

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ
ДЛЯ ГЕНЕРАЦІЇ ЗВІТНОЇ ДОКУМЕНТАЦІЇ
SOFTWARE TO GENERATE
REPORTING DOCUMENTATION

Виконав(ла): студент(ка) 4 курсу, групи КНТ-121
Спеціальності 121 Інженерія програмного
(код і найменування спеціальності)

забезпечення

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

СИНЯВСЬКИЙ М.А.

(ПРІЗВИЩЕ та ініціали)

Керівник ГОФМАН Є.О.

(ПРІЗВИЩЕ та ініціали)

Рецензент СКРУПСЬКИЙ С.Ю.

(ПРІЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)

Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2025 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

СИНЯВСЬКОГО Михайла Артемовича

(ПРІЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Програмне забезпечення для генерації звітної документації. Software to Generate Reporting Documentation

керівник проєкту (роботи) к.т.н., доцент, ГОФМАН Євгеній Олександрович,
(науковий ступінь, вчене звання, ПРІЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “ 28 ” квітня 2025 року № 209

2. Строк подання студентом проєкту (роботи) 03 червня 2025 року
3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Опис програми. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів) Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРІЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	ОЛІЙНИК А.О., професор		
Нормоконтроль	КАМІНСЬКА Ж.К., асистент		

7. Дата видачі завдання “28” квітня 2025 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка структури програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-4 тижні	Розділи 3, 4
6	Тестування та експериментальне дослідження програмного забезпечення.	5 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	6 тиждень	Додатки
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

(підпис) Михайло СИНЯВСЬКИЙ
(Ім'я ПРІЗВИЩЕ)

Керівник проєкту (роботи)

(підпис) Євгеній ГОФМАН
(Ім'я ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра:
85 с., 3 табл., 26 рис., 3 дод., 18 джерел.

C#, RIDER, ДОКУМЕНТ, МОВА ПРОГРАМУВАННЯ, ПРОГРАМНЕ
ЗАБЕЗПЕЧЕННЯ, СЕРЕДОВИЩЕ РОЗРОБКИ.

Об'єкт дослідження – процес розробки програмного забезпечення для генерації звітної документації.

Предмет дослідження – програмні засоби для генерації звітної документації.

Мета роботи – розробка програмного забезпечення для генерації звітної документації.

Матеріали, методи та технічні засоби: мова програмування C#, середовище розробки Rider.

Результати. Розроблено проєктні рішення для створення програмного забезпечення для генерації звітної документації. Створено програмне забезпечення для генерації звітної документації. Проведено тестування розробленої програмної системи для генерації звітної документації.

Висновки. Мету роботи досягнуто. Розроблено програмне забезпечення для генерації звітної документації за допомогою мови програмування C# та середовища розробки Rider.

Галузь використання – офісні застосунки.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 85 pages, 3 tables, 26 figures, 3 appendixes, 18 sources.

C#, RIDER, DOCUMENT, PROGRAMMING LANGUAGE, SOFTWARE, DEVELOPMENT ENVIRONMENT.

The object of research is the process of developing software for working with documents.

The subject of the research is software for generating reporting documentation.

The purpose of this work is to develop the software for generating reporting documentation.

Materials, methods and technical tools: C# programming language, Rider development environment.

Results. Project solutions for the creation of software for generating reporting documentation has been designed. The software for generating reporting documentation has been developed. The developed software system for generating reporting documentation has been tested.

Conclusions. The goal of the work has been achieved. The software tools for generating reporting documentation using the C# programming language and the Rider development environment have been developed.

Scope of use – office applications.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок	8
Вступ	9
1 Аналіз предметної області	11
1.1 Програмне забезпечення для генерації звітної документації	11
1.2 Аналіз програмного забезпечення для генерації звітної документації ...	23
1.3 Висновки за розділом 1	31
2 Матеріали і методи	33
2.1 Вибір мови програмування	33
2.2 Вибір середовища розробки для створення програмного забезпечення для генерації звітної документації	36
2.3 Висновки за розділом 2	39
3 Опис програми	41
3.1 Структура програмного забезпечення для генерації звітної документації	41
3.2 Функціонування програмного забезпечення для генерації звітної документації	43
3.3 Розробка бази даних програмного забезпечення для генерації звітної документації	45
3.4 Проєктування інтерфейсу програмного забезпечення для генерації звітної документації	47
3.5 Висновки за розділом 3	49
4 Експлуатація, тестування та експериментальне дослідження програми	51
4.1 Призначення й умови застосування програми	51
4.2 Характеристики програми для генерації звітної документації	52
4.3 Інструкція по експлуатації програми	53
4.3.1 Звернення до програми	53
4.3.2 Вхідні й вихідні дані	53
4.3.3 Повідомлення	53

4.4 Виконання програмного забезпечення для генерації звітної документації	54
4.5 Тестування програмного забезпечення для генерації звітної документації	57
4.6 Висновки за розділом 4	57
Висновки	58
Перелік джерел посилань	61
Додаток А Технічне завдання	63
Додаток Б Фрагмент тексту програми	67
Додаток В Слайди презентації	78

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

API – application programming interface;

БД – база даних;

ЗД – звітна документація;

ПЗ – програмне забезпечення.

ВСТУП

У сучасних умовах функціонування організацій спостерігається постійне зростання обсягів даних, які необхідно обробляти, аналізувати й представляти у вигляді звітів. У зв'язку з цим дедалі більшої значущості набувають програмні рішення, що дозволяють автоматизувати створення звітної документації. Такі програмні інструменти забезпечують не лише зручність у підготовці звітів, а й сприяють підвищенню точності, оперативності та відповідності документації чинним вимогам [1]-[3].

Застосування спеціалізованого програмного забезпечення для генерації звітної документації дає змогу уникати помилок, пов'язаних із ручним введенням даних, а також значно скорочує час, необхідний для підготовки звітів. Завдяки можливості підключення до різноманітних джерел інформації забезпечується формування актуальних і достовірних звітів у реальному часі. Водночас автоматизована генерація документів дозволяє підтримувати єдиний стиль і формат подання інформації, що полегшує подальше опрацювання та використання таких матеріалів у внутрішніх чи зовнішніх процесах [4], [5].

Зважаючи на те, що в багатьох галузях звітність повинна відповідати певним шаблонам, виникає потреба у таких програмних засобах, які б враховували регламентовані вимоги без потреби ручного втручання. Крім того, з огляду на динамічність бізнес-середовища, програмні рішення повинні забезпечувати адаптивність до змін [6], [7].

Проте деяке програмне забезпечення для генерації звітної документації є складним для налаштування. Щоб програмне забезпечення працювало коректно, часто потрібне попереднє моделювання структури даних, створення шаблонів звітів та інтеграція з іншими програмними системами. Ці процеси можуть вимагати специфічних знань, часу та технічних ресурсів. У випадку помилок на етапі конфігурації результати генерації звітів можуть виявитися некоректними або неповними. Ще одним ускладненням є обмежена гнучкість деяких програмних інструментів. У разі необхідності створення

нестандартного звіту, який не відповідає заздалегідь заданим шаблонам або не передбачений системою, можуть виникати труднощі або навіть потреба в ручному втручанні. Важливою проблемою залишається й залежність від технічної інфраструктури. При нестабільному доступі до баз даних, недостатній обчислювальній потужності або відсутності регулярного оновлення програмного забезпечення ефективність систем генерації звітів може знижуватися. Це може призвести до затримок, збоїв або втрати даних у процесі підготовки документації. Зрештою, варто зазначити, що вартість ліцензійного програмного забезпечення або його технічного обслуговування може бути досить високою [1]-[6].

Тому актуальною є розробка програмного забезпечення для генерації звітної документації. У дипломній кваліфікаційній роботі бакалавра розв'язується актуальне завдання розробки програмного забезпечення для генерації звітної документації.

Для досягнення поставленої мети у кваліфікаційній роботі бакалавра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та програмних засобів для генерації звітної документації;
- здійснити проєктування програмного забезпечення для генерації звітної документації;
- створити програмне забезпечення для генерації звітної документації;
- виконати тестування розробленого програмного забезпечення для генерації звітної документації.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Програмне забезпечення для генерації звітної документації

Генерація звітної документації — це процес автоматизованого створення звітів на основі зібраних і оброблених даних. Така документація використовується для аналізу, оцінки та інформування про стан процесів, результатів діяльності, відповідність стандартам тощо [1], [2].

У межах цього процесу формуються структуровані документи, які містять аналітичну, статистичну або фактологічну інформацію. Генерація може охоплювати різні етапи — від збору даних з баз даних, сервісів або систем до їх візуалізації, форматування та подальшого розповсюдження у вигляді таблиць, графіків, діаграм або текстових звітів [1], [2].

Цей підхід широко застосовується в бізнесі, аудиті, фінансах, інформаційних технологіях, освіті та інших сферах, де регулярно потрібно формувати звіти для внутрішнього чи зовнішнього використання. Застосування спеціалізованого програмного забезпечення дозволяє знизити витрати часу, зменшити кількість помилок і підвищити ефективність управлінських рішень [1], [2].

Процес створення звітної документації часто пов'язаний із використанням спеціалізованих програмних засобів, які дають змогу згенерувати необхідні звіти без зайвих труднощів. Завдяки таким програмам можна автоматизувати отримання, структурування та представлення інформації, що дозволяє формувати звіти швидко, без зайвого втручання користувача та з мінімальними помилками. Потреба в ручному збиранні даних з різних джерел, як це зазвичай відбувалося під час роботи в табличних редакторах, більше не є актуальною. Усе це замінюється централізованим процесом, де шаблони, джерела даних і правила форматування вже налаштовані й доступні у зручному інтерфейсі [1], [2].

Переваги сучасних інструментів генерації звітної документації полягають у тому, що не потрібно постійно дублювати одну й ту саму роботу

для щоденних, тижневих чи місячних звітів. За один раз налаштовані правила застосовуються повторно, що значно економить час. Дані можуть бути об'єднані з кількох систем або баз автоматично, а звіти сформовані та доставлені в потрібному вигляді незалежно від технічної підготовки користувача [1], [2].

У багатьох випадках звіти створюються та надсилаються без участі людини, відповідно до графіка, що було встановлено заздалегідь. Шаблони при цьому використовуються повторно, адаптуючись під різні потреби бізнесу. Можливість перегляду створених документів у браузері або з мобільного пристрою дає змогу керівникам отримувати потрібну інформацію в будь-який момент. Таким чином, використання програмного забезпечення для генерації звітів дає змогу зосередитися не на підготовці даних, а на прийнятті рішень, що базуються на цих даних [1], [2].

Проаналізуємо процес створення звітної документації за допомогою програмних засобів. Звіти формуються шляхом опрацювання наявних даних та їх структурування в зручному для сприйняття форматі. За допомогою спеціалізованого програмного забезпечення цей процес можна реалізувати як повністю автоматизовано, так і з частковим залученням користувача. Автоматизовані системи здатні створювати фінансові, статистичні та інші типи звітів на основі наперед визначених шаблонів і формул. Достатньо здійснити підключення до потрібних джерел інформації, після чого дані інтегруються до відповідних секцій документа без додаткового втручання. При цьому забезпечується стандартизація звітності та економія часу [1], [2].

У випадках, коли необхідна індивідуалізація або специфічна структура документа, застосовується підхід з частковою участю користувача. У такому разі здійснюється налаштування параметрів підключення, визначення полів для виводу, редагування форматування та узагальнення даних відповідно до вимог конкретного завдання. Після завершення підготовки матеріали можуть бути опубліковані, збережені чи надіслані відповідним користувачам [1], [2].

Використання таких програмних систем варіюється залежно від технологічної платформи: одні побудовані на Java, інші використовують інші мови або комбіновані підходи. Різні типи інтерфейсів — від таблиць, подібних до електронних таблиць, до графічних з можливістю вільного компоновання — дозволяють адаптувати інструмент до специфіки бізнес-процесів [1], [2].

У практиці звітності автоматизовані засоби відіграють ключову роль, дозволяючи спростити щоденну аналітичну діяльність та підвищити достовірність інформації. Вибір відповідного інструменту потребує попередньої оцінки функціональних можливостей з урахуванням реальних потреб організації [1], [2].

Програмні платформи для створення звітності забезпечують можливість трансформувати необроблену інформацію з різних джерел у структуровані аналітичні матеріали, що відображають стан справ у бізнес-середовищі. Застосування таких систем дозволяє формувати звітність за широким спектром параметрів — від операційної діяльності до стратегічних орієнтирів. Часто підґрунтям для створення звіту слугує необхідність оцінити результативність або виявити поточні проблеми у функціонуванні організації. Завдяки такому програмному забезпеченню з'являється змога інтегрувати інформацію з численних джерел — баз даних, таблиць, редакторів, сервісів вебаналітики — і на основі цієї інформації будувати візуальні представлення, діаграми, таблиці та графіки. Ці процеси можуть відбуватись автоматизовано, з мінімальною участю користувача, що суттєво оптимізує часові витрати на підготовку звітів [1], [2].

Гнучкість налаштувань дає змогу адаптувати звітність під специфічні вимоги окремих підрозділів чи напрямів бізнесу. Зібрані дані, представлені у вигляді звітів, слугують джерелом для формування аналітичних висновків, ідентифікації тенденцій, прогнозування змін на ринку та контролю ключових показників діяльності [1], [2].

Автоматизація таких задач знижує операційне навантаження та дає можливість зосередитись на аналізі змісту, а не на механіці створення документів. У процесі вибору відповідного інструменту для звітності слід враховувати не лише його технічні характеристики, а й ступінь адаптованості до специфіки бізнес-процесів компанії [1], [2].

На ринку доступно багато програмних рішень: від багатофункціональних систем бізнес-аналітики, здатних опрацьовувати великі масиви даних, до програм з фокусом на створення динамічних інформаційних панелей, або ж класичних табличних додатків, які традиційно застосовуються для бюджетного планування та фінансових оцінок. Окрему нішу займають інструменти, що орієнтовані на можливості налаштування форм звітності для управлінських потреб, а також сервіси, що відстежують поведінку користувачів в онлайн середовищі [1], [2].

Ефективність використання кожного з програмних інструментів визначається контекстом завдань та очікуваним рівнем деталізації звітів. Під час оцінювання таких систем доцільно враховувати також підтримку асинхронної взаємодії, можливість командної роботи та доступність інтерфейсу для користувача з різним рівнем технічної підготовки. У результаті відповідний вибір програмного рішення має ґрунтуватися не лише на функціональності, а й на здатності інструмента органічно вписатися у наявну інфраструктуру [1], [2].

Програмні засоби для створення звітів та рішення в сфері бізнес-аналітики виконують схожі функції, проте мають відмінні напрямки використання. Перші орієнтовані на формування підсумкових звітів на основі даних із обмежених джерел, як-от окрема база даних чи електронна таблиця. Натомість інтелектуальні системи охоплюють широкий спектр задач: обробку інформації з багатьох джерел, глибокий аналіз, побудову моделей та інтерактивну візуалізацію [1], [2].

Застосування звичайних програмних інструментів для звітності зазвичай не потребує спеціальних технічних знань, оскільки вони

створювалися з урахуванням потреб користувачів без досвіду в інформаційних технологіях. Навпаки, використання бізнес-аналітичних платформ часто передбачає оперування складними аналітичними механізмами й вимагає певної підготовки [1], [2].

У питанні функціоналу програмне забезпечення для формування звітів забезпечує базові можливості – створення шаблонів, фільтрація, вивід даних у зручній формі. Бізнес-аналітичні програмні інструменти, у свою чергу, дозволяють будувати прогнози, виконувати багатовимірний аналіз та використовувати методи штучного інтелекту для виявлення прихованих закономірностей [1], [2].

Вартісна доступність також відрізняється: прості засоби звітності підходять для невеликих організацій або індивідуальних користувачів, тоді як комплексні Бізнес-аналітичні програмні рішення більше орієнтовані на середній та великий бізнес, де потреби в аналізі значно вищі [1], [2].

Попри різницю в функціоналі та вартості, обидва підходи мають свої переваги і застосовуються відповідно до потреб конкретної організації. Перед впровадженням певного типу програмного забезпечення рекомендується проаналізувати специфіку задач і доступні ресурси [1], [2].

Використання інструментів для автоматизованого створення звітів значно впливає на продуктивність організацій. Завдяки інтуїтивному інтерфейсу й можливості швидко компонувати структуру звіту, користувачі здатні зменшити витрати часу й уникнути людських помилок під час формування звітної документації [1], [2].

Практика показує, що налаштування структури звіту під конкретні потреби організації дозволяє значно покращити обґрунтованість управлінських рішень, адже виводяться саме ті метрики, які найбільше відповідають бізнес-цілям [1], [2].

Графічне подання даних також спрощує сприйняття великих обсягів інформації. Завдяки візуальним елементам, таким як графіки чи діаграми, можна оперативно виявляти тенденції й робити висновки навіть без глибокого

аналізу. Ще одним важливим аспектом є автоматизація: процес формування звітів може бути запущений за розкладом або з певною періодичністю, що вивільняє час фахівців для більш стратегічних задач. Точність результатів підвищується за рахунок того, що система здатна перевіряти логіку обробки даних, виявляти аномалії та усувати помилки, що зменшує ризики прийняття хибних рішень [1], [2].

У цифрову епоху, коли ефективне управління інформацією має вирішальне значення, впровадження сучасних програмних рішень для звітності стає ключовим фактором успіху. Такі програмні інструменти не лише покращують доступ до бізнес аналітики, а й інтегруються з наявною програмною інфраструктурою, забезпечуючи оперативну підтримку бізнесу на основі реальних даних. Це дає змогу організаціям не лише оперативно реагувати на зміни, а й залишатися конкурентоспроможними в динамічному середовищі [2], [3].

Під час визначення найбільш доцільного інструмента для формування звітності доцільно орієнтуватися на низку характеристик, що визначають його функціональну придатність у конкретному середовищі застосування. До основних аспектів належать технічні можливості платформи щодо створення потрібних видів звітності, швидкість генерації результатів, а також гнучкість у налаштуванні відповідно до індивідуальних запитів [2], [3].

Не менш значущим є рівень сумісності програмного інструмента з чинними джерелами інформації. За відсутності належної інтеграції з наявними базами даних, табличними документами або сторонніми сервісами аналітики ефективність застосування значно знижується. Важливо також, щоб інтерфейс програми був інтуїтивно зрозумілим, зручним у навігації та дозволяв швидко орієнтуватися у функціоналі навіть користувачам без глибоких технічних знань [2], [3].

Окремої уваги потребує захист даних. Використання технологій шифрування, управління правами доступу та інших засобів забезпечення

безпеки є критичним фактором при роботі з конфіденційною або стратегічною інформацією [2], [3].

У перспективі зростання обсягів інформації та ускладнення структури бізнес-процесів доцільно віддавати перевагу рішенням, здатним масштабуватися відповідно до зростаючих вимог. Підтримка великих обсягів даних та вбудовані засоби для глибинної аналітики сприяють підвищенню довготривалої ефективності системи [2], [3].

В умовах цифрової трансформації дедалі частіше спостерігається поєднання функціоналу традиційних систем звітності з динамічними інформаційними панелями. Така комбінація надає змогу не лише здійснювати докладний аналіз історичних даних, а й оперативно контролювати основні показники діяльності завдяки графічним відображенням у реальному часі. Подібна інтеграція дозволяє зберігати стратегічний контроль над процесами та приймати рішення на основі актуальних даних без затримок [2], [3].

Процес формування звітності передбачає використання спеціалізованих програмних засобів, що дають змогу опрацьовувати інформацію з різних джерел, обробляти її згідно з визначеними правилами та трансформувати в узагальнені підсумки, придатні для подальшого аналізу. Для цього виконуються підключення до джерел даних, визначаються ключові метрики й обираються оптимальні способи відображення результатів [3], [4].

Звіти, як правило, готуються вручну або автоматизовано. У першому випадку зростає точність і контроль над кожним етапом, що важливо для уникнення неточностей. Водночас автоматизовані засоби дають змогу значно пришвидшити процес, зберігаючи при цьому гнучкість щодо формату, структури та періодичності публікації [3], [4].

Програмне забезпечення для створення звітів виконує функції візуального упорядкування інформації: воно може об'єднувати, обробляти й представляти дані в зрозумілому вигляді. Усі дії — від побудови таблиць до формування графіків — здійснюються на основі попередньо заданих параметрів. Також передбачено можливість налаштування доступу: різні ролі

користувачів визначають, хто і які саме звіти може переглядати, редагувати або поширювати [3], [4].

Виведення результатів може відбуватися як у вигляді інтерактивних панелей, доступних у вебсередовищі або на мобільних пристроях, так і через експорт документів у зручних форматах — наприклад, CSV, PDF чи Excel. Платформи звітності часто оснащуються функціями синхронізації з хмарними сервісами, що дозволяє завжди працювати з актуальними даними без потреби у повторному завантаженні файлів [3], [4].

Інтерфейс сучасних генераторів звітів орієнтований на користувача. Завдяки інтуїтивно зрозумілим інструментам побудови вмісту за допомогою drag-and-drop, навіть користувачі без технічної підготовки можуть швидко формувати якісні звіти з використанням діаграм різних типів: стовпчастих, лінійних, кільцевих, воронкоподібних тощо [3], [4].

Налаштовані регулярні оновлення дозволяють системам самостійно формувати звітність згідно з визначеним розкладом або у відповідь на зміну показників. Це особливо зручно для моніторингу ключових бізнес-процесів і прийняття рішень на основі фактичних даних. Застосування таких засобів покликане підвищити точність, прозорість і ефективність у роботі з інформацією [4], [5].

Автоматизовані процеси в галузі звітності дають змогу оптимізувати використання ресурсів підприємства, скорочуючи витрати часу та коштів. Завдяки цьому сучасному підходу забезпечується ефективне управління інформацією без необхідності володіти глибокими технічними знаннями чи навичками програмування. Програмні платформи для створення таких звітів функціонують як на локальних пристроях, так і в хмарному середовищі, використовуючи інтерфейси прикладного програмування для збору, обробки та об'єднання даних з різноманітних джерел. Це значно знижує потребу у виконанні рутинних операцій вручну [4], [5].

Використання програмних інструментів дозволяє отримувати актуальну звітність у зручний момент, що особливо корисно для аналізу

результативності маркетингових кампаній. Дані поширюються автоматично, що сприяє швидкому реагуванню на зміни в робочому середовищі. Наявність таких рішень допомагає зменшити навантаження, пов'язане з обробкою неструктурованої інформації, і водночас зменшує ризик дублювання зусиль при створенні звітів [4], [5].

Програмні інструменти автоматизації сприяють більшій відкритості, забезпечують постійний доступ до інформації та дозволяють приймати обґрунтовані рішення без затримок. Користування такими системами зменшує потребу в довготривалому очікуванні зведених даних. Замість цього, завдяки функціоналу автоматичного оновлення, можна відразу отримати точну картину поточних подій, що особливо важливо для невеликих підприємств та команд з обмеженим ресурсом часу [4], [5].

Постійна доступність до реальних показників діяльності дозволяє своєчасно виявляти відхилення або тренди. У разі виявлення проблем прийняття рішень відбувається без затримок, що дозволяє швидше реагувати на ситуацію. Замість того, щоб користуватися застарілими інформаційними панелями з даними тижневої чи місячної давності, краще застосовувати рішення, які дозволяють об'єднувати та порівнювати статистику з різних джерел в одному середовищі. Такі підходи сприяють стабільному зростанню підприємства. Завдяки потужним аналітичним можливостям, навіть ті, хто не спеціалізується на цифровому маркетингу, можуть оперативно орієнтуватися в інформаційному просторі. Це створює передумови для подальшого вдосконалення внутрішніх процесів та підтримує динамічний розвиток бізнесу на всіх рівнях [4], [5].

Для впровадження програмних засобів автоматичного формування звітної документації існує чимало можливостей, які відрізняються між собою рівнем складності, ціною та функціональністю. Вибір відповідного програмного забезпечення залежить від поставлених завдань, характеру звітів, що мають генеруватися, а також від специфіки бізнес-процесів. Після визначення аналітичних потреб і очікуваних результатів обирається відповідне

технічне рішення, яке найкраще відповідає поточним умовам. У разі розробки програмної системи для формування звітності власноруч, процес вимагає значних ресурсів: часу, технічної обізнаності, а також готовності до тривалої підтримки й удосконалення. Хоча подібний підхід відкриває широкі можливості адаптації під індивідуальні вимоги, він одночасно ускладнює масштабування і унеможливорює отримання допомоги у випадку непередбачених труднощів [4], [5].

Використання програмних рішень класу бізнес аналітичних систем дозволяє здійснювати глибший аналіз завдяки потужним вбудованим функціям. Проте освоєння таких систем потребує часу і певної підготовки. Подібні інструменти ефективні лише за умови наявності якісних і достатньо репрезентативних даних, оскільки працюють переважно з великими обсягами інформації [4], [5].

Інтерфейси панелей для обробки даних дають змогу швидко генерувати зручні для подальшого аналізу звіти і спрощують взаємодію між учасниками проекту. Завдяки простоті доступу та мінімальним вимогам до технічних навичок, навіть користувачі без досвіду можуть самостійно формувати потрібну аналітику та ділитися нею з іншими [5], [6].

Використання готових програмних сервісів автоматизованого звітування значно полегшує процес обробки даних. Після підключення до обраного джерела інформації достатньо лише обрати відповідні метрики або скористатися шаблонними конфігураціями. У результаті значно прискорюється формування звітності, а також забезпечується доступ до даних з різних платформ без зайвих зусиль [5], [6].

Автоматизація створення звітів має значний потенціал у підвищенні ефективності бізнес-процесів. Завдяки автоматичному збору та обробці даних зникає потреба в рутинних діях, що дозволяє фахівцям зосередитися на аналітиці, прийнятті рішень та обговореннях, які потребують людського залучення. Замість того, щоб витрачати час на ручні операції, увага перемикається на смислове осмислення отриманої інформації [5], [6].

Щоб отримати повноцінну віддачу від програмної системи автоматичного формування звітної документації, доцільно заздалегідь усвідомити основні цілі впровадження й підготуватися до ключових етапів організації процесу. Варто обмірковувати, які саме аспекти діяльності потребують оптимізації, і яка інформація має бути доступною для аналізу. Це дозволить краще зрозуміти, де саме автоматизація матиме найбільший ефект, чи то в економії часу, чи в точності аналітики, чи в зменшенні кількості ручної роботи, чи в поліпшенні доступу до корпоративних даних [5], [6].

Ретельно підібрані показники ефективності слугують основою будь-якого вибору відповідного програмного забезпечення. Саме навколо них вибудовується логіка подання інформації, тому варто приділяти особливу увагу їх відповідності до поточних цілей. Якщо визначити приблизний набір ключових метрик, які найточніше відображають прогрес і якість роботи, можна значно покращити інформативність звітів. Важливо також усвідомити, з якою періодичністю має надходити інформація. Якщо оперативні завдання потребують щоденного моніторингу, то стратегічний аналіз може базуватися на даних, що збираються щотижнево чи щомісяця. Частота має відповідати характеру прийняття рішень у конкретному контексті [5], [6].

Окремо варто визначити, у якому вигляді звіти мають подаватися. Може знадобитися інтерактивна форма з постійним оновленням даних або ж фіксована, яка зберігає стан показників на певну дату. Структура має бути прозорою, а подання — адаптованим під потреби аудиторії [5], [6].

Ще одним критичним чинником є доступність даних для співробітників і їх обізнаність щодо способів роботи з автоматизованими системами. Без відповідної підготовки та відкритого доступу навіть найкраще організована система може виявитися неефективною. Завдяки належному навчанню й зручному доступу до звітів забезпечується швидка реакція на зміни, а рівень залучення персоналу до аналітики суттєво зростає [6], [7]. Таким чином, використання спеціалізованого програмного забезпечення дозволяє суттєво підвищити аналітичну зрілість організації [6], [7].

Під час організації звітності варто особливу увагу приділяти правильному добору програмних рішень, здатних задовольнити поточні й майбутні запити організації. Оскільки ринок рясніє різноманітними варіантами, вибір відповідного інструменту може перетворитися на непросте завдання. Перш за все, слід брати до уваги можливість адаптації системи до зростання обсягів даних і розширення штату. Також важливо, щоб механізми захисту інформації відповідали сучасним вимогам безпеки, забезпечуючи конфіденційність і контроль доступу. Не менш важливо, щоб звіти могли створюватися без затримок, а доступ до них залишався зручним і безперервним — незалежно від розташування користувачів [6], [7].

Інтеграція сучасних програмних платформ дозволяє суттєво спростити процес створення звітів. Всі необхідні дані можуть бути зведені в єдину візуальну структуру, адаптовану до конкретних завдань. Завдяки цьому аналітика стає зрозумілою та доступною навіть для тих, хто не має глибоких технічних знань. Автоматизовані програмні інструменти дозволяють заощаджувати ресурси й зосереджуватися на стратегічних кроках, замість рутинного створення звітів. У довгостроковій перспективі такі підходи забезпечують стабільне зростання ефективності без потреби в значному збільшенні затрат [6], [7].

Визначено, що генерація звітної документації є процесом автоматизованого створення звітів на основі зібраних і оброблених даних. Така документація використовується для аналізу, оцінки та інформування про стан процесів, результатів діяльності, відповідність стандартам тощо. У межах цього процесу формуються структуровані документи, які містять аналітичну, статистичну або фактологічну інформацію. Процес створення звітної документації часто пов'язаний із використанням спеціалізованих програмних засобів, які дають змогу згенерувати необхідні звіти без зайвих труднощів. Завдяки таким програмам можна автоматизувати отримання, структурування та представлення інформації, що дозволяє формувати звіти швидко, без зайвого втручання користувача та з мінімальними помилками.

1.2 Аналіз програмного забезпечення для генерації звітної документації

Проаналізуємо програмне забезпечення для генерації звітної документації [8]-[14].

Програмне забезпечення Supermetrics (рис. 1.1) використовується як засіб, що забезпечує збирання, обробку та передачу інформації для подальшого створення звітної документації. За допомогою цього рішення здійснюється автоматизоване перенесення даних із різноманітних джерел через API (application programming interface), після чого вони спрямовуються до середовищ, що підтримують візуалізацію та оформлення звітів, зокрема до таблиць чи панелей звітності. Завдяки інтеграції з іншими сервісами, програмне забезпечення Supermetrics відіграє роль проміжної ланки, яка забезпечує потік даних, необхідний для генерації звітної документації в аналітичних платформах [8], [12].

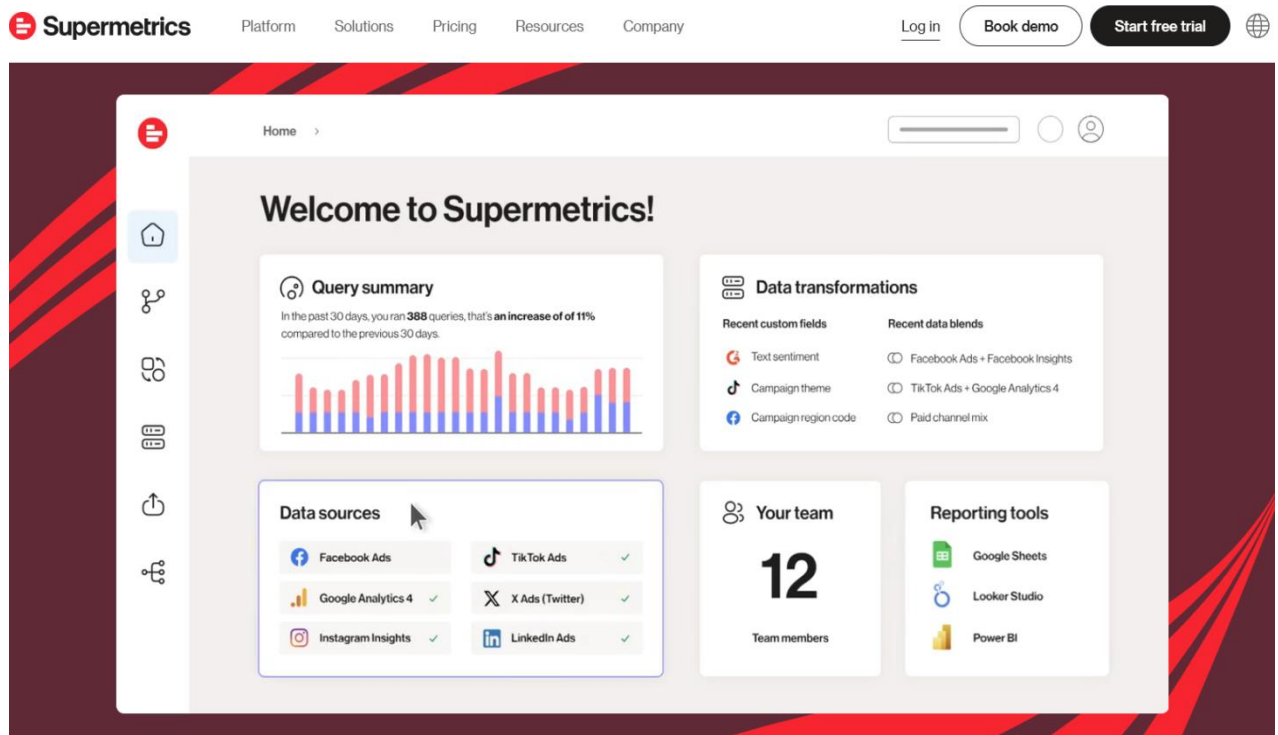


Рисунок 1.1 – Програмне забезпечення Supermetrics [12]

Програмне забезпечення Supermetrics не виконує функції оцінки достовірності інформації та не формує остаточну структуру звітів, але воно значно спрощує попередній етап, пов'язаний з обробкою та оновленням вихідних даних для створення звітних документів. У процесі роботи інструмент не здійснює експорту для побудови звітів на основі атрибