файлових сховищ
Мета проекту	Розробка програмного забезпечення для автоматизованого збору, аналізу та візуалізації даних про файлові сховища, що дозволить оптимізувати використання дискового простору та підвищити ефективність управління ІТ-ресурсами організацій.
Цільова аудиторія	<ul style="list-style-type: none"> <li>- ІТ-відділи середніх та великих підприємств</li> <li>- Системні адміністратори</li> <li>- ІТ-менеджери</li> <li>- Компанії з великою кількістю даних та файлових серверів</li> </ul>

Параметр	Опис
Проблема, яку вирішує проект	Необхідність ефективного управління дисковим простором, виявлення аномалій у використанні файлових сховищ, прогнозування потреб у ресурсах та зниження витрат на зберігання даних.
Унікальна пропозиція	<ul style="list-style-type: none"> <li>- Інтуїтивно зрозумілий інтерфейс</li> <li>- Автоматизований збір та аналіз даних</li> <li>- Візуалізація інформації в реальному часі</li> <li>- Можливість інтеграції з існуючими ІТ-системами</li> <li>- Висока масштабованість та безпека</li> </ul>
Потенційні вигоди для користувача	<ul style="list-style-type: none"> <li>- Оптимізація використання дискового простору</li> <li>- Зниження витрат на зберігання даних</li> <li>- Підвищення ефективності роботи ІТ-відділу</li> <li>- Швидке виявлення та усунення проблем з файловими сховищами</li> <li>- Прийняття обґрунтованих рішень на основі актуальних даних</li> </ul>

В таблиці 8.2 наведено аналіз потенційних техніко-економічних переваг ідеї. Таблиця 8.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

Характеристика	Опис	Оцінка
Сильні сторони	<ul style="list-style-type: none"> <li>- Використання сучасних технологій (.NET Core, ASP.NET Core MVC, Entity Framework Core)</li> <li>- Висока продуктивність та масштабованість</li> <li>- Інтуїтивний інтерфейс користувача</li> <li>- Автоматизація процесів збору та аналізу даних</li> <li>- Можливість інтеграції з іншими системами через API</li> </ul>	Позитивна

Характеристика	Опис	Оцінка
Слабкі сторони	<ul style="list-style-type: none"> <li>- Висока конкуренція на ринку аналогічних рішень</li> <li>- Потреба у висококваліфікованих спеціалістах для розробки та підтримки</li> <li>- Можливі складнощі з впровадженням у великих організаціях з комплексною інфраструктурою</li> </ul>	Негативна
Нейтральні фактори	<ul style="list-style-type: none"> <li>- Залежність від платформи Windows Server</li> <li>- Вимоги до безпеки та відповідності стандартам можуть збільшити час розробки</li> </ul>	Нейтральна

Аналіз техніко-економічних характеристик ідеї, що передбачає розробку системи, дає змогу оцінити її переваги порівняно з конкурентними рішеннями. Розроблювана система характеризується високою швидкістю роботи, ефективним використанням ресурсів, гнучкістю та високим рівнем безпеки інформації. Особливою перевагою є інтуїтивно зрозумілий інтерфейс і доступна вартість впровадження та підтримки, що робить її привабливою для сучасного ринку.

## 8.2 Технологічний аудит ідеї проекту

Технологічний аудит ідеї проекту є важливим етапом у визначенні її життєздатності та ефективності реалізації. Цей процес дозволяє оцінити потенціал обраних технологій, їх відповідність вимогам проекту та можливість інтеграції в єдину функціональну систему. Технологічний аудит допомагає виявити сильні та слабкі сторони ідеї, що створює основу для обґрунтованого вибору інструментів і ресурсів, необхідних для успішної розробки та впровадження проекту [29]. Інформація щодо технологічного аудиту проекту наведена в таблиці 8.3.

Таблиця 8.3 – Технологічна здійсненість ідеї проекту

Критерій	Опис	Оцінка здійсненності
Використовувані технології	<ul style="list-style-type: none"> <li>- .NET Core для серверної частини</li> <li>- ASP.NET Core MVC для веб-інтерфейсу</li> <li>- Entity Framework Core для роботи з БД</li> <li>- PowerShell для збору даних з серверів Windows</li> <li>- Microsoft SQL Server як СУБД</li> <li>- Bootstrap та Chart.js для інтерфейсу та візуалізації</li> </ul>	Висока
Придатність технологій для завдань проекту	Обрані технології добре підходять для розробки корпоративних додатків з високими вимогами до продуктивності, безпеки та масштабованості. Використання PowerShell дозволяє ефективно збирати дані з Windows Server.	Висока
Інтеграція компонентів	Використання єдиної платформи .NET забезпечує легку інтеграцію між компонентами системи. API дозволяє взаємодіяти з іншими системами.	Висока
Наявність ресурсів та експертизи	Є достатня кількість розробників з досвідом роботи з .NET та пов'язаними технологіями. Велика спільнота підтримки та документація.	Висока
Ризики та обмеження	Можлива залежність від оновлень та підтримки з боку Microsoft. Необхідність забезпечення високого рівня безпеки.	Помірна

### 8.3 Аналіз ринкових можливостей запуску стартап-проекту

Розробка автоматизованої системи вимагає не лише оцінки технічної ефективності, але й аналізу ринкового середовища. У цьому контексті важливо розглянути як ринкові можливості, так і потенційні загрози, які можуть вплинути на успішність і ефективність розробленої системи, спроектованої для масштабованої та гнучкої роботи в інтернет-просторі. Інформація щодо актуального ринку таких систем наведена в таблиці 8.4.

