МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук,
управління та адміністрування
Кафедра інформаційних технологій

Кваліфікаційна робота бакалавра

на тему: Розробка десктопного застосунку для створення backup даних на ftp-сервері

Виконав студент групи **К-21і**
спеціальності 122 Комп'ютерні науки
Буркадзе Владислав Русланович

Керівник **асистент**
Клепатська В.В.

Консультант **д.т.н., професор**
Казакова Н.Ф.

Рецензент **заступник директора**
- начальник відділу Департаменту
інформації та цифрових рішень
Одеської міської ради
Корчемний П.А.

Одеса 2023

ЗМІСТ

ТЕРМІНИ, СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

Диференціальне резервне копіювання – це процедура резервного копіювання даних, яка записує зміни даних, що відбулися з часу останнього повного резервного копіювання.

Повне резервне копіювання (Full Backup або L0) – повна копія даних (рівень, який забезпечує створення повної копії об'єкта резервного копіювання).

Правило резервного копіювання 3-2-1 – метод резервного копіювання даних для гарантування безпеки бізнес-даних.

Хмарне сховище (англ. cloud storage, або ще хмара, backup) – це модель збереження даних у комп'ютері, в якій цифрові дані накопичуються в логічні пули, а фізичне зберігання охоплює кілька серверів (зазвичай у кількох місцях).

Шифрування AES-256 – це спосіб захисту секретних повідомлень або інформації від людей, які не повинні бачити їх.

AceBackup – програмне забезпечення для резервного копіювання з підтримкою FTP.

API – набір визначень підпрограм, протоколів взаємодії та засобів для створення програмного забезпечення.

FTP – це процес, за допомогою якого організації можуть передавати великі та/або конфіденційні файли з одного місця в інше;

FTPS – це форма FTP використовує Transport Layer Security (TLS) під час підключення та є більш безпечним способом надсилання та отримання файлів із ідентифікаторами, паролями та сертифікатами для перевірки автентичності системи;

Handy Backup – це просте програмне забезпечення для резервного копіювання FTP-сервера, яке може створювати резервні копії будь-яких файлів і навіть підтримує резервне копіювання електронної пошти

Iperius Backup – це програма для резервного копіювання, яка також до-

зволяє виконувати резервне копіювання на сервери SFTP, Amazon S3, Microsoft Azure Storage, Google Drive, Dropbox і OneDrive.

SFTP – Secure File Transfer Protocol– SFTP забезпечує механізм у протоколі Secure Shell (SSH), використовуючи алгоритми та шифрування для захисту даних, а також ідентифікатори, паролі та ключі SHH для перевірки автентичності.

AES – Advanced Encryption Standard.

API – Application Programming Interface.

FTP – File Transferring Protocol .

FTPS – File Transfer Protocol Secure.

SFTP – Secure File Transfer Protocol.

SSH – Secure Shell.

TLS – Transport Layer Security.

ВСТУП

У сучасну цифрову епоху дані стали одним із найважливіших активів як для компаній, так і для окремих осіб. Від конфіденційних документів до дорогих спогадів, дані мають вирішальне значення як з особистих, так і з професійних причин. Однак втрата даних може статися з різних причин. Це включало збій апаратного забезпечення, стихійні лиха або кібератаки. Все це може призвести до незворотних наслідків. Отже, важливо мати стратегію резервного копіювання для захисту та відновлення цінних даних у разі будь-яких непередбачених обставин. У цій роботі проаналізовано важливість резервного копіювання даних і надано кілька корисних порад і стратегій, які допоможуть уникнути втрати даних і зберегти дані в безпеці.

Метою кваліфікаційної роботи бакалавра є розробка десктопного застосунку для створення резервних копій даних на FTP-сервері полягає в створенні зручного та надійного інструменту, який дозволить користувачам зберігати свої дані на віддаленому сервері для запобігання втрати і відновлення даних в разі необхідності.

FTP (File Transferring Protocol), також відомий як протокол передачі файлів, – це процес, за допомогою якого організації можуть передавати великі та/або конфіденційні файли з одного місця в інше. Захищені «версії», або альтернативи FTP SFTP (Secure File Transfer Protocol), FTPS (File Transfer Protocol Secure) особливо корисні в оборонній, юридичній, охороні здоров’я, виробництві та фінансах. Проте FTP загалом використовується майже в усіх галузях, де необхідний обмін інформацією. Резервне копіювання FTP – це форма передачі та захисту даних за допомогою протоколу передачі файлів із пристрою чи пристроїв на виділений сервер FTP. Резервне копіювання FTP зазвичай виконується віддалено за допомогою FTP-сервера, розташованого в централізованому центрі обробки даних подалі від користувача та його пристроїв, хоча користувач також може мати власний виділений сервер на місці, але це робить його апаратне забезпечення та дані вразливими на будь-які

стихійні лиха, такі як пожежа, вода або пошкодження, що можуть виникнути.

У зв’язку з цим рекомендовано, щоб усі резервні копії, створені за допомогою цього методу, створювалися як віддалене резервне копіювання FTP, щоб звести до мінімуму мзаожливість втрати даних через стихійні лиха, оскільки всі компанії, які пропонують службу резервного копіювання FTP на базі хмарної системи, матимуть багато більш захищений від такої форми пошкодження. Протокол передачі файлів найчастіше використовується для створення та оновлення веб-сайтів, коли веб-розробники підключають власний веб-сервер за допомогою FTP, щоб надсилати оновлені версії веб-сторінок на сервер. Веб-сайти та веб-застосунки також можна повністю оновити за допомогою FTP [1]. Поява хмарних обчислень як нової парадигми, що надає постачальникам послуг можливість розгортати економічно ефективні рішення, сприяло розробці низки нових послуг, включаючи програми для зберігання даних в Інтернеті. Через економію масштабу хмарних служб зберігання даних, витрати, понесені кінцевими користувачами на передачу даних у віддалене місце зберігання в Інтернеті, значно зменшились. Таким чином, сервіси менеджменту резервного копіювання даних оберігають користувачів від більшої частини трудомістких неприємностей при резервному копіюванні даних: взаємодія з користувачем мінімальна, а у випадку втрати даних через аварію, відновлення вихідних даних − це безперебійна операція.

Об’єкт дослідження автоматизованих систем резервного копіювання даних сервісів в системах. Предмет дослідження характеристики надійності резервування та створення консистентної резервної копії даних.

Перелік завдань, які потрібно розробити:

– аналіз проблеми резервного копіювання даних в застосунках;
– огляд існуючих рішень, створення математичної моделі,
– розробка алгоритму створення та автоматизації процесу резервного копіювання даних.

Дана кваліфікаційна робота бакалавра складається з 54 сторінки, 15 рисунків та 7 джерел посилання.

1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ

Кількість нових даних продовжує зростати в геометричній прогресії. У 2020 році створювались в середньому 1,7 МБ даних на секунду для кожного чоловіка, жінки та дитини на планеті Земля. Навіть характер даних, які збираються, змінюються з роками: від майже виключно структурованих даних, таких як таблиці та бази даних, до сучасних неструктурованих даних, включаючи зображення, відео та, що найважливіше, документи. Корпоративні дані також стають паливом для нових проєктів машинного навчання та штучного інтелекту, оскільки організації використовують інструменти для аналізу даних, щоб витягувати знання, створювати контекст і надавати практичну інформацію [2].

Зі збільшенням обсягу даних, які збирають підприємства, зростає і кількість резервних копій, які зберігаються для різних цілей. Щоденне, щотижневе, щомісячне та щорічне резервне копіювання часто потрібне для підтримки регулятивних аудитів, фінансових аудитів і юридичних відкриттів, це лише деякі вимоги. Підприємства все більше покладаються на розподілену робочу силу – незалежно від того, чи працюють вони з дому чи по всьому світу, – і в міру того, як обсяг даних зростає, користувачі також працюють більше годин поза звичайним робочим часом, створюючи більший тиск на скорочення вікон резервного копіювання.

Закриття вікна резервного копіювання це один із способів керування скороченням вікон резервного копіювання – повністю покладатися на недорогий основний дисковий накопичувач як носій резервного копіювання. Цей підхід є швидким, але стає надто дорогим, враховуючи кількість терабайт диска, необхідну для довгострокового зберігання щотижневих, місячних і річних резервних копій.

Іншим поширеним підходом до зберігання резервних копій у центрі обробки даних є вбудовані пристрої дедуплікації. Однак вони мають більше

проблем, ніж основний диск. По-перше, вони повільніші як для резервного копіювання, так і для відновлення через час, потрібний для виконання вбудованої дедуплікації та повторної гідратації резервних копій, коли потрібне відновлення. Основною проблемою є відсутність масштабованості з обсягом.

Розширення пристрою дедуплікації зазвичай відбувається лише на диск, тобто, незважаючи на додавання додаткового диска, обсяг ЦП, пам'яті та мережевих ресурсів у пристрої залишається незмінним. Це означає, що пристрій має дедуплікати більше даних із тією самою обчислювальною потужністю – фізично неможливе завдання. Таким чином, просте додавання диска до пристрою дедуплікації може фактично сповільнити процес резервного копіювання, ще більше зменшивши вікно резервного копіювання. Відсутність можливості масштабування призводить до двох небажаних наслідків: старіння продукту та необхідності модернізації навантажувача, коли потрібна більша потужність обробки.

Багаторівневе сховище резервних копій відповідає вимогам масштабованості. Багаторівневе сховище резервних копій, як з рівнями продуктивності, так і з рівнями збереження, допомагає прискорити продуктивність резервного копіювання та відновлення. З багаторівневим пристроєм, коли обсяг даних збільшується, пристрій можна масштабувати, а не просто збільшувати. Кожна багаторівнева система включає не лише диск, але й процесор, пам'ять і мережеві ресурси, щоб підтримувати фіксовану довжину вікна резервного копіювання, навіть коли обсяг даних зростає.

Потім кілька багаторівневих пристроїв працюють разом, щоб розподіляти робочі навантаження резервного копіювання, тож у міру зростання обсягу даних кожен новий пристрій автоматично підключається до існуючої мережевої топології резервного копіювання та бере на себе свою частку робочого навантаження резервного копіювання та дедуплікації. Якщо врахувати ці фактори, стає зрозуміло, що багаторівневе сховище є відповіддю на скорочення вікна резервного копіювання.

Резервне копіювання веб-сайту є необхідним кроком для всіх користу-

вачів. При використанні, для резервного копіювання файлів у локальну папку, FTP-клієнт може підсумувати процес у три основні кроки:

– отримати доступ і вибрати файли за допомогою FTP;
– створити або знайти каталог для резервного копіювання;
– завантажити файли веб-сайту на каталог, який створили або обрали.

Також є можливість використовувати FTP для резервного копіювання пошти або файлів бази даних. Програма резервного копіювання FTP забезпечує найбільш безпечний і надійний спосіб резервного копіювання даних. Зберігання файлів на віддаленому сервері резервного копіювання забезпечує захист від апаратних збоїв, стихійних лих та інших причин втрати даних. Для резервного копіювання на FTP потрібне програмне забезпечення віддаленого резервного копіювання, наприклад Handy Backup.

FTP використовується в основному для:

– резервного копіювання (FTP дозволяє користувачам створювати резервні копії даних і файлів з одного місця на безпечний сервер);
– дублювання (FTP є чудовим способом дублювання даних і файлів з однієї системи в іншу);
– завантаження даних (FTP часто використовується для доступу до послуг веб-хостингу та допомагає із завантаженням даних і файлів у віддалену систему).

Існує кілька різних типів FTP, залежно від потреб безпеки:

– анонімний FTP – це оригінальна і найпростіша форма FTP без шифрування, імен користувачів і паролів для передачі даних і файлів;
– FTP Secure (FTPS) – ця форма FTP використовує Transport Layer Security (TLS) під час підключення та є більш безпечним способом надсилання та отримання файлів із ідентифікаторами, паролями та сертифікатами для перевірки автентичності системи;
– безпечний FTP (SFTP) – SFTP забезпечує механізм у протоколі Secure Shell (SSH), використовуючи алгоритми та шифрування для захисту даних, а також ідентифікатори, паролі та ключі SHH для пе-

ревірки автентичності.

FTP і його вдосконалені версії, SFTP і FTPS, є одними з найстаріших типів онлайн-сховищ. Протокол FTP дозволяє передавати файли та папки до та з віддаленого сховища, доступного за попередньо визначеним іменем або IP-адресою. Завдяки популярності цієї технології можна легко знайти та завантажити безкоштовний інструмент резервного копіювання FTP. Порівняно з Handy Backup цим інструментам часто бракує універсальності, зручності, автоматизації чи безпеки, а можливо і усіх цих функцій разом. FTP існує з початку 1970-х років, а перша специфікація протоколу була використана в 1971 році. FTP швидко розвивався і до 1980 року була розгорнута версія протоколу RFC 765, а в 1985 році – оновлена версія RFC 959. До 1991 року це був найпоширеніший протокол передачі даних у світі, перш ніж наприкінці 1990-х із додаванням SSL з'явилося третє покоління протоколу FTPS [3].

1.1 Аналіз зручності використання резервного копіювання через FTP

Цілий ряд компаній використовує FTP як перевірений спосіб безпечної передачі файлів. Архітектори, наприклад, покладаються на FTP для надсилання та обміну кресленнями та діаграмами. Креслення систем автоматизованого проєктування (CAD) часто є великими файлами, тому швидкість FTP є ключовою перевагою. Масова передача файлів є ще однією ключовою перевагою протоколу FTP, на який друкарні покладаються, щоб надсилати великі файли для друку на затвердження клієнта. І веб-розробники також покладаються на FTP, коли переміщують файли зі свого комп'ютера на сервери розміщення веб-сайтів. Розширений механізм передачі резервного копіювання FTP. Handy Backup постачається з повнофункціональним FTP-клієнтом, який здатний передавати дані в будь-якому напрямку: віддалено зберігати резервні копії ПК або створювати резервні копії даних із FTP-сервера, або просто скопіювати файли з одного місця FTP в інше [3].

1.2 Аналіз сумісності програмного забезпечення резервного копіювання через FTP

Повна підтримка Unicode дозволяє створювати резервні копії файлів, названих японською, німецькою та іншими мовами. Розширені алгоритми дозволяють виконувати резервну роботу на FTP або синхронізувати дані з будь-якого типу FTP або навіть проксі-серверів SOCKS5, SOCKS4. FTPS (FTP через SSL) і SFTP (протокол передачі файлів SSH). У той час як версія Standard дозволяє автоматично створювати резервні копії лише на FTP, версії вищих версій (Professional і вище) підтримують захищені протоколи SFTP і FTPS.

Завдяки багатому набору функцій і зосередженості на взаємодії з користувачем Handy Backup широко визнана найпотужнішою утилітою резервного копіювання у своєму класі [1]. Простіше кажучи, FTP-сервери виконують 2 основні завдання: «Подати» та «Отримати». Є можливість розміщувати файли на FTP-сервері або отримувати файли з FTP-сервера. Якщо безпека не викликає занепокоєння, програмне забезпечення FTP Server є простим і недорогим способом досягти цього. Однак FTP не можна використовувати в таких ситуаціях як:

- якщо обробляються конфіденційні дані;
- якщо потрібно підтримувати відповідність, наприклад PCI-DSS, GDPR, ISO тощо;
- якщо потрібна підвищена безпека;
- якщо мається більше пересадок, ніж у є людино-годин.

У цьому випадку потрібно розглянути можливість керованої передачі файлів, яка підтримує безпеку та відповідність нормам, а також забезпечує автоматизацію та інтеграцію.

1.3 Огляд історії розробки FTP

FTP існує з початку 1970-х років, а перша специфікація протоколу була використана в 1971 році. Концепція FTP бере власний початок у квітні 1971 року, коли Абхай Бхушан вперше написав специфікацію протоколу передачі файлів, яку він опублікував у RFC 114.

У перші роки FTP працював на NCP, що означає програму керування мережею, і полегшував протокол стек на комп'ютерах, на яких розміщено ARPANET (попередник сучасного Інтернету). У 80-х FTP перейшов до протоколу керування передачею/протоколу Інтернету (TCP/IP), де він залишається й сьогодні.

FTP швидко розвивався, і до 1980 року була розгорнута версія протоколу RFC 765, а в 1985 році – оновлена версія RFC 959. До 1991 року це був найпоширеніший протокол передачі даних у світі, перш ніж наприкінці 1990-х із додаванням SSL з'явилося третє покоління протоколу FTPS. У 1990-х нові стандарти увімкнули дружній до брандмауера FTP (у 1994 році), запропонували розширення безпеки (1998) і додали підтримку IPv6, а також визначили нову версію пасивного режиму (1998). Як ви побачите нижче, FTP і його можливості пройшли довгий шлях у сучасну епоху [2].

1.4 Опис структури FTP

Найпростіше кажучи, FTP – це метод короткочасного з'єднання комп'ютерів, які називаються «клієнтами» та «серверами», для полегшення передачі файлів від одного до іншого (рис. 1). FTP-сервери схожі на веб-сервери, але вони відрізняються тим, що містять файли для завантаження, а не файли, які можна отримати як веб-сторінки через браузери. Веб-сайти працюють на HTTP/HTTPS, а не на FTP. Сервери FTP полегшують як завантаження «на» так і завантажуватися «з». Під час завантаження файли передаються з локальних комп'ютерів або серверів на віддалені сервери FTP. Під

час завантажень файли передаються з FTP-серверів на локальні сервери або комп’ютери.

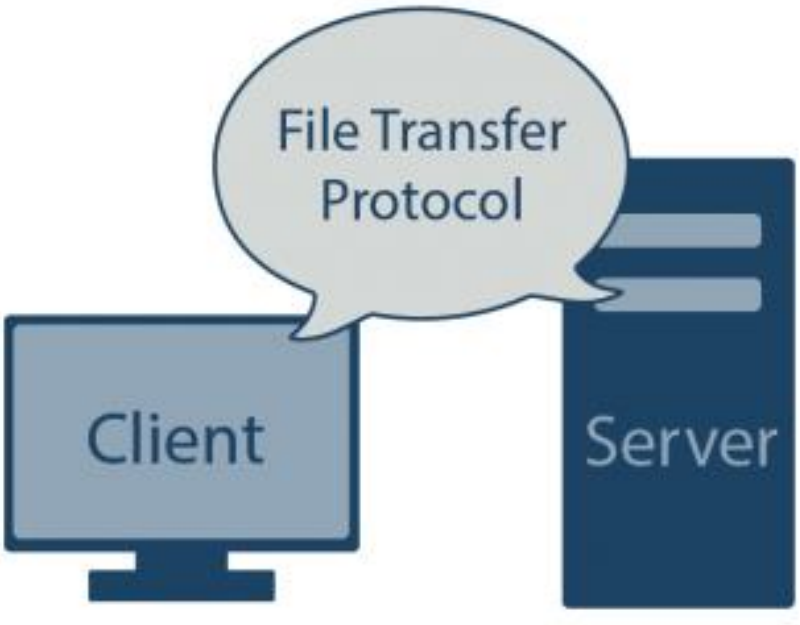

Рисунок 1 – Архітектура FTP

Метафорично кажучи, локальні системи можна розглядати як відправників листів і одержувачів листів. FTP як протокол був би схожий на поштову вантажівку, яка використовується для доставки листів. А FTP-сервер можна розглядати як поштове відділення або центр розподілу пошти. Наприклад, компанії А потрібно надіслати велику кількість файлів компанії В. Компанія А завантажить файли на FTP-сервер, а потім компанія В перенесе дані з FTP-сервера на власний комп’ютер.

Усі FTP-сервери мають адресу. Адреса може виглядати як веб-адреса (починається з ftp://) або більше схожа на рядок чисел, які представляють IP-адресу. Хоча деякі FTP-сервери не потребують облікових даних для входу, що називається «анонімним» з’єднанням, більшість FTP-серверів мають функції безпеки, які потребують імені користувача та пароля для доступу.

1.5 Аналіз різновидів застосунків для створення резервних копій на ftp-сервері

Існує кілька різних типів застосунків, які можна розробити для створення резервних копій даних на FTP-сервері. Нижче наведені деякі з них:

– клієнт FTP: цей тип застосунку дозволяє користувачам підключатися до FTP-сервера і передавати файли на нього, розроблений клієнт FTP може мати функції автоматичного резервного копіювання, що дозволяє користувачам налаштовувати регулярне копіювання даних з їхнього комп'ютера на FTP-сервер;
– резервне копіювання з синхронізацією: цей тип застосунку може відслідковувати зміни в файлах та папках на комп'ютері користувача і автоматично синхронізувати їх з FTP-сервером, це забезпечує постійну актуалізацію резервних копій на сервері;
– розробка на основі хмарних служб: деякі хмарні служби надають можливість підключитися до FTP-сервера для зберігання резервних копій, розробка застосунку, який інтегрується з такими хмарними службами, дозволить користувачам зручно створювати та керувати резервними копіями даних на FTP-сервері через інтерфейс хмарної платформи;
– автоматизований планувальник резервного копіювання: цей тип застосунку дозволяє користувачам налаштовувати планувальник для регулярного резервного копіювання даних на FTP-сервер, користувачі можуть встановлювати графік, параметри та вибирати файли та папки для резервного копіювання, автоматизований планувальник забезпечує надійну і регулярну створення резервних копій;
– backup-менеджер: цей тип застосунку надає повний набір функцій для створення та управління резервними копіями на FTP-сервері; він може мати інтерфейс, що дозволяє користувачам вибирати файли та папки для резервного копіювання, розкладувати резервне ко-

піювання, відновлювати дані з резервних копій та багато іншого.

Вибір конкретного типу застосунку залежить від потреб та вимог користувача. Деякі користувачі можуть використовувати Prefere клієнт FTP з функцією резервного копіювання, тоді як інші можуть шукати повніші функції, які надаються резервними менеджерами або хмарними службами.

Резервне копіювання даних є важливою практикою для забезпечення захисту і безпеки інформації:

- надійність даних;
- захист від випадковості;
- безпека від загроз;
- зручність відновлення;
- різні методи резервного копіювання.

До надійності даних можна віднести резервне копіювання, яке дозволяє створювати додаткові копії важливих даних, що в свою чергу дозволяє уникнути втрати інформації в разі аварій, видалення, пошкодження або злому даних. Застосування резервних копій забезпечує можливість відновити дані і продовжити роботу навіть у випадку непередбачуваних проблем.

Резервне копіювання є захистом від випадковості і незалежно від того, чи втратились дані через помилку користувача, технічний збій або злочинну діяльність, наявність резервних копій забезпечує можливість відновлення цих даних і продовження роботи без значних збоїв або затримок.

Резервне копіювання також є ефективним заходом для захисту даних від загроз безпеки, таких як віруси, шкідливі програми, кібератаки та викрадення даних і якщо виникає загроза або комп'ютер піддається атакам, можна відновити дані з резервних копій і запобігти серйозним наслідкам.

Резервне копіювання даних також спрощує процес відновлення інформації та застосування правильної стратегії резервного копіювання і збереження копій на безпечних носіях або в хмарних сховищах та дозволяє легко і швидко відновити дані, навіть якщо оригінальні файли стали недоступними.

На сьогоднішній день існує багато різних методів резервного копію-

вання, таких як повний, інкрементальний, диференційний, а також вибір правильного методу залежить від потреб і характеру даних. Комбінування різних методів може забезпечити ефективну та оптимальну стратегію резервного копіювання.

Узагальнюючи можна підкреслити, що резервне копіювання даних є важливою практикою для забезпечення безпеки, захисту від втрати та відновлення важливих даних. Вибір методу та розробка відповідного застосунку для резервного копіювання дозволяє забезпечити ефективну та надійну стратегію резервного копіювання даних.

2 ВИБІР ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ

Існує кілька способів створення віддаленої резервної копії FTP. Багато комерційних провайдерів веб-хостингу пропонують резервне копіювання FTP-сервера, коли користувач/замовник підписується на їхні послуги веб-хостингу, створюючи власну систему для ручного перенесення та автоматизовані системи, які є останнім із тріо параметри, які можна використовувати для цієї форми захисту даних.

Кожен із них має свої переваги та недоліки, але варіант веб-хостингу приховує у собі найбільшу небезпеку, оскільки хоча вони можуть дозволити зберігати резервні копії файлів веб-хостингу, не всі ці постачальники можуть дозволити зберігати резервні копії локальних файлів. Крім того, відомо, що деякі з цих постачальників послуг видаляють резервні копії без попереднього повідомлення про те, що це може статися, тому інформацію про політику кожного веб-хосту потрібно перевірити у постачальника послуг, перш ніж використовувати їхні сервери для резервного копіювання FTP.

Найпоширенішим варіантом створення резервної копії FTP, яка виконується вручну, є Filezilla, і це галузевий стандарт програмного забезпечення для створення завдань віддаленого резервного копіювання FTP. Filezilla працює в операційних системах Windows, Mac OS X і Linux і має зручний інтерфейс, який дозволяє легко відстежувати збережені FTP-сервери та пов’язані з ними дані з’єднань, деталі локальних і віддалених каталогів і поточні передачі. Також слід додати до цього той факт, що Filezilla підтримує протоколи SFTP (Secure File Transfer Protocol) і FTPS (File Transfer Protocol Secure) разом із FTP, і розробник зрозуміє, чому це найкращий на ринку безкоштовний варіант із відкритим кодом для віддаленого резервного копіювання.

Останнім із трьох методів створення резервної копії FTP є автоматизований варіант, і хоча цей варіант дорожчий, ніж спосіб створення резервної копії Filezilla, завдяки автоматизації завдань і той факт, що ліцензія охоплює можливість запуску кількох резервних копій у більш ніж одному місці одно-

часно. Існує багато різних частин програмного забезпечення, які можна придбати та завантажити, але далі більшу увагу приділено програмному забезпеченні резервного копіювання Iperius.

Це програмне забезпечення є одним із найдешевших програм FTP, доступних зараз на ринку, а також одним із найширших і найпотужніших у створенні резервних копій FTP. Окрім перерахованих раніше можливостей завантажувати та оновлювати веб-сайти та веб-застосунки, його можна легко налаштувати для створення одночасних віддалених резервних копій FTP і SFTP, а також дозволити користувачам виконувати інкрементне резервне копіювання після початкового віддаленого резервного копіювання.

Програмне забезпечення для резервного копіювання Iperius також підтримує стиснення zip і 256-бітове шифрування AES, забезпечуючи таким чином максимальний рівень безпеки, залишаючи користувачів на 100% у безпеці, знаючи, що їхні дані в цілковитій безпеці за допомогою шифрування військового рівня.

Нижче представлено інформацію щодо аналізу найкращих серверних програм резервного копіювання FTP. Резервне копіювання даних є важливим, і багато користувачів віддають перевагу резервному копіюванню своїх файлів через FTP. Щоб вирішити різні проблеми з ПК, рекомендовано Restoro PC Repair Tool.

Дане програмне забезпечення виправить поширені комп’ютерні помилки, захистить від втрати файлів, зловмисного програмного забезпечення, збою апаратного забезпечення та оптимізує ПК для максимальної продуктивності. Вирішіть проблеми з ПК і видалить віруси в 3 прості кроки. Резервне копіювання файлів є важливим, і багато користувачів вирішують робити резервні копії своїх файлів онлайн. Є багато інструментів, які можуть це зробити, але найзручнішими є ті, що підтримують FTP.

2.1 Опис програмного середовища Handy Backup

Handy Backup – це просте програмне забезпечення для резервного копіювання FTP-сервера, яке може створювати резервні копії будь-яких файлів і навіть підтримує резервне копіювання електронної пошти (рис. 2).

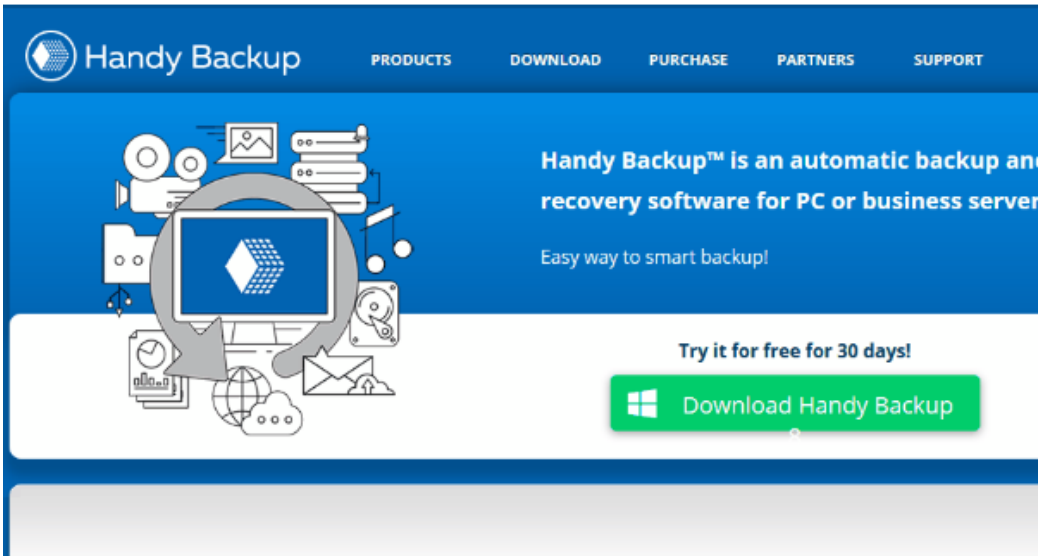

Рисунок 2 – Головне вікно Handy Backup

За допомогою цього програмного забезпечення є можливість створювати резервні копії файлів на локальний жорсткий диск, зовнішні жорсткі диски та хмарне сховище. Існує навіть підтримка резервного копіювання по FTP, резервного копіювання на NAS і підтримка протоколу WebDAV. Handy Backup дозволяє планувати, і він може створювати повні, інкрементні або диференціальні резервні копії. Звичайно, можна стискати дані в архівах ZIP, а також є підтримка надійного шифрування. Варто зазначити, що програмне забезпечення має журнал активності, тому є можливість контролювати свої резервні копії, а також є підтримка сповіщень електронною поштою.

Handy Backup пропонує чудові функції, і це ідеальний інструмент, якщо потрібно створити резервну копію комп’ютера через FTP. Воно підтримує різні типи джерел даних, такі як локальні пристрої, мережеві ресурси, FTP-

сервери, хмари, а також дозволяє виконувати резервне копіювання на зовнішні пристрої, такі як USB-накопичувачі. Handy Backup також пропонує планувальник завдань, можливість створення і налаштування різних профілів резервного копіювання та опції стиснення та шифрування даних.

Функції Handy Backup:

- підтримка FTP і хмарних резервних копій;
- підтримка локального, зовнішнього та NAS резервного копіювання;
- підтримка протоколу WebDAV;
- повне, інкрементне та диференціальне резервне копіювання;
- стиснення та шифрування резервних копій.

Handy Backup відомий своєю надійністю та стабільністю. Він надає можливість перевірки цілісності резервних копій, а також опції відновлення файлів і даних у разі втрати або пошкодження. Крім того, Handy Backup забезпечує захист від несанкціонованого доступу до даних за допомогою опцій шифрування та паролів.

Handy Backup надає хорошу технічну підтримку для своїх користувачів. Вони пропонують документацію, онлайн-поради, часті питання та контактну інформацію для зв'язку з командою підтримки. Користувачі можуть звертатися за допомогою у разі виникнення проблем або запитань. Загалом, Handy Backup є потужним програмним середовищем для резервного копіювання та відновлення даних, яке має широкий функціонал, зручний інтерфейс та надійність. Враховуючи ці аспекти, воно може бути цікавим варіантом для користувачів, які шукають надійне рішення для збереження своїх даних.

2.2 Опис програмного середовища Iperius Backup

Iperius Backup – це програма для резервного копіювання, яка також дозволяє виконувати резервне копіювання на сервери SFTP, Amazon S3, Microsoft Azure Storage, Google Drive, Dropbox і OneDrive. Програмне забезпечення підтримує додаткове резервне копіювання та синхронізацію FTP, а

також забезпечує повну сумісність з будь-яким сервером FTP/FTPS/SFTP.

Застосунок пропонує широкий спектр можливостей для резервного копіювання та відновлення даних. Воно підтримує резервне копіювання файлів, папок, дисків, баз даних, віртуальних машин і хмарних служб. Iperius Backup пропонує чудовий захист, а завдяки 256-бітному шифруванню на стороні клієнта SSL і AES можна бути впевненим, що файли завжди в безпеці (рис. 3).

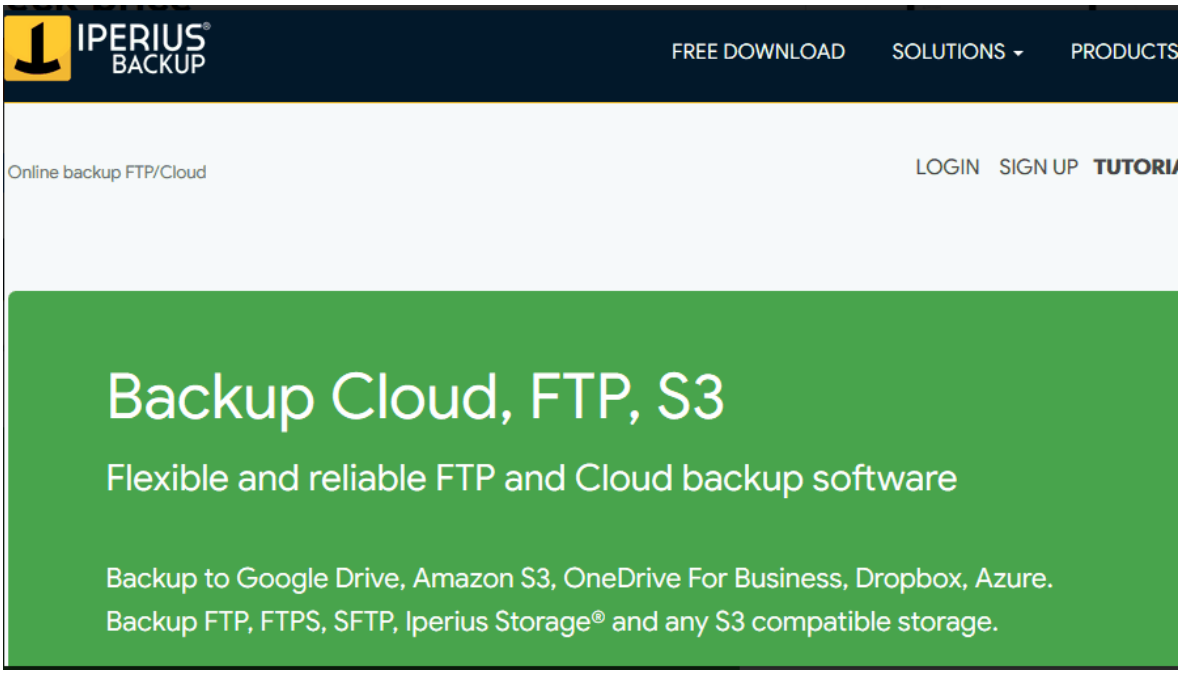

Рисунок 3 – Головне вікно Iperius Backup

Програмне забезпечення також дозволяє копіювати файли, які використовуються або заблоковані, і навіть доступна резервна копія FTP на NAS. Iperius Backup має простий і зрозумілий інтерфейс користувача, який дозволяє легко налаштовувати та керувати процесом резервного копіювання. Інтерфейс має інтуїтивно зрозумілу структуру, що спрощує навігацію та налаштування резервних копій. Користувачі можуть легко створювати розклади резервного копіювання та виконувати миттєві копії файлів та папок.

Iperius Backup відомий своєю надійністю та стабільністю. Воно надає можливість перевірки цілісності резервних копій, а також виключає можливість пошкодження або втрати даних під час процесу копіювання. Iperius

Backup також пропонує захист даних шляхом шифрування резервних копій та налаштування паролів доступу. Загалом, Iperius Backup пропонує чудові функції, і він буде ідеальним, якщо необхідне програмне забезпечення для резервного копіювання FTP-сервера.

Функції Iperius Backup:

- підтримка резервного копіювання та синхронізації по FTP;
- сумісний з будь-яким сервером FTP/FTPS/SFTP;
- підтримка Amazon S3, Microsoft Azure Storage, Google Drive, DropBox і резервного копіювання OneDrive;
- 256-бітне шифрування SSL і AES;
- FTP до NAS.

Iperius Backup надає технічну підтримку для своїх користувачів. Вони пропонують документацію, часті питання, онлайн-поради та можливість зв'язку з командою підтримки. Користувачі можуть звертатися за допомогою, якщо виникнуть проблеми або питання.

Iperius Backup є програмним середовищем з багатофункціональністю, простим інтерфейсом та надійністю. Враховуючи ці аспекти, воно може бути привабливим варіантом для користувачів, які шукають рішення для резервного копіювання та відновлення даних.

2.3 Опис програмного середовища AceBackup

Ще одне програмне забезпечення для резервного копіювання з підтримкою FTP – AceBackup. Програмне забезпечення пропонує резервне копіювання на FTP-сервери, що може бути надзвичайно корисним. Варто зазначити, що програма підтримує шифрування, тому всі дані залишаться в безпеці. Що стосується підтримуваних стандартів, він підтримує Rijndael 128/192/256, Blowfish 128 і Triple-DES 128.

Програмне забезпечення підтримує планування, і при необхідності можна створювати резервні копії даних через певні проміжки часу, щоб забез-

печити безпечне резервне копіювання всіх файлів. Користувачі можуть звертатися за допомогою через електронну пошту або форуми для отримання відповідей на власні запитання та вирішення проблем.

Деякі проблеми з ПК важко вирішити, особливо коли мова йде про пошкоджені сховища або відсутні файли Windows. Якщо виникли проблеми з виправленням помилки, то система може бути частково зламана. Важливо зазначити, що програмне забезпечення підтримує стиснення ZIP, і використовуючи його, заощаджується дорогоцінний простір для зберігання. AceBackup – це просте у використанні програмне забезпечення для резервного копіювання, і якщо потрібне просте програмне забезпечення з підтримкою FTP, це може бути правильним інструментом.

Функції AceBackup:

- повністю безкоштовно;
- резервне копіювання FTP ;
- шифрування Rijndael 128/192/256, Blowfish 128 і Triple-DES 128 ;
- підтримка планування резервного копіювання ;
- вбудоване стиснення ZIP.

AceBackup є програмним середовищем зі зручним інтерфейсом, надійною функціональністю та підтримкою різних джерел даних. Враховуючи ці аспекти, воно може бути привабливим варіантом для користувачів, які шукають програму для резервного копіювання та відновлення власних даних.

2.4 Опис програмного середовища Leo Backup

Якщо потрібна програма для резервного копіювання з підтримкою резервного копіювання FTP, то Leo Backup (рис. 4) може бути саме тою програмою, що потрібна. Програмне забезпечення підтримує резервне копіювання на локальні диски, локальні мережі та через FTP. Що стосується резервного копіювання, є підтримка Amazon S3, Google Drive і One Drive. Варто зазначити, що файли завжди будуть у безпеці завдяки шифруванню AES-256.

Рисунок 4 – Головне вікно Leo Backup

Також є вбудована підтримка стиснення Zip64, тому резервні копії не займатимуть надто багато місця. Щоб забезпечити безпечне резервне копіювання даних, існує повна підтримка планування резервного копіювання. Щодо розширених функцій, є резервне копіювання бази даних SQL, а також резервне копіювання реєстру, але необхідно мати на увазі, що деякі з цих функцій доступні лише у версіях Pro та Enterprise.

Функції Leo Backup:

– резервне копіювання FTP;
– сповіщення електронною поштою;
– планування резервного копіювання;
– 256-бітне шифрування та стиснення;
– резервне копіювання хмарного сховища;
– резервне копіювання бази даних (у версії Enterprise).

2.5 Опис програмного середовища Duplicati 2.0

Ще одне програмне забезпечення для резервного копіювання FTP для ПК – Duplicati 2.0 (рис. 5). Програмне забезпечення має шифрування AES-256, тому файли та папки завжди будуть захищені.

Рисунок 5 – Головне вікно Duplicati 2.0

Програмне забезпечення розроблено для економії місця, тому воно повністю підтримує інкрементне резервне копіювання та дедуплікацію даних. Duplicati 2.0 має веб-інтерфейс, тому його можна запускати в будь-якому веб-браузері, і можна отримати доступ до нього з будь-якого місця. Застосунок підтримує FTP, SSH і WebDAV, а також може працювати з такими популярними сервісами, як Backblaze B2, Tardigrade, Microsoft OneDrive, Amazon S3, Google Drive, BOC, Mega та іншими.

Програмне забезпечення оптимізоване для онлайн-резервного копіювання, і навіть якщо процес резервного копіювання перервано, його можна легко відновити в будь-який час. Duplicati 2.0 повністю оптимізовано для онлайн-резервного копіювання, а завдяки підтримці FTP, SSH і WebDAV стане ідеальним інструментом для професіоналів.

Функції Duplicati 2.0:

- повністю безкоштовний і відкритий код;
- підтримує протоколи FTP, SSH і WebDAV;
- сумісний із Microsoft OneDrive, Amazon S3, Google Drive та багатьма іншими службами;
- шифрування AES-256 і GPG;
- веб-інтерфейс.

Вибираючи програмне забезпечення FTP, слід врахувати такі фактори, як безпека даних, обмеження доступу, підтримка хмари, відповідність нормативним вимогам тощо. Варто перевірити наявність детальних елементів керування, шифрування, підтримуваних протоколів та іншої відповідності вимогам безпеки з точки зору безпеки.

3 ПРОЄКТУВАННЯ ДЕСКТОПНОГО ЗАСТОСУНКУ

3.1 Опис типів резервних копій

Існує кілька типів резервних копій, які можуть бути створені для забезпечення захисту даних. Цей список не є вичерпним, існують і інші типи резервних копій, які можуть бути використані в залежності від потреб і специфіки системи. Важливо обрати тип копіювання, який найкраще відповідає потребам з точки зору обсягу даних, доступності та часу відновлення.

3.1.1 Опис повного резервного копіювання

Повне резервне копіювання (Full Backup) є простим типом резервного копіювання, оскільки всі дані копіюються в інше місце. Оскільки для всіх даних створюється резервна копія, процес відновлення відбувається відносно швидко.

Однак створення повної резервної копії вимагає більше часу, оскільки всі дані потрібно скопіювати, а також займає багато пам’яті. Наприклад, якщощ99 є 2 ТБ даних і виконується повне резервне копіювання двічі на тиждень, знадобиться 4 ТБ додаткового місця для зберігання для одного тижня резервного копіювання (рис. 6). Але він також має перевагу в тому, що має простий у використанні контролю версій і дуже простий пошук файлів. А оскільки кожна повна резервна копія не залежить від будь-яких попередніх резервних копій і містить усі дані, її дуже легко відновити.

Повна резервна копія забезпечує повну ідентичність з оригінальними даними, але може займати багато місця на сховищі та вимагати більше часу для створення та відновлення. Забезпечення безпеки під час передачі даних може включати використання методів аутентифікації, таких як пароль, ключ або сертифікат, для забезпечення ідентифікації та авторизації користувача.

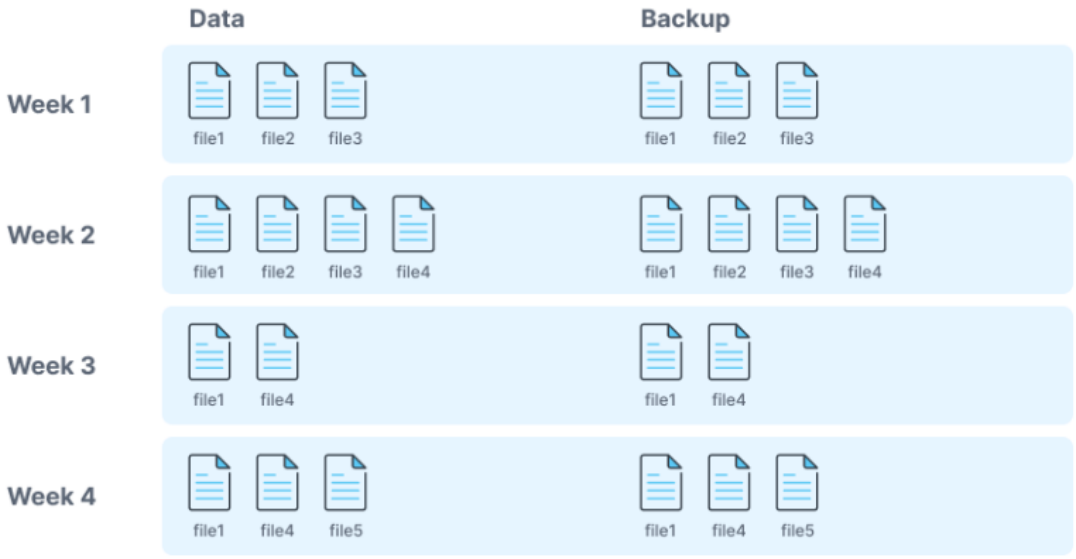

Рисунок 6 – Схема повного резервного копіювання

3.1.2 Опис інкрементного резервного копіювання

Тут буде скопійовано лише дані, змінені з моменту останнього резервного копіювання. Остання резервна копія може бути останньою інкрементною резервною (Incremental Backup) копією або повною резервною копією. Через це виконання інкрементного резервного копіювання відбувається швидше, ніж повне резервне копіювання, і вимагає набагато менше місця для зберігання (рис. 7).

Однак процес відновлення інкрементної резервної копії є більш складним і трудомістким у порівнянні з повною резервною копією, оскільки потрібно отримати доступ до кількох резервних копій і відновити їх. Інкрементальні копії зазвичай займають менше місця та потребують менше часу для створення та відновлення, але процес відновлення може бути трохи складнішим, оскільки потрібно відновити останню повну копію, а також всі інкрементальні зміни.

Важливо мати історію резервних копій, яка вказує, коли копія була створена, де вона зберігається та інші релевантні деталі. Це допомагає відстежувати різні версії даних та полегшує відновлення попередніх станів.

Рисунок 7 – Схема інкрементального резервного копіювання

3.1.3 Опис диференціального резервного копіювання

Цей тип резервного копіювання знаходиться між інкрементним і повним резервним копіюванням: диференціальне резервне копіювання (Differential Backup) копіює всі дані, які змінилися з часу останнього повного резервного копіювання. Щоб відновити дані, потрібна лише остання повна резервна копія та остання диференціальна резервна копія (рис. 8). Це призводить до швидшого часу відновлення порівняно з додатковим резервним копіюванням, а також заощаджує місце для зберігання порівняно з повним резервним копіюванням.

Цей тип копії також включає тільки зміни, зроблені після останньої повної копії, але відмінність полягає в тому, що диференціальні копії включають всі зміни з моменту останньої повної копії, а не лише з останньої інкрементальної копії. Відновлення диференціальних копій може бути швидшим, оскільки потрібно відновити лише останню повну копію та останню диференціальну копію.

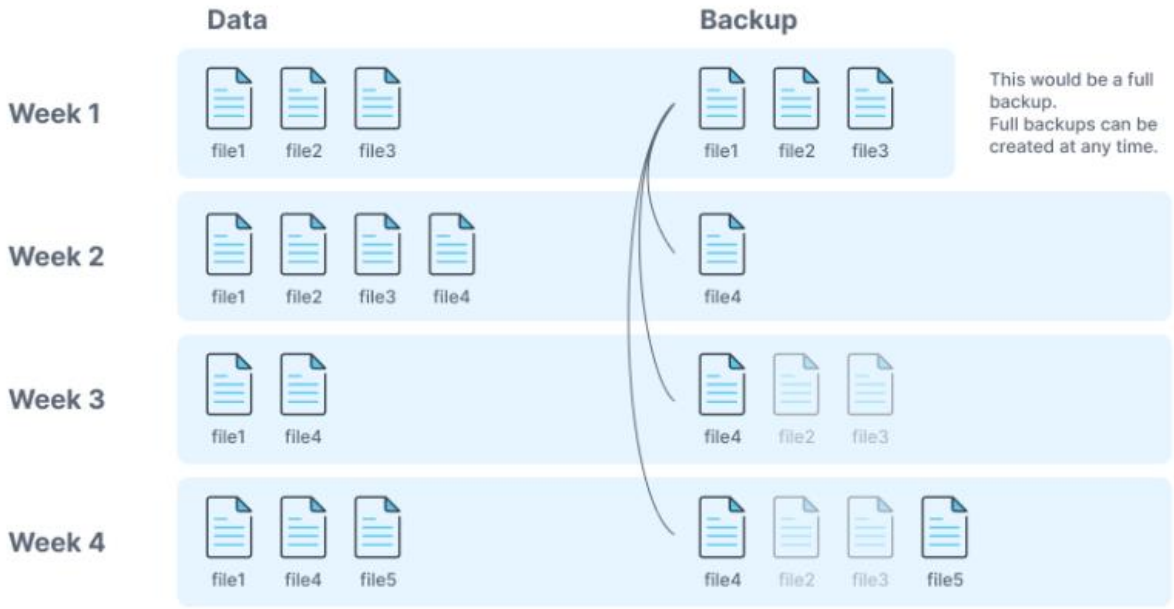

Рисунок 8 – Схема диференціального резерву

3.2 Опис правила резервного копіювання 3-2-1

«Правило резервного копіювання 3-2-1» – це означає, що є принаймні три копії даних у двох різних місцях і одна копія за межами сайту, тобто:

- локальний / на місці (наприклад, головний сервер);
- локальний / на місці – але на іншому пристрої, ніж перша копія (наприклад, зовнішній жорсткий диск);
- поза сайтом (наприклад, зберігання об’єктів).

Резервні копії на місці дозволяють швидко шукати резервні копії, оскільки не потрібно їх спочатку завантажувати. Це також скорочує час, необхідний для відновлення даних. Зовнішня резервна копія існує, щоб забезпечити доступність, якщо не можна отримати доступ до резервних копій на місці, наприклад, через збій у центрі обробки даних.

3.3 Опис систем RAID і Snapshots

Важливо зазначити, що RAID і Snapshots не є стратегіями резервного копіювання та не замінюють згадані вище типи резервного копіювання. Натомість система RAID забезпечує резервування у разі збою диска та запобігає простою. Однак у разі інших типів втрати даних (наприклад, атаки програм-вимагачів) розробник не може розраховувати на RAID-систему, і натомість потрібна зовнішня резервна копія. Це також означає, що користувачі повинні використовувати RAID і згадані стратегії резервного копіювання разом, щоб запобігти простою та втраті даних.

Знімки фіксують поточний стан сервера та вказують точку, до якої можна повернутися. Якщо щось спробувати, але це не спрацює, можна повернутися до знімка та спробувати ще раз. Але миттєві знімки залежать від хост-машини і, наприклад, не можуть використовуватися для відновлення на іншому сервері.

3.4 Опис параметрів зберігання резервних копій

Параметри зберігання резервних копій можуть варіюватися в залежності від потреб користувача та конкретних вимог. Ці параметри зберігання резервних копій можуть бути налаштовані відповідно до потреб та умов. Важливо регулярно переглядати та оновлювати свої стратегії збереження для забезпечення безпеки та доступності даних. Нижче описано кілька основних параметрів, які можна врахувати при збереженні резервних копій:

3.4.1 Аналіз зовнішнього жорсткового диску

Для резервного копіювання на місці, слід використовувати щось на зразок зовнішніх жорстких дисків. Якщо є швидкі жорсткі диски, процес резервного копіювання набагато швидший, ніж якщо потрібно буде надіслати дані

на хмарний сервер на іншому кінці світу. І це також стосується відновлення даних, не потрібно завантажувати резервну копію, просто скопіювати файли з резервного диска на основний диск. Але потрібно мати на увазі, про відповідальність за безпеку та доступність жорстких дисків, і необхідність запобігати несанкціонованому доступу до цього жорсткого диска.

3.4.2 Опис FTP

FTP є дуже популярним рішенням, окільки FTP має оренду місця для зберігання в хмарі та завантаження туди копії даних. Наприклад, є можливість орендувати старе сховище FTP, яке можна використовувати як зовнішній жорсткий диск.

3.4.3 Аналіз зберігання об'єктів

Якщо шукати економічно ефективний спосіб зберігання великих обсягів даних (зазвичай у хмарі) – Object Storage, ймовірно, найкращий вибір. Великою перевагою Object Storage є його здатність масштабуватися майже без обмежень за низькою ціною. Однією з причин, чому Object Storage може бути таким рентабельним, є те, що він не призначений для обробки даних, які сильно змінюються. Але оскільки зазвичай не змінюються резервні копії після їх збереження, це не є проблемою та приносить перевагу наявності недорогого рішення для зберігання великих обсягів даних.

За допомогою чудових інструментів, таких як rclone, процес переміщення резервних копій у Object Storage може бути таким же простим, як і переміщення їх на Диск Google або інший FTP-сервер.

Користувач може переглянути статтю про rclone і Object Storage на офіційному сайті. Contabo Object Storage пропонує всеохоплююче (необмежену безкоштовну передачу) Object Storage з простою та передбачуваною моделлю ціноутворення та S3-сумісним API, щоб полегшити міграцію. Він

дає можливість перемістити зовнішні резервні копії в Contabo Object Storage сьогодні та заощаджуйте принаймні 7,5 разів щомісяця [4].

3.5 Опис переваг FTP і пов’язаних протоколів

FTP і більш безпечні протоколи, згадані в розділі вище, стали широко поширеними для передачі файлів у 21 столітті. Оскільки переваги використання FTP добре відомі та широко оцінені окремими особами та організаціями, яким потрібно безпечно передавати конфіденційні файли. вимоги до передачі файлів для багатьох компаній виходять за рамки надсилання одного документа Microsoft Word [5].

FTP спрощує передачу великих розмірів і вимагає вищої швидкості передачі файлів. FTP дозволяє надсилати сотні мегабайт даних одночасно. Загалом, FTP та його пов'язані протоколи мають кілька переваг, таких як надійність, широке поширення, підтримка аутентифікації, гнучкість, підтримка захисту та простота використання. Вони залишаються популярними засобами для передачі файлів у мережі, зокрема для створення резервних копій на FTP-сервері.

3.5.1 Опис переваги ємності

Користувач рано чи піздно потрапляє в ситуацію, коли великий файл просто не можна прикріпити та надіслати електронною поштою. FTP вирішує цю проблему, полегшуючи передачу великих окремих файлів, а також великої кількості менших файлів. Якщо вибрати метод передачі файлів, який не має необхідної ємності, можна побачити, що передавання та підключення часто не вдається. Звичайно, це дуже неефективно для будь-якої організації, якій потрібно регулярно передавати великі файли. FTP є одним з найбільш поширених протоколів для передачі файлів у мережі. Він підтримується багатьма операційними системами та програмними засобами, що робить його

універсальним і доступним для багатьох користувачів. FTP дозволяє передавати великі файли, масштабувати обробку даних, працювати на різних платформах, бути легким у використанні, керувати доступом до файлів та підтримувати розширення для поліпшення безпеки та функціональності.

3.5.2 Опис переваги безпеки

Незважаючи на те, що FTP, задуманий у 1971 році, мав уразливість у безпеці, зараз існують зашифровані протоколи передачі файлів, які надають поверх FTP захист, необхідний для конфіденційних файлів. Зокрема, FTPS і SFTP є варіантами для організацій, яким потрібні високобезпечні можливості передачі файлів. FTP та його варіації забезпечують надійну передачу файлів через мережу. Вони використовують механізми перевірки цілісності даних та перезапуску передачі в разі виникнення помилок. Це дозволяє гарантувати, що файли будуть передані повністю та без пошкоджень.

3.5.3 Опис переваги контролю

Більшість сучасних провайдерів FTP пропонують адміністративні інформаційні панелі, які дають користувачам високий рівень контролю над своїми файлами. Ці елементи керування дозволяють адміністраторам надавати дозволи кожному користувачеві завантажувати, ділитися, редагувати та навіть видаляти файли, що зберігаються на FTP-серверах. FTPS та SFTP, які є розширеннями FTP, надають додатковий рівень безпеки за допомогою шифрування даних під час передачі. Це забезпечує конфіденційність та захист від перехоплення чутливої інформації. FTP може забезпечити контроль доступу до файлів та папок на сервері. Адміністратор може налаштувати права доступу для користувачів, встановити обмеження на читання, запис або виконання файлів, що забезпечує безпеку та конфіденційність даних.

3.5.4 Опис переваги ефективності

Коли користувач визначає та запроваджує правильне рішення FTP для організації, можна створити високоефективний робочий процес передачі файлів, який зрозуміють і використовуватимуть усі члени команди. Знову ж таки, багато організацій регулярно обробляють великі та конфіденційні файли. Використовуючи FTP для створення ефективного робочого процесу, ці організації можуть рухатися швидше, робити більше та загалом покращувати свою діяльність. FTP має простий та зрозумілий для багатьох користувачів інтерфейс. Клієнти FTP, які надають графічний інтерфейс, роблять процес передачі файлів легким та зручним.

3.5.5 Опис переваги надмірності

Найкращі провайдери FTP також забезпечують регулярне резервне копіювання серверів, щоб конфіденційні файли завжди залишалися в безпеці в разі стихійного лиха чи іншого інциденту, який скомпрометує один сервер. Сервери резервного копіювання FTP зазвичай розташовані в абсолютно різних географічних регіонах. Це гарантує відсутність перерв у доступі організації до їхніх файлів.

FTP-сервери можуть бути налаштовані для роботи з великою кількістю файлів і обробляти багато одночасних підключень. Це дозволяє ефективно управляти великим обсягом даних та забезпечити швидку передачу файлів.

3.5.6 Опис переваги автоматизації

Автоматизація, ймовірно, є найбільшою перевагою використання цих протоколів. Передачі можна легко налаштувати та запланувати без втручання людини. Наприклад, можна синхронізувати локальні та віддалені папки кожні 5 хвилин цілодобово.

FTP-сервери часто надають можливість аутентифікації користувачів, що дозволяє обмежувати доступ до файлів лише авторизованим особам. Це забезпечує певний рівень безпеки та контролю над передачею файлів.

FTP є стандартним протоколом, який підтримується багатьма операційними системами, програмами та мережевими пристроями. Це означає, що файли, створені на одній платформі, можуть бути передані та отримані на іншій платформі без проблем.

3.6 Аналіз розвитку FTP

Світ FTP великий і може бути складним, якщо спробувати створити та впровадити власну систему. У FTP зараз надається низка різних варіантів для організацій, яким потрібно зберігати та обмінюватися конфіденційними файлами. FTP є одним з найстаріших та найпоширеніших протоколів передачі файлів у мережі. Незважаючи на свою широку популярність, з'явилися нові технології та протоколи, які пропонують більш безпечну та ефективну передачу даних [6].

Одним з прикладів є SSH File Transfer Protocol (SFTP), який використовує шифрування для захисту даних під час передачі. SFTP набуває все більшої популярності, оскільки він забезпечує вищий рівень безпеки порівняно з FTP.

Також, зростає популярність хмарних сховищ та файлових обмінників, які надають зручні інтерфейси та розширені функціональні можливості для зберігання та передачі файлів. Вони часто використовують власні протоколи або комбінацію різних протоколів для забезпечення безпеки та ефективності передачі даних.

Однак, FTP все ще залишається популярним і використовується у деяких випадках, особливо там, де потрібна простота та швидкість передачі файлів без складностей шифрування або інших додаткових функцій.

Загалом, майбутнє FTP може полягати в більшому застосуванні безпе-

чних варіацій, таких як SFTP, а також в інтеграції з іншими сучасними технологіями передачі файлів, такими як хмарні сховища та файлові обмінники.

Необхідно також враховувати, що FTP має свої обмеження, зокрема відсутність шифрування під час передачі даних, що може становити ризик з точки зору безпеки. Однак, в багатьох випадках, коли потрібна простота та швидкість передачі файлів, FTP може бути ефективним рішенням.

3.7 Опис розробки бази даних: проєктування та реалізація

Створення бази даних (БД) для є важливим кроком у розробці десктопного застосунку для створення backup даних на ftp-сервері. Зв'язок між таблицею "FTP-сервери" і таблицею "Завдання бекапу": у таблиці "Завдання бекапу" є стовпець "FTP-сервер_ID", який є зовнішнім ключем, посилаючимся на стовпець "ID" таблиці "FTP-сервери". Цей зв'язок дозволяє зв'язати кожне завдання бекапу з відповідним FTP-сервером, на який потрібно зберігати бекап. Зв'язок між таблицею "Завдання бекапу" і таблицею "Журнал бекапів":у таблиці "Журнал бекапів" є стовпець "Завдання_бекапу_ID", який є зовнішнім ключем, посилаючимся на стовпець "ID" таблиці "Завдання бекапу". Цей зв'язок дозволяє зв'язати кожен запис журналу бекапу з відповідним завданням бекапу, для якого було виконано бекап. Зв'язок між таблицею "Завдання бекапу" і таблицею "Користувачі" у таблиці "Завдання бекапу" є стовпець "Користувач_ID", який є зовнішнім ключем, посилаючимся на стовпець "ID" таблиці "Користувачі". Цей зв'язок дозволяє зв'язати кожне завдання бекапу з відповідним користувачем, який створив це завдання.Зв'язок між таблицею "Користувачі" і таблицею "Ролі користувачів": у таблиці "Користувачі" є стовпець "Роль_ID", який є зовнішнім ключем, посилаючимся на стовпець "ID" таблиці "Ролі користувачів". Цей зв'язок дозволяє зв'язати кожного користувача з його роллю в системі (наприклад, адміністратор, звичайний користувач тощо). Зв'язок між таблицею "Журнал бекапів" і таблицею "Статуси бекапів": у таблиці "Журнал бекапів" є стовпець "Статус_бекапу_ID", який є

зовнішнім ключем, посилаючимся на стовпець "ID" таблиці "Статуси бекапів". Цей зв'язок дозволяє зв'язати кожен запис журналу бекапу з його статусом (наприклад, успішний, неуспішний тощо). Схема бази даних представлена на рис. 9.

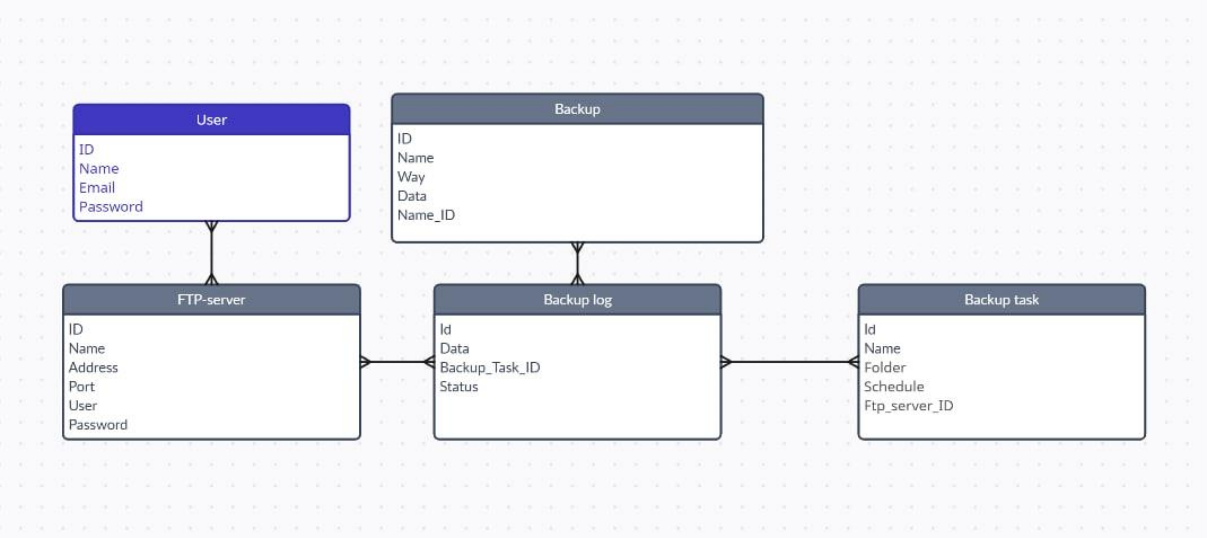

Рисунок 9 – Схема аналізу аналогів

Ця схема бази даних дозволить зберігати і керувати інформацією про FTP-сервери, завдання бекапу та журнал виконаних бекапів.

3.8 Аналіз аналогічних рішень та порівняння

У моїй роботі я вивчав різні аспекти резервного копіювання даних, починаючи з аналізу проблеми, огляду існуючих програмних середовищ і основних характеристик резервних копій. Далі, я розробив десктопний застосунок, який дозволяє користувачам легко і зручно створювати резервні копії даних на FTP-сервері.

Під час розробки застосунку, я використав різні етапи та технології, щоб забезпечити ефективність та надійність процесу. Мої дослідження та робота показали, що створений десктопний застосунок забезпечує зручний і

швидкий спосіб створення резервних копій даних, а також забезпечує їх безпеку під час передачі на FTP-сервер.

В результаті мого дипломного дослідження, я прийшов до важливих висновків щодо ефективності та важливості резервного копіювання даних на FTP-сервері. Цей десктопний застосунок може бути цінним інструментом для багатьох організацій, які прагнуть зберегти свої дані в безпеці.

Десктопні застосунки для резервного копіювання на FTP-сервери надають зручність та контроль користувачам, оскільки вони можуть локально керувати процесом створення, завантаження та відновлення резервних копій. Це дає користувачам більшу гнучкість і можливість використовувати FTP-сервери своєї власної вибору [7].

Незважаючи на те, що існують інші протоколи передачі файлів, такі як SFTP і FTPS, які надають більшу безпеку за допомогою шифрування, FTP залишається популярним і широко використовується у багатьох сценаріях. Тому, десктопні застосунки для резервного копіювання на FTP-сервери можуть продовжувати бути актуальними для тих, хто використовує FTP-сервери для збереження своїх даних. Приклад аналогів (рис. 10).

Програма	Тип програми	Протоколи	Інтерфейс	Функціонал
FileZilla	Безкоштовна з відкритим вихідним кодом	FTP, FTPS, SFTP	Простий інтуїтивний	Створення резервних копій, розкладування резервного копіювання
WinSCP	Безкоштовна з відкритим вихідним кодом	FTP, FTPS, SCP SFTP,	Двохпанельний з підтримкою перетягування	Створення резервних копій, автоматична синхронізація, розкладування резервного копіювання
Cyberduck	Комерційна	FTP, FTPS, SFTP, WebDAV	Інтуїтивний з підтримкою перетягування	Створення резервних копій, автоматична синхронізація, розкладування резервного копіювання
GoodSync	Комерційна	FTP, FTPS, SFTP	Розширений з підтримкою перетягування	Створення резервних копій, автоматична синхронізація, розкладування резервного копіювання
Drops	Комерційна	FTP, FTPS, SFTP	Стильний інтуїтивний перетягування	Створення резервних копій, автоматична синхронізація, розкладування резервного копіювання
FileZilla Server	Безкоштовна з відкритим вихідним кодом	FTP, FTPS	Простий інтерфейс	Створення FTP-сервера для збереження резервних копій, керування доступом та налаштування серверних параметрів
Salamander FTP	Комерційна	FTP, FTPS	Простий інтерфейс	Створення резервних копій, керування розкладування та налаштування FTP-сервера

Рисунок 10 – Схема аналізу аналогів

Розробка власного десктопного застосунку для створення backup даних на ftp-сервері має актуальність порівняно з іншими аналогічними роботами через свою універсальність, зручний інтерфейс, додаткові функціональні можливості. Важливо зазначити деякі переваги даного застосунку, а саме: безкоштовна з відкритим вихідним кодом, простий та інтуїтивно зрозумілий інтерфейс, функції створення резервних копій, автоматичної синхронізація та розкладування резервного копіювання

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ДЕСКТОПНОГО ЗАСТОСУНКУ ДЛЯ СТВОРЕННЯ BACKUP ДАНИХ НА FTP-СЕРВЕРІ

Щоб підвищити надійність зберігання інформації, доводиться вдаватися до резервного копіювання даних. А коли обсяг даних стає занадто великим, постає питання про пошук програми для автоматизації цього процесу. При розробці застосунку, були поставлені наступні вимоги до програми, а саме створити/розробити застосунок з:

- простим, зрозумілим, бажано український інтерфейс, щоб не витрачалося багато часу на вивчення програми;
- можливістю копіювання цілих директорій з одного місця в інше, саме у тому вигляді, в якому вони представлені у джерелі, з іконками, вкладеними папками тощо (це дозволяє користуватися локальною резервною копією так само, як і джерелом; як не дивно, але більшість програм резервного копіювання намагаються запакувати бекап в якийсь тільки їм зрозумілий формат);
- вмінням зберігати бекап на віддаленому FTP сервері (нині це можна зробити абсолютно безкоштовно);
- вмінням архівувати бекап в один із популярних архівних форматів шифрування резервної копії, щоб зберігати конфіденційні дані на віддаленому сервері;
- вмінням виконувати завдання за розкладом;
- невимогливістю до фінансів.

Після запуску програми на екрані з’являється головне вікно (рис. 11). Інтерфейс розробленої програми максимально простий тому має всього одне вікно.

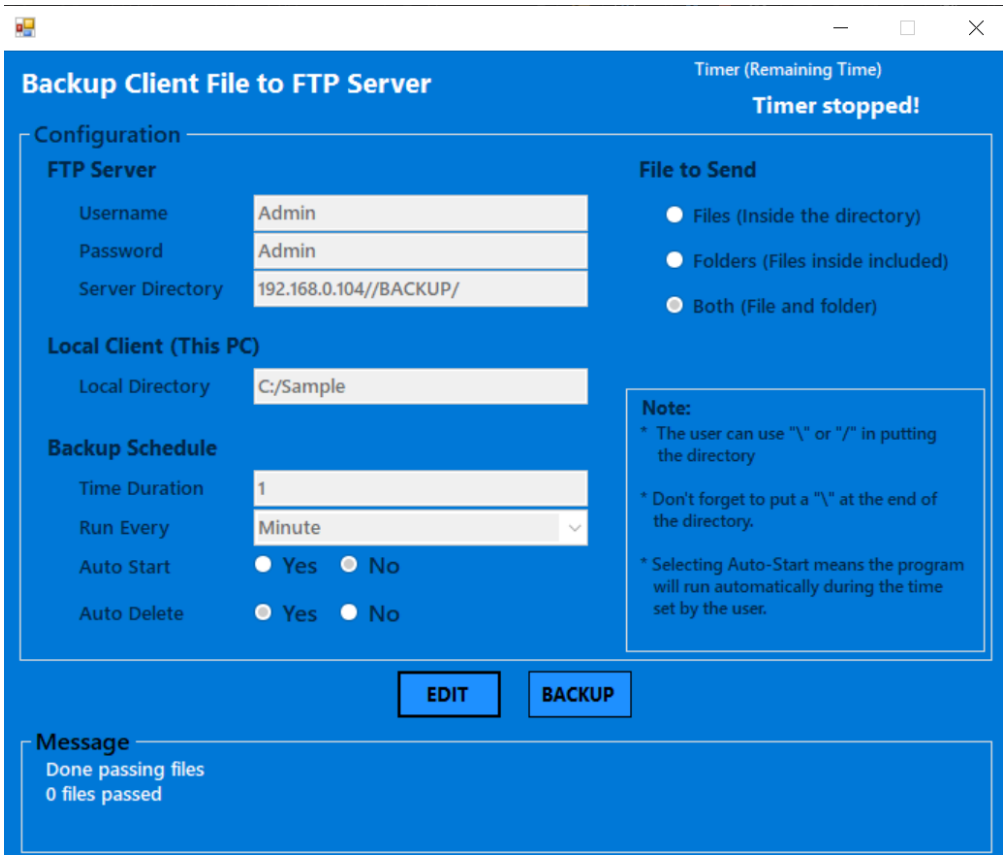

Рисунок 11 – Головне вікно програми

Після першого запуску програми необхідно провести налаштування. По-перше вказати данні FTP серверу, а саме: логін та пароль користувача, шлях до каталогу на сервері де будуть зберігатися файли з комп’ютеру (рис 12)

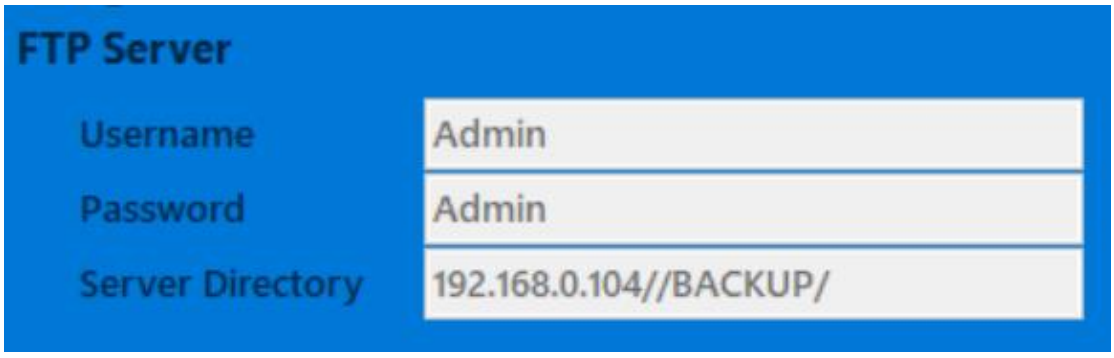

Рисунок 12 – Підключення до FTP серверу

Наступним етапом, необхідно вказати локальний каталог, який буде синхронізуватися з сервером (рис 13)

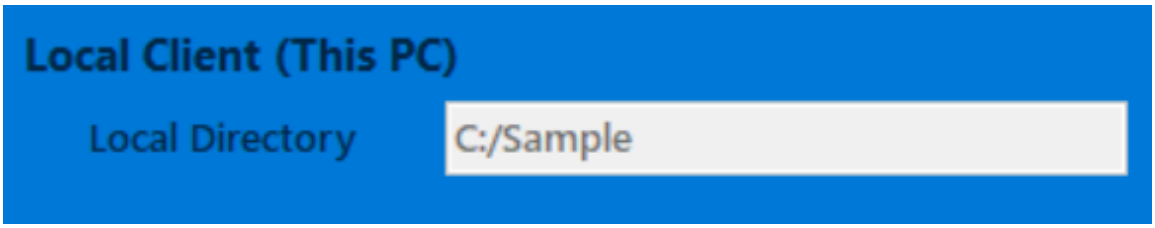

Рисунок 13 – Локальний каталог для синхронізації

Після, слід обрати тип, який буде синхронізуватися (рис. 14).

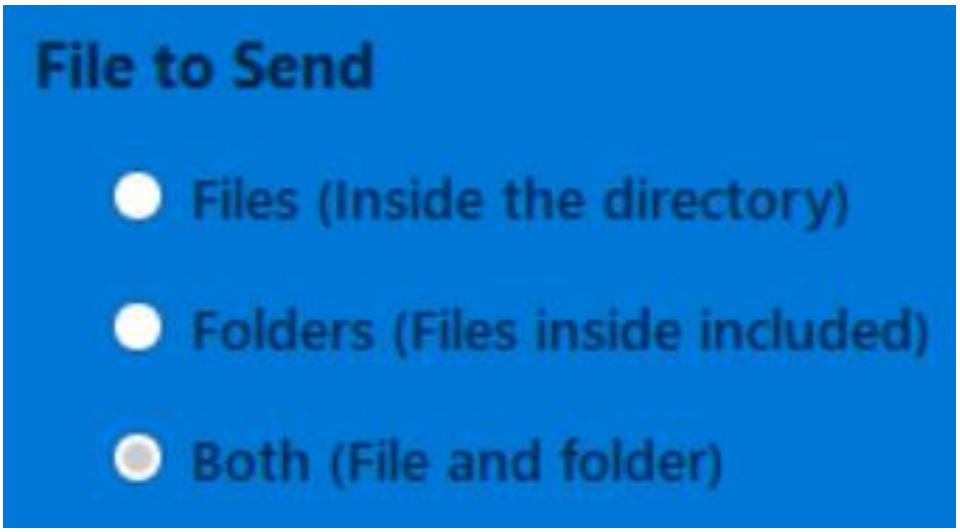

Рисунок 14 – Налаштування синхронізації

Синхронізація може проводитися в двох режимах: ручному та автоматичному. Для роботи автоматичного режиму потрібно налаштувати розклад (рис. 15).

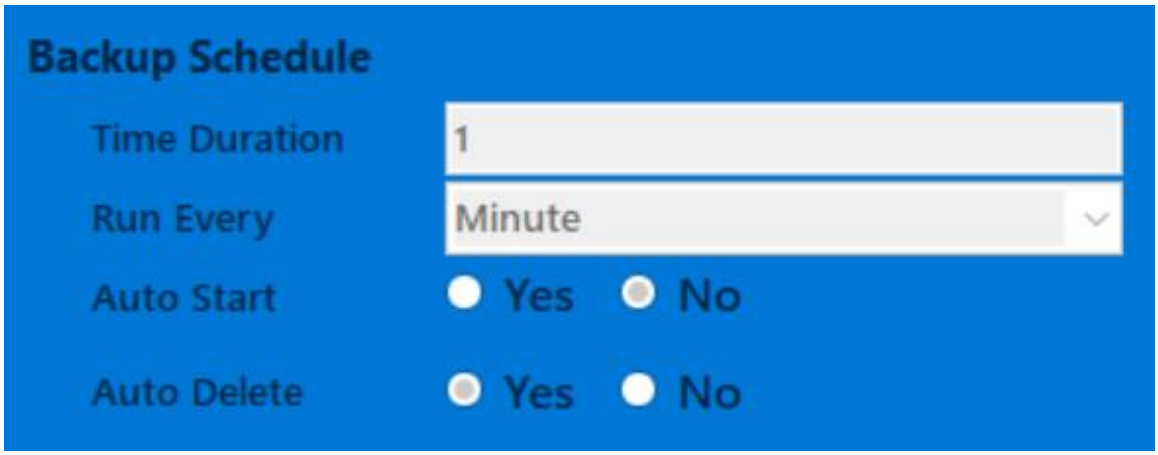

Рисунок 15 – Налаштування розкладу синхронізації

Пункти налаштування розкладу представлені нижче:

– Time Duration: дозволяє вказати числове значення від 1 до 60;
– Run Every: обирається одиниця хвилини, чи годинию;
– Auto Start: якщо поставити відмітку «Yes», то синхронізація буде проводитися автоматична через вказаний користувачем час;
– Auto Delete: якщо поставити відмітку «Yes», то на сервері буде замінюватися синхронізуючий файл, якщо є відмінності.

У верхній правій частині програмі відображається таймер з кількістю часу, що залишився до синхронізації, а у нижній частині у блоці повідомлень можна відслідковувати стан.

```
public FtpSyncScheduler(string localDir, string remoteDir,
string host, string username, string password)
    {
        localDirectory = localDir;
        remoteDirectory = remoteDir;
        ftpHost = host;
        ftpUsername = username;
        ftpPassword = password;
        timer = new Timer();
        timer.Elapsed += TimerElapsed;
    }
```

В цьому прикладі клас FtpSyncScheduler використовує таймер для запуску синхронізації файлів між локальною директорією і FTP-сервером за заданим інтервалом. Конструктор класу приймає шляхи до локальної і віддаленої директорій, а також хост FTP-сервера, ім'я користувача і пароль.

Метод StartSync запускає таймер і встановлює заданий інтервал синхронізації. Після встановлення таймеру виконується перша синхронізація файлів.

```
public void StartSync(int intervalMinutes){
   timer.Interval = intervalMinutes * 60 * 1000; // перетворення
хвилин в мілісекунди
   timer.Start();
   Console.WriteLine("Синхронізація розкладу запущена. Інтервал:
{0} хвилин.", intervalMinutes);
   SyncFiles(); // спочатку виконуємо синхронізацію без затримки
}
```

Метод StopSync зупиняє таймер.

```
    public void StopSync(){
        timer.Stop();
        Console.WriteLine("Синхронізація розкладу зупинена.");
    }
   private void TimerElapsed(object sender, ElapsedEventArgs e){
        SyncFiles();
    }
```

Метод SyncFiles виконує синхронізацію файлів з локальної директорії на FTP-сервер. Він проходить через всі файли в локальній директорії, завантажує кожен файл на FTP-сервер і виводить повідомлення про успішну синхронізацію.

```
    private void SyncFiles()
    {try{
            string[] files = Directory.GetFiles(localDirectory);
            foreach (string file in files){
       string fileName = Path.GetFileName(file);
```

```
        string remoteFilePath = remoteDirectory + "/" + fileName;
    FtpWebRequest ftpRequest = (FtpWebRequest)
WebRequest.Create("ftp://" + ftpHost + "/" + remoteFilePath);
          ftpRequest.Method = WebRequestMethods.Ftp.UploadFile;
     ftpRequest.Credentials = new NetworkCredential(ftpUsername,
ftpPassword);
        using (Stream ftpStream = ftpRequest.GetRequestStream())
           using (FileStream fileStream = File.OpenRead(file))
                {fileStream.CopyTo(ftpStream); }
       Console.WriteLine("Файл {0} успішно синхронізований з
FTP-сервером.", fileName);}}
        catch (Exception ex)
  {Console.WriteLine("Помилка при синхронізації файлів:"+
ex.Message);
        }}}
```

У головному методі Main можна налаштувати параметри синхронізації, такі як шляхи до директорій, хост FTP-сервера, облікові дані користувача та інтервал синхронізації. Після запуску розкладу синхронізації програма очікує натискання клавіші Enter для завершення, після чого розклад зупиняється.

```
public static void Main(string[] args) {
        string localDirectory = @"C:\LocalDirectory";
        string remoteDirectory = "/RemoteDirectory";
        string ftpHost = "example.com";
        string ftpUsername = "username";
        string ftpPassword = "password";
        int syncIntervalMinutes = 60;
        FtpSyncScheduler scheduler = new
FtpSyncScheduler(localDirectory, remoteDirectory, ftpHost,
ftpUsername, ftpPassword);
        scheduler.StartSync(syncIntervalMinutes);
        Console.WriteLine("Натисніть Enter для завершення.");
        Console.ReadLine();
        scheduler.StopSync();
    }}
```

Розробка десктопного застосунку для створення резервних копій даних на FTP-сервері є корисним та важливим завданням. Такий застосунок забезпечує зручність та ефективність для користувачів, дозволяючи їм створювати та керувати резервними копіями даних. Автоматизація процесу допомагає

уникнути ручної роботи та забезпечує регулярне створення копій.

Застосунок забезпечує безпеку даних шляхом зберігання резервних копій на віддаленому FTP-сервері та застосування механізмів автентифікації та шифрування. Користувачі можуть легко відновлювати дані з резервних копій в разі необхідності. Такий десктопний застосунок є надійним інструментом для забезпечення безпеки та збереження важливої інформації на FTP-сервері.

Нижчеописаний код представляє функцію download, яка виконує завантаження файлу з FTP-сервера. Основні кроки, які він виконує, такі:

```
public ftp(string hostIP, string userName, string password) { host =
hostIP; user = userName; pass = password; }
public void download(string remoteFile, string localFile){  try
    {ftpRequest = (FtpWebRequest)FtpWebRequest.Create(host + "/" +
remoteFile);
     ftpRequest.Credentials = new NetworkCredential(user, pass);
     ftpRequest.UseBinary = true;
     ftpRequest.UsePassive = true;
     ftpRequest.KeepAlive = true;
     ftpRequest.Method = WebRequestMethods.Ftp.DownloadFile;
     ftpResponse = (FtpWebResponse)ftpRequest.GetResponse();
     ftpStream = ftpResponse.GetResponseStream();
     FileStream localFileStream = new FileStream(localFile,
FileMode.Create);
     byte[] byteBuffer = new byte[bufferSize];
     int bytesRead = ftpStream.Read(byteBuffer, 0, bufferSize);
try{
     while (bytesRead > 0)
     {localFileStream.Write(byteBuffer, 0, bytesRead);
     bytesRead = ftpStream.Read(byteBuffer, 0, bufferSize);}}
```

Основні кроки, які виконує застосунок:

- створення об'єкта FtpWebRequest з використанням переданого хосту (hostIP) та віддаленого файлу (remoteFile);
- налаштування облікових даних для аутентифікації на FTP-сервері за допомогою об'єкта NetworkCredential;
- встановлення параметрів передачі файлу, зокрема використання бінарного режиму, пасивного режиму, збереження з'єднання та вста-

новлення методу передачі файлу як DownloadFile;

- виклик GetResponse для отримання відповіді з FTP-сервера;
- отримання потоку для читання даних з FTP-сервера за допомогою GetResponseStream;
- створення локального файлу за допомогою FileStream;
- читання даних з FTP-сервера та запис їх у локальний файл у циклі;
- закриття потоків та очищення ресурсів.

Загалом, цей код виконує операцію завантаження файлу з FTP-сервера у локальний файл за допомогою переданих параметрів хосту, імені користувача та пароля. Він використовує стандартні класи .NET для роботи з FTP-протоколом. У наступному прикладі функція UploadFile використовує об'єкт FtpWebRequest для завантаження файлу на FTP-сервер. Далі, необхідно замінити "ftp://example.com/" на URL FTP-сервера, "username" та "password" на облікові дані для аутентифікації.

```
public void UploadFile(string localFile, string remoteFile){try {
FtpWebRequest ftpRequest =
(FtpWebRequest)WebRequest.Create("ftp://example.com/" + remoteFile);
        ftpRequest.Method = WebRequestMethods.Ftp.UploadFile;
        ftpRequest.Credentials = new NetworkCredential("username",
"password");
        byte[] fileBytes = File.ReadAllBytes(localFile);
        using (Stream ftpStream = ftpRequest.GetRequestStream())
        {ftpStream.Write(fileBytes, 0, fileBytes.Length);}
        Console.WriteLine("Файл успішно збережений на FTP-сервері.");}
    catch (Exception ex)
    {Console.WriteLine("Помилка при збереженні файлу на FTP-сервері: "
+ ex.Message);}
}.Message); } }
```

Функція спочатку читає вміст локального файлу у байтовий масив fileBytes. Потім вона відкриває потік ftpStream за допомогою GetRequestStream і записує байтовий масив на FTP-сервер. Після успішного завершення запису вона виводить повідомлення про успішне збереження файлу, а в разі виникнення помилки – виводить повідомлення про помилку.

ВИСНОВКИ

FTP-сервер – це сервер, який працює за протоколом File Transfer Protocol і призначений для обміну файлами через Інтернет чи локальну комп'ютерну мережу.

В результаті написання кваліфікаційної роботи бакалавра було розроблено десктопний застосунку для створення резервних копій даних на FTP-сервері. Даний застосунок має зручний інтерфейс та використовує базові методи надійності, він дозволяє користувачам зберігати свої дані на віддаленому сервері для запобігання втрати і відновлення даних в разі необхідності.

В кваліфікаційній роботі бакалавра було розроблено десктопний застосунок для створення backup даних на ftp-сервері.

Перелік завдань, які було виконано:

- аналіз проблеми резервного копіювання даних в застосунках;
- огляд існуючих рішень, створення математичної моделі,
- розробка десктопного застосунка для створення backup даних на ftp-сервері.

Розробка десктопного застосунку для створення резервних копій даних на FTP-сервері є важливим і корисним завданням. Ось деякі висновки, які можна зробити з роботи:

- зручність та ефективність: розробка десктопного застосунку дозволяє користувачам зручно та ефективно створювати резервні копії своїх даних на FTP-сервері; застосунок може мати простий та інтуїтивно зрозумілий інтерфейс, який дозволяє користувачам швидко налаштувати параметри резервного копіювання та легко керувати процесом;
- автоматизація: десктопний застосунок може бути налаштований для автоматичного резервного копіювання даних на FTP-сервер і це дозволяє користувачам уникнути ручного процесу копіювання та забезпечити регулярне й надійне створення резервних копій; автома-

тизований застосунок може запускатися за заданим графіком або відслідковувати зміни в файловій системі і виконувати резервне копіювання при необхідності;

- безпека даних: застосунок для створення резервних копій на FTP-сервері допомагає забезпечити безпеку даних; резервні копії, збережені на віддаленому сервері, захищають дані від випадкового видалення, втрати чи пошкодження, крім того, FTP-сервери часто мають механізми автентифікації та шифрування, що забезпечують конфіденційність інформації під час передачі;
- легкість відновлення: резервні копії, створені за допомогою десктопного застосунку, легко відновлюються у раз.

На сьогоднішній день, створення десктопних застосунків для резервного копіювання даних на FTP-сервери залишається актуальним. FTP (File Transfer Protocol) є одним з найпоширеніших протоколів передачі файлів, і багато організацій та користувачів використовують його для збереження резервних копій своїх даних.

Десктопні застосунки для резервного копіювання на FTP-сервери надають зручність та контроль користувачам, оскільки вони можуть локально керувати процесом створення, завантаження та відновлення резервних копій. Це дає користувачам більшу гнучкість і можливість використовувати FTP-сервери своєї власної вибору.

Незважаючи на те, що існують інші протоколи передачі файлів, такі як SFTP і FTPS, які надають більшу безпеку за допомогою шифрування, FTP залишається популярним і широко використовується у багатьох сценаріях. Тому, десктопні застосунки для резервного копіювання на FTP-сервери можуть продовжувати бути актуальними для тих, хто використовує FTP-сервери для збереження своїх даних.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1 Handy Backup. URL: https://www.handybackup.net/ftp-backup.shtml (дата звернення 01.02.2023).
2 Techtarget. URL: https://www.techtarget.com/searchdatabackup/modern-backup-considerations/resources/Best-Practices-in-Backup-Storage(дата звернення 10.03.2023).
3 Pro2col. URL: https://pro2col.com/file-transfer-protocol-ftp (дата звернення 15.04.2023).
4 Contabo. URL: https://contabo.com/blog/backup-strategies/ (дата звернення 12.05.2023).
5 Top 10 Backup Software. URL: www.techradar.com/best/backup-software (дата звернення 15.09.2022).
6 Guidelines for Data Backup. URL: www.nist.gov/guidelines/data-backup-recovery(дата звернення 09.06.2019).
7 Data Backup and Recovery: Strategies and Best Practices. URL: https://blog.tbconsulting.com/critical-data-backup-and-recovery-strategy-best-practices (дата звернення 2018).