Таблиця 8.4 – Попередня характеристика потенційного ринку стартап-проекту

Параметр	Опис
Розмір ринку	Значний, оскільки багато середніх та великих підприємств використовують Windows Server та мають потребу в управлінні файловими сховищами.
Темпи зростання ринку	Стабільне зростання у зв'язку зі збільшенням обсягів даних та потребою в ефективному управлінні ними.
Цільові сегменти	<ul style="list-style-type: none"> <li>- Корпоративні клієнти</li> <li>- Державні установи</li> <li>- ІТ-компанії</li> <li>- Фінансові установи</li> </ul>
Конкурентне середовище	Присутні кілька основних конкурентів, але ринок не перенасичений. Є можливість зайняти нішу завдяки унікальним перевагам продукту.
Тенденції ринку	<ul style="list-style-type: none"> <li>- Підвищення значення управління даними</li> <li>- Переорієнтація на хмарні технології</li> <li>- Зростання уваги до безпеки та конфіденційності даних</li> </ul>

За результатами аналізу таблиці можна зробити висновок, що ринок систем аналізу файлових сховищ є перспективним для входу. Оцінка основних показників свідчить про стабільний обсяг продажів, постійне зростання попиту та високу середньорічну рентабельність. Хоча існують певні обмеження для входу, що можуть стати викликом для нових гравців, швидке прийняття інновацій на ринку

вказує на наявність можливостей для успішного виходу з конкурентоспроможними продуктами та рішеннями.

Наступні етапи нашого дослідження включають визначення характеристик і вимог для різних груп клієнтів, що представлені в таблиці 8.5. Це дозволить адаптувати продукт до різних потреб і уявлень про ідеальну систему аналізу.

Таблиця 8.5 – Характеристика потенційних клієнтів стартап-проекту

Група клієнтів	Характеристики	Потреби
Корпорації	Великі підприємства з розгалуженою ІТ-інфраструктурою та великими обсягами даних	<ul style="list-style-type: none"> <li>- Ефективне управління ресурсами</li> <li>- Безпека та відповідність стандартам</li> <li>- Інтеграція з існуючими системами</li> </ul>
Державні установи	Організації з високими вимогами до безпеки та надійності	<ul style="list-style-type: none"> <li>- Захист даних та конфіденційність</li> <li>- Надійність та відмовостійкість</li> <li>- Підтримка нормативних вимог</li> </ul>
ІТ-компанії	Компанії, що надають ІТ-послуги або мають власні центри обробки даних	<ul style="list-style-type: none"> <li>- Гнучкість та масштабованість</li> <li>- Можливість кастомізації</li> <li>- Підтримка сучасних технологій</li> </ul>

Група клієнтів	Характеристики	Потреби
Фінансові установи	Банки, страхові компанії та інші фінансові організації	- Високий рівень безпеки - Детальний аудит та звітність - Підтримка відповідності регуляторним вимогам

Ці загальні потреби ринку дають змогу більш широко розглянути вимоги споживачів та орієнтувати розробку продукту на задоволення ключових потреб різних цільових аудиторій.

Для розроблюваного проекту, що передбачає створення та впровадження автоматизованої системи аналізу файлових сховищ, важливо детально вивчити фактори, які можуть вплинути на його ринковий успіх. У цьому контексті таблиці 8.6, що містять фактори загроз, та таблиці 8.7, які містять фактори можливостей, надають необхідний інструментарій для аналізу зовнішнього середовища та планування стратегій для подальшого розвитку проекту.

Таблиця 8.6 – Фактори загроз

Фактор	Опис	Вплив
Висока конкуренція	Присутність на ринку сильних конкурентів з подібними продуктами	Високий
Зміни в технологіях	Швидкий розвиток технологій може зробити продукт застарілим	Середній
Економічна нестабільність	Економічні кризи можуть знизити інвестиції в ІТ	Низький

Фактор	Опис	Вплив
Регуляторні зміни	Нові закони та регуляції можуть вимагати додаткових змін у продукті	Середній

Таблиця 8.7 – Фактори можливостей

Фактор	Опис	Вплив
Зростання обсягів даних	Потреба в ефективному управлінні великими даними зростає	Високий
Переорієнтація на цифровізацію	Бізнеси інвестують у цифрові технології та автоматизацію	Високий
Попит на безпеку даних	Збільшується потреба у безпечних рішеннях для управління даними	Високий
Можливості партнерства	Співпраця з іншими компаніями може розширити ринок збуту	Середній

В результаті аналізу таблиць факторів загроз і можливостей було визначено ключові аспекти, які слід враховувати в стратегії ринкового впровадження нашого проекту.

Наступним кроком проведемо ступеневий аналіз конкуренції на ринку, що наведено в таблиці 8.8.

Таблиця 8.8 – Ступеневий аналіз конкуренції на ринку

Рівень конкуренції	Характеристика	Приклади компаній
Прямі конкуренти	Компанії, що пропонують аналогічні продукти з подібною функціональністю	- SolarWinds
Непрямі конкуренти	Компанії, що пропонують альтернативні рішення для управління даними	- Jam software
Потенційні конкуренти	Нові гравці або існуючі компанії, які можуть увійти в цей сегмент ринку	- Великі ІТ-корпорації - Стартапи з інноваційними рішеннями

Ця таблиця дозволяє провести ступеневий аналіз конкуренції на ринку, що дає важливі вказівки для формування стратегій конкурентоспроможності та розвитку системи в умовах інтенсивної конкуренції.

У рамках аналізу конкуренції в галузі систем аналізу файлових сховищ за методологією Майкла Портера розглядається взаємодія різних сил, які визначають рівень конкурентоспроможності компаній у цій галузі. Модель "П'ять сил Портера" визначає основні фактори, що впливають на конкурентну боротьбу та можливості для розвитку бізнесу [30]. Опис цієї моделі наведений в таблиці 8.9.

Таблиця 8.9 – Аналіз конкуренції в галузі за М. Портером

Сила	Опис	Рівень впливу
Конкуренція між існуючими гравцями	Високий рівень конкуренції з боку інших компаній, що пропонують схожі продукти	Високий

Сила	Опис	Рівень впливу
Загроза появи нових конкурентів	Можливість входу на ринок нових гравців з інноваційними рішеннями	Середній
Сила постачальників	Залежність від технологій та інструментів, постачальники яких можуть впливати на бізнес	Низький
Сила покупців	Покупці можуть вимагати кращих умов або знижок, впливаючи на прибутковість	Середній
Загроза заміників	Можливість появи альтернативних рішень, які можуть замінити продукт	Середній

Зрозуміло, що на ринку присутня конкуренція, але є можливість продавати продукт, якщо підтримувати його інноваційність та закликати клієнтів, наприклад спеціальними планами тарифування. Фактори конкурентоспроможності наведені в таблиці 8.10.

Таблиця 8.10 – Обґрунтування факторів конкурентоспроможності

Фактор	Опис	Ступінь впливу
Інноваційність	Впровадження нових функцій та технологій, які відрізняють продукт від конкурентів	Високий
Якість продукту	Надійність, продуктивність та зручність використання	Високий
Цінова політика	Конкурентоспроможні ціни або гнучкі моделі тарифікації	Середній
Підтримка клієнтів	Якісний сервіс та технічна підтримка	Високий

Фактор	Опис	Ступінь впливу
Маркетинг та бренд	Відомість бренду та ефективність маркетингових кампаній	Середній

Далі необхідно провести аналіз сильних та слабких сторін розроблюваної системи відносно конкурентів, що наведено в таблиці 8.11.

Таблиця 8.11 – Порівняльний аналіз сильних та слабких сторін систем аналізу

Критерій	Наш продукт	Конкуренти
Функціональність	Широкий набір функцій, можливість кастомізації	Від середнього до високого рівня
Інтерфейс користувача	Інтуїтивно зрозумілий, сучасний дизайн	Може бути складним або застарілим
Ціна	Конкурентоспроможна, гнучкі тарифи	Часто вища або менш гнучка
Технічна підтримка	Оперативна, високий рівень сервісу	Варіюється, може бути недостатньою
Інтеграція	Легка інтеграція з іншими системами	Може вимагати додаткових зусиль

SWOT-аналіз (таблиця 8.12) – це стратегічний інструмент, який допомагає оцінити внутрішні та зовнішні чинники, що впливають на організацію або проект.

Таблиця 8.12 – SWOT-аналіз стартап-проекту

Сильні сторони (Strengths)	Слабкі сторони (Weaknesses)
<ul style="list-style-type: none"> <li>- Інноваційний продукт</li> <li>- Висока якість та надійність</li> <li>- Гнучкість та масштабованість</li> <li>- Сильна команда розробників</li> </ul>	<ul style="list-style-type: none"> <li>- Обмежені ресурси стартапу</li> <li>- Відсутність відомого бренду</li> <li>- Необхідність значних інвестицій у маркетинг</li> </ul>

Можливості (Opportunities)	Загрози (Threats)
<ul style="list-style-type: none"> <li>- Зростання попиту на управління даними</li> <li>- Можливість виходу на міжнародні ринки</li> <li>- Партнерство з великими компаніями</li> </ul>	<ul style="list-style-type: none"> <li>- Сильна конкуренція</li> <li>- Швидкі технологічні зміни</li> <li>- Економічні та політичні ризики</li> </ul>

На основі проведеного SWOT-аналізу формуються альтернативи ринкової поведінки для виведення стартап-проекту на ринок. Розробляється перелік заходів, який враховує потенційні загрози від проектів конкурентів та орієнтовний оптимальний час їх ринкової реалізації (табл. 8.13).

Таблиця 8.13 – Альтернативи ринкового впровадження стартап-проекту

Альтернатива	Опис	Переваги	Недоліки
Розробка MVP (Minimum Viable Product)	Створення мінімально життєздатного продукту для швидкого виходу на ринок	<ul style="list-style-type: none"> <li>- Швидкий фідбек від користувачів</li> <li>- Менші витрати на початковому етапі</li> </ul>	<ul style="list-style-type: none"> <li>- Обмежена функціональність</li> <li>- Може не задовольнити всі потреби клієнтів</li> </ul>
Партнерство з великою компанією	Співпраця з відомим гравцем ринку для спільного просування продукту	<ul style="list-style-type: none"> <li>- Доступ до ресурсів та клієнтської бази</li> <li>- Підвищення довіри до продукту</li> </ul>	<ul style="list-style-type: none"> <li>- Можливість втрати контролю над проектом</li> <li>- Поділ прибутку</li> </ul>

Альтернатива	Опис	Переваги	Недоліки
Самостійний вихід на ринок з повноцінним продуктом	Розробка повнофункціонального продукту перед виходом на ринок	- Повна функціональність - Можливість відразу задовольнити потреби клієнтів	- Більші витрати часу та ресурсів - Ризик, що продукт застаріє до виходу

З усіх запропонованих варіантів найбільш імовірним є розробка MVP. Це пояснюється високою ймовірністю отримання необхідних ресурсів та коротшими термінами реалізації порівняно з іншими варіантами. Такий підхід дозволяє швидше вивести продукт на ринок, отримати зворотний зв'язок від користувачів і оцінити його конкурентоспроможність.

#### 8.4 Розроблення ринкової стратегії продукту

Важливим етапом у розробці ринкової стратегії є визначення стратегії охоплення ринку, що включає аналіз цільових груп потенційних клієнтів. У таблиці наведено детальний опис профілів цільових груп, їх готовність до сприйняття нового продукту, орієнтований попит у відповідних сегментах, рівень конкуренції та складність входу в ці сегменти. Це дозволяє здійснити ретельний аналіз ринкового середовища та сформулювати оптимальну стратегію для виведення стартап-проекту на ринок. Детальний аналіз наведено в таблиці 8.14.

Таблиця 8.14 – Вибір цільових груп потенційних споживачів

Цільова група	Готовність приймати новий продукт	Орієнтований попит	Рівень конкуренції	Складність входу в сегмент
Корпорації	Висока	Високий	Середній	Середня
Державні установи	Середня	Середній	Низький	Висока (через бюрократію)

Цільова група	Готовність приймати новий продукт	Орієнтований попит	Рівень конкуренції	Складність входу в сегмент
ІТ-компанії	Висока	Середній	Середній	Низька
Малі та середні підприємства	Середня	Низький	Низький	Низька

На основі проведеного аналізу було визначено та обрано стратегічно важливі цільові сегменти: корпорації та ІТ-компанії. Ці групи характеризуються високою готовністю споживачів приймати новий продукт, значним орієнтованим попитом, а також стабільним попитом завдяки помірній інтенсивності конкуренції в сегментах. Таким чином, ці цільові групи є найбільш сприятливими для успішного виведення продукту на ринок.

Для ефективного впровадження продукту серед обраних сегментів необхідно розробити базову стратегію розвитку, стратегію конкурентної поведінки та стратегію позиціонування. Це дозволить забезпечити відповідність потребам ринку, успішно конкурувати та правильно позиціювати продукт серед споживачів, що наведено в таблицях 8.15, 8.16, 8.17 відповідно.

Таблиця 8.15 – Визначення базової стратегії розвитку

Елемент стратегії	Опис	Цілі	Кроки реалізації
Інновації та розвиток продукту	Постійне вдосконалення функціональності системи аналізу файлових сховищ шляхом впровадження нових технологій та можливостей.	- Підвищення конкурентоспроможності продукту - Задоволення змінних потреб користувачів	- Впровадження нових функцій на основі зворотного зв'язку - Проведення досліджень

Елемент стратегії	Опис	Цілі	Кроки реалізації
			ринку та технологій
Розширення ринків збуту	Вихід на нові ринкові сегменти та географічні регіони для збільшення обсягу продажів та охоплення потенційних клієнтів.	- Збільшення частки ринку - Диверсифікація джерел доходу	- Аналіз потенційних ринків - Розробка маркетингових стратегій для нових регіонів
Партнерства та співпраця	Встановлення стратегічних партнерств з іншими компаніями для спільного розвитку продукту та розширення функціональності.	- Розширення можливостей продукту - Спільне використання ресурсів	- Пошук потенційних партнерів - Укладання угод про співпрацю
Оптимізація внутрішніх процесів	Вдосконалення внутрішніх процесів розробки та підтримки системи для підвищення ефективності та зниження витрат.	- Зниження операційних витрат - Підвищення продуктивності команди	- Впровадження Agile методологій - Автоматизація процесів тестування та розгортання
Навчання та розвиток персоналу	Інвестування у професійний розвиток співробітників через навчальні програми,	- Підвищення кваліфікації команди - Залучення та утримання	- Організація регулярних тренінгів - Підтримка

Елемент стратегії	Опис	Цілі	Кроки реалізації
	тренінги та сертифікації.	талановитих спеціалістів	участі у професійних конференціях та семінарах

Таблиця 8.16 – Визначення базової стратегії конкурентної поведінки

Елемент стратегії	Опис	Цілі	Кроки реалізації
Диференціація продукту	Відмінність системи аналізу файлових сховищ від конкурентів за допомогою унікальних функцій та високої якості обслуговування.	- Збільшення лояльності клієнтів - Підвищення цінової привабливості продукту	Розробка унікальних функцій - Забезпечення високого рівня технічної підтримки
Цінова стратегія	Встановлення конкурентних цін на продукт з урахуванням вартості розробки, ринкових умов та цін конкурентів.	- Привернення ціново орієнтованих клієнтів - Збільшення обсягу продажів	- Аналіз цін конкурентів - Встановлення гнучкої системи ціноутворення
Стратегія фокусування	Орієнтація на конкретні сегменти ринку, де продукт може мати найвищу цінність для користувачів.	- Задоволення специфічних потреб клієнтів - Підвищення ефективності	- Визначення цільових сегментів - Розробка спеціалізованих

Елемент стратегії	Опис	Цілі	Кроки реалізації
		маркетингових зусиль	маркетингових кампаній
Стратегія інновацій	Постійне впровадження інноваційних рішень та технологій для підтримки конкурентоспроможності продукту на ринку.	- Підтримка лідерства на ринку - Забезпечення технологічного переваги	- Впровадження нових технологій у продукт
Стратегія партнерства	Співпраця з іншими компаніями та постачальниками для покращення пропозиції продукту та розширення функціональності.	- Розширення можливостей продукту - Збільшення ринкової присутності	- Укладання стратегічних альянсів - Спільна розробка продуктів з партнерами
Маркетингова стратегія	Використання ефективних маркетингових інструментів для просування продукту та залучення нових клієнтів.	- Підвищення обізнаності про продукт - Залучення нових клієнтів	- Використання цифрового маркетингу - Проведення рекламних кампаній та участь у виставках

Таблиця 8.17 –Визначення стратегії позиціонування

Елемент стратегії	Опис	Цілі	Кроки реалізації
Визначення цільового ринку	Ідентифікація та аналіз сегментів ринку, на які буде орієнтований продукт, з урахуванням їхніх потреб та особливостей.	- Чітке розуміння потреб клієнтів - Ефективне таргетування маркетингових зусиль	- Проведення ринкових досліджень - Сегментація ринку за галузями, розміром компаній тощо
Унікальна торгова пропозиція (УТП)	Визначення унікальних переваг продукту, які роблять його привабливим для цільових клієнтів у порівнянні з конкурентами.	- Виділення продукту на ринку - Підвищення цінової привабливості	- Формулювання чітких та привабливих УТП - Комунікація УТП через маркетингові канали
Брендова стратегія	Розробка та просування сильного бренду, який асоціюється з надійністю, інноваціями та високою якістю продукту.	- Побудова довіри клієнтів - Підвищення впізнаваності бренду	- Створення брендової ідентичності - Використання брендингових елементів у всіх маркетингових матеріалах
Комунікаційна стратегія	Визначення ефективних каналів	- Забезпечення ефективної	- Використання цифрових та

Елемент стратегії	Опис	Цілі	Кроки реалізації
	комунікації з цільовою аудиторією для передачі ключових повідомлень про продукт.	комунікації - Залучення та утримання клієнтів	традиційних каналів комунікації - Розробка контент-стратегії
Ціннісне позиціонування	Визначення цінностей, які пропонує продукт клієнтам, та комунікація цих цінностей для створення позитивного образу в умах споживачів.	- Зміцнення позитивного іміджу продукту - Підвищення лояльності клієнтів	- Комунікація цінностей через маркетингові матеріали - Підкреслення переваг продукту у всіх комунікаціях
Конкурентне позиціонування	Аналіз конкурентів та визначення позиції продукту на ринку відносно них, з урахуванням сильних та слабких сторін конкурентів.	- Забезпечення конкурентної переваги - Визначення ніші на ринку	- Проведення конкурентного аналізу - Визначення та акцентування на своїх сильних сторонах
Позиціонування за якістю	Фокус на високій якості продукту як	- Створення образу	- Забезпечення високих

Елемент стратегії	Опис	Цілі	Кроки реалізації
	ключовому елементі позиціонування, що відрізняє його від конкурентів.	високоякісного продукту - Залучення клієнтів, орієнтованих на якість	стандартів якості - Отримання сертифікатів та нагород, підтверджуючих якість
Позиціонування за ціною	Визначення стратегії цінового позиціонування, яка може бути як преміальною, так і бюджетною, залежно від цільового ринку та конкурентів.	- Залучення ціноорієнтованих клієнтів - Підвищення конкурентоспроможності через цінову політику	- Встановлення цінової політики відповідно до ринкових умов - Використання знижок та акцій для залучення клієнтів

Стратегія наслідування лідера дозволяє уникнути значних ризиків, пов'язаних із впровадженням новаторських рішень. Опіраючись на успіх лідера, компанія може зменшити ймовірність невдач та забезпечити стабільний старт.

### 8.5 Розроблення маркетингової програми стартап-проекту

У розділі, що стосується розробки маркетингової програми стартап-проекту, розглядається стратегічний план взаємодії з цільовою аудиторією та рекламно-комунікаційні заходи, необхідні для успішного виведення продукту на ринок.

Маркетингова програма визначає ключові напрямки, що спрямовані на підвищення впізнаваності бренду, залучення клієнтів та забезпечення конкурентоспроможності, узгоджуючи ці дії з поставленими стратегічними цілями.

Потенційні вигоди для користувачів, які підсумовують попередній аналіз конкурентоспроможності продукту, наведені в таблиці 8.18.

Таблиця 8.18 – Визначення ключових переваг концепції потенційного товару

Перевага	Опис
Інноваційність	Використання сучасних технологій та методів аналізу даних.
Висока продуктивність	Швидка обробка великих обсягів даних без втрати продуктивності.
Інтуїтивний інтерфейс	Зручний та зрозумілий користувацький інтерфейс, що підвищує ефективність роботи.
Гнучкість та масштабованість	Легке налаштування під потреби клієнта та можливість масштабування системи.
Безпека даних	Високі стандарти захисту інформації та відповідність регуляторним вимогам.

Далі наведена маркетингова модель проекту (таблиця 8.19), що уточнює ідеї проекту, послуги, які він надає, та технічну реалізацію.

Таблиця 8.19 – Опис трьох рівнів моделі товару

Рівень товару	Опис
Концепція товару	Автоматизована система аналізу файлових сховищ, що допомагає організаціям оптимізувати використання дискового простору та підвищити ефективність ІТ-інфраструктури.
Формальний товар	Програмне забезпечення з певним набором функцій: збір даних, аналіз, візуалізація, сповіщення, звітність, інтерфейс користувача, технічна підтримка.
Додатковий товар	Додаткові послуги та можливості: кастомізація під потреби клієнта, навчання персоналу, інтеграція з іншими системами, оновлення та підтримка, консультації.

Для визначення оптимальної стратегії ціноутворення продукту необхідно провести аналіз цін на товари-замінники та аналоги, який представлений у таблиці 8.20, а також врахувати рівень доходів цільової групи споживачів. Такий підхід дозволить встановити верхні та нижні межі ціноутворення, забезпечуючи конкурентоспроможність продукту та його привабливість для цільової аудиторії.

Таблиця 8.20 – Визначення меж встановлення ціни

Показник	Значення
Верхня межа ціни	Ціна конкурентів на аналогічні продукти або максимальна ціна, яку готові сплатити клієнти за додаткову цінність.
Нижня межа ціни	Собівартість продукту з урахуванням витрат на розробку, підтримку та маркетинг.
Пропонована ціна	Ціна, яка забезпечує конкурентоспроможність та прибутковість, враховуючи переваги продукту та готовність клієнтів платити за нього.

Наступним кроком йде формування системи збуту. Їх потрібно формулювати з урахуванням цілі найбільшого прибутку і найменших втрат (таблиця 8.21).

Таблиця 8.21 – Формування системи збуту

Канал збуту	Опис	Переваги
Прямі продажі	Продаж продукту безпосередньо клієнтам через власний відділ продажів або онлайн-платформу	- Контроль над процесом продажів - Прямий контакт з клієнтом - Вища маржинальність
Партнерські програми	Співпраця з іншими компаніями або дистриб'юторами, які продають продукт своїм клієнтам	- Розширення ринку збуту - Зниження витрат на маркетинг

Канал збуту	Опис	Переваги
		- Використання існуючих клієнтських баз партнерів
Онлайн-маркетплейси	Розміщення продукту на спеціалізованих платформах для продажу програмного забезпечення	- Доступ до широкої аудиторії - Спрощений процес покупки для клієнта

Таблиця 8.22 – Концепція маркетингових комунікацій

Інструмент	Опис	Ціль
SEO та контент-маркетинг	Оптимізація веб-сайту для пошукових систем, створення корисного контенту (блоги, статті, вебінари)	- Підвищення видимості в інтернеті - Залучення потенційних клієнтів
Соціальні мережі	Присутність у професійних мережах (LinkedIn, Twitter), взаємодія з аудиторією	- Формування бренду - Залучення клієнтів через соціальні канали
Email-маркетинг	Розсилка інформаційних та промоційних листів до бази потенційних клієнтів	- Підтримка контакту з аудиторією - Інформування про оновлення та пропозиції
Участь у виставках та конференціях	Представлення продукту на галузевих заходах	- Підвищення обізнаності про продукт - Налагодження партнерських відносин

Інструмент	Опис	Ціль
Реклама в інтернеті	Контекстна реклама, банери, таргетована реклама в соціальних мережах	- Швидке залучення цільової аудиторії - Підвищення трафіку на веб-сайт

## Висновки до розділу 8

Розділ "Розроблення стартап-проекту" охоплює кілька ключових етапів, які спрямовані на всебічне вивчення, аналіз і підготовку ідеї проекту для її успішного виведення на ринок. Цей процес є основою для забезпечення конкурентоспроможності продукту та його адаптації до потреб цільової аудиторії.

На першому етапі розглядається сама ідея проекту: визначаються її основні концепції, цілі та ключові аспекти, що підкреслюють унікальність і цінність продукту. Особлива увага приділяється ідентифікації проблем, які продукт має вирішувати, і формуванню його ціннісної пропозиції.

Етап технологічного аудиту передбачає детальний аналіз технічних аспектів інноваційної ідеї. Оцінюються технічні можливості, потенціал технологій і ресурси, які необхідні для реалізації проекту. Визначається, наскільки обрана технологія відповідає цілям проекту, та оцінюється її інтеграція з іншими складовими системи.

Далі проводиться комплексний ринковий аналіз, що дозволяє оцінити можливості та загрози в зовнішньому середовищі. Вивчаються характеристики цільової аудиторії, аналізуються конкуренти, виявляються їх сильні та слабкі сторони, а також визначаються ключові фактори, що впливають на успіх проекту. Цей етап дає змогу сформулювати реалістичне уявлення про ринок і шляхи його освоєння.

Етап розробки ринкової стратегії включає визначення основних конкурентних переваг продукту, які виділяють його серед аналогів. Вибираються найбільш ефективні способи для виведення продукту на ринок, його просування та

позиціонування серед цільової аудиторії. Важливо також визначити стратегії роботи з кожним сегментом ринку, щоб максимізувати успіх.

Маркетингова програма розробляється для забезпечення ефективної комунікації з цільовою аудиторією. Вона охоплює стратегії комунікації, створення бренду, аналіз конкуренції та розробку ключових повідомлень, що підкреслюють переваги продукту. Крім того, плануються заходи для підвищення впізнаваності бренду та формування лояльності споживачів.

Цей розділ спрямований на створення чіткого, детального та стратегічного плану для успішного запуску стартап-проекту. Він забезпечує фундамент для досягнення конкурентних позицій на ринку, залучення цільової аудиторії та розвитку бізнесу в умовах інтенсивної конкуренції.

## ВИСНОВКИ

При роботі над даною магістерською дисертацією була розроблена та впроваджена автоматизована система аналізу файлових сховищ для середовища Windows Server. Робота включала всебічний аналіз предметної області, огляд існуючих рішень, постановку задачі, формування системних вимог, математичне забезпечення, проектування системи, розробку бази даних, дизайн інтерфейсу користувача та UX/UI, тестування та оптимізацію системи, а також розроблення стартап-проекту.

Дослідження розпочалося з аналізу предметного середовища та процесу діяльності, пов'язаного з управлінням файловими сховищами на серверах Windows Server. Проведений огляд існуючих рішень, таких як SolarWinds Storage Resource Monitor, TreeSize Professional та ManageEngine FileAudit Plus, дозволив визначити їхні переваги та недоліки. Це стало основою для визначення ключових вимог та функціональності, яку має забезпечувати нова система.

На основі проведеного аналізу була сформульована постановка задачі, яка включала визначення призначення та цілей розробки, а також інформаційне забезпечення. Головною метою було створення інструменту, який автоматизує збір та аналіз даних про файлові сховища на серверах, надаючи компанії можливість моніторити використання дискового простору різними відділами, розраховувати вартість зберігання та планувати фінансові звіти та бюджетування.

Формування системних вимог включало розробку функціональних та нефункціональних вимог, а також визначення основних сутностей системи. Функціональні вимоги охоплювали можливості збору даних, генерації звітів, сповіщення про аномалії та інтеграцію з існуючими системами. Нефункціональні вимоги стосувалися продуктивності, безпеки, масштабованості та зручності використання системи.

У розділі математичного забезпечення були розроблені формули для розрахунку вартості зайнятого дискового простору, розподілу витрат між відділами, планування бюджетування та аналізу ефективності використання дискового простору. Додатково були розроблені методи для розрахунку середнього

зростання даних та часу до повного заповнення дискового простору. Ці математичні моделі були інтегровані в систему, забезпечуючи точні та актуальні розрахунки.

Проектування системи включало розробку сценаріїв використання, моделювання та проектування архітектури, а також вибір технологій. Серед сценаріїв були: збір даних про файлові сховища, перегляд звітів про використання дискового простору, планування бюджету на зберігання даних, отримання сповіщень про аномалії та запит інформації про використання дискового простору. Архітектура системи базувалася на модульному підході з чітким розподілом серверної та клієнтської частин. Для серверної частини були використані PowerShell-скрипти для збору даних та API для передачі інформації. Клієнтська частина забезпечувала інтуїтивний інтерфейс для візуалізації даних.

У процесі вибору технологій були обрані оптимальні інструменти для розробки, включаючи мову програмування, фреймворки та середовище розробки. Це забезпечило ефективну реалізацію системи з урахуванням вимог до продуктивності та масштабованості.

Розробка бази даних включала проектування структури та опис таблиць, таких як Servers (Сервери), Folders (Папки), Departments (Відділи), Users (Користувачі), DiskUsages (Використання дисків), Notifications (Сповіщення) та Settings (Налаштування). База даних була оптимізована для швидкого доступу та обробки великих обсягів даних.

Дизайн інтерфейсу користувача та UX/UI базувався на принципах простоти та мінімалізму, інтуїтивної навігації, консистентності дизайну, зворотного зв'язку, доступності та можливості персоналізації. Було забезпечено адаптивність та кросплатформенність інтерфейсу, що дозволило використовувати систему на різних пристроях та платформах.

Тестування та оптимізація системи включали функціональне та навантажувальне тестування, оптимізацію продуктивності, а також тестування безпеки та надійності. Функціональне тестування підтвердило коректність роботи всіх модулів системи. Навантажувальне тестування виявило можливості системи

обробляти великі обсяги даних без втрати продуктивності. Тестування безпеки виявило та дозволило усунути потенційні уразливості, підвищуючи загальний рівень захищеності системи.

У заключному розділі було розроблено стартап-проект, що включав опис ідеї проекту, технологічний аудит, аналіз ринкових можливостей, розробку ринкової стратегії та маркетингової програми. Було створено детальний план для успішного виведення системи на ринок, з урахуванням конкурентного середовища та потреб потенційних клієнтів.

Загальний висновок дослідження свідчить про успішне розв'язання поставлених завдань у магістерській дисертації, а отже мету дисертації було досягнуто. Розроблена автоматизована система аналізу файлових сховищ для середовища Windows Server відповідає високим стандартам якості та готова до успішного впровадження в корпоративних середовищах. Вона надає компаніям ефективні інструменти для моніторингу та аналізу використання дискового простору, що сприяє оптимізації витрат, покращенню управління ресурсами та прийняттю обґрунтованих управлінських рішень.

У рамках дослідження було впроваджено систему в компанію ТОВ “АТБ ЕНЕРГО”, яка підтвердила свою ефективність у реальних умовах. Довідка про впровадження результатів магістерської дисертації представлена в Додатку Л.

Результати роботи можуть бути використані для подальших досліджень та розробок у галузі систем управління файловими сховищами та ІТ-інфраструктурою. Запропонована система має потенціал для розширення функціональності, адаптації до інших операційних систем та інтеграції з іншими корпоративними системами. QR-код з посиланням на серверну частину системи представлений в Додатку М

Таким чином, виконана робота підтверджує актуальність та важливість автоматизації процесів збору та аналізу даних у сучасних інформаційних системах. Розроблена система не лише вирішує поставлені задачі, але й відкриває нові можливості для підвищення ефективності та конкурентоспроможності компаній у сфері управління ІТ-ресурсами.

**ПЕРЕЛІК ІНФОРМАЦІЙНИХ ДЖЕРЕЛ**

1. SolarWinds. (2024). Storage Resource Monitor Overview. URL: <https://www.solarwinds.com/storage-resource-monitor> (дата звернення: 15.04.2024).
2. JAM Software. (2024). TreeSize Free. URL: [https://www.jam-software.com/treesize\\_free](https://www.jam-software.com/treesize_free) (дата звернення: 12.04.2024).
3. ManageEngine. (2024). FileAudit Plus User Guide. URL: <https://www.manageengine.com/fileaudit-plus> (дата звернення: 12.04.2024).
4. Microsoft. (2023). PowerShell Documentation. URL: <https://learn.microsoft.com/en-us/powershell/> (дата звернення: 20.04.2024).
5. Booch, G. (1994). Object-Oriented Analysis and Design with Applications. Addison-Wesley. ISBN: 978-0201895513.
6. Provost, F., & Fawcett, T. (2013). Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking. O'Reilly Media. ISBN: 978-1449361327.
7. Fowler, M. (2002). Patterns of Enterprise Application Architecture. Addison-Wesley. ISBN: 978-0321127426.
8. Evans, E. (2003). Domain-Driven Design: Tackling Complexity in the Heart of Software. Addison-Wesley. ISBN: 978-0321125217.
9. Hohpe, G., & Woolf, B. (2003). Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Addison-Wesley. ISBN: 978-0321200686.
10. Martin, R. C. (2017). Clean Architecture: A Craftsman's Guide to Software Structure and Design. Prentice Hall. ISBN: 978-0134494166.
11. Cambell, E. (2018). Microservices Architecture: Make the Architecture of a Software System Simple. Packt Publishing. ISBN: 978-1788621751.
12. Lankhorst, M. (2013). Enterprise Architecture at Work: Modelling, Communication and Analysis. Springer. ISBN: 978-3642208320.
13. Kleppmann, M. (2017). Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems. O'Reilly Media. ISBN: 978-1449373320.

14. Microsoft. (2023). C# Programming Guide. URL: <https://learn.microsoft.com/en-us/dotnet/csharp/> (дата звернення: 20.04.2024).
15. Microsoft. (2023). ASP.NET Documentation. URL: [https://learn.microsoft.com/en-us/aspnet/core/?view=aspnetcore-9.0&WT.mc\\_id=dotnet-35129-website](https://learn.microsoft.com/en-us/aspnet/core/?view=aspnetcore-9.0&WT.mc_id=dotnet-35129-website) (дата звернення: 20.04.2024).
16. Microsoft. (2023). Entity Framework Documentation. URL: <https://learn.microsoft.com/en-us/ef/> (дата звернення: 20.04.2024).
17. ChartJs. (2023). Chart.js Documentation. URL: <https://www.chartjs.org/docs/latest/> (дата звернення: 20.04.2024).
18. Bootstrap. (2023). Bootstrap Documentation. URL: <https://getbootstrap.com/docs/5.3/getting-started/introduction/> (дата звернення: 20.04.2024).
19. Garcia-Molina, H., Ullman, J. D., & Widom, J. (2008). Database Systems: The Complete Book. Pearson. ISBN: 978-0131873254.
20. Microsoft. (2023). Database Design Fundamentals. URL: <https://learn.microsoft.com/en-us/sql/ssdt/how-to-design-and-create-a-database?view=sql-server-ver15> (дата звернення: 20.04.2024).
21. Smashing Magazine. (2018). The Ultimate Guide to UX Design. URL: <https://www.smashingmagazine.com/2018/05/ultimate-guide-ux-design/> (дата звернення: 20.04.2024).
22. Krug, S. (2013). Don't Make Me Think: A Common Sense Approach to Web Usability. New Riders. ISBN: 978-0321965516.
23. Frain, B. (2015). Responsive Web Design with HTML5 and CSS3. Apress. ISBN: 978-1484204747.
24. Nielsen Norman Group. (2023). UX Design for Mobile Devices. URL: <https://www.nngroup.com/articles/ux-design-for-mobile-devices/> (дата звернення: 20.04.2024).
25. Myers, G. J. (2011). The Art of Software Testing (4th ed.). Wiley. ISBN: 978-1118031964.

26. Jorgensen, P. C. (2013). *Software Testing: A Craftsman's Approach*. CRC Press. ISBN: 978-1439820182.
27. Test Automation University. (2023). *Introduction to Load Testing*. URL: <https://testautomationu.applitools.com/load-testing/> (дата звернення: 20.04.2024).
28. Horowitz, B. (2014). *The Hard Thing About Hard Things: Building a Business When There Are No Easy Answers*. HarperBusiness. ISBN: 978-0062273208.
29. Blank, S., & Dorf, B. (2012). *The Startup Owner's Manual: The Step-by-Step Guide for Building a Great Company*. K&S Ranch. ISBN: 978-0984999309.
30. Sinek, S. (2009). *Start with Why: How Great Leaders Inspire Everyone to Take Action*. Penguin. ISBN: 978-1591846444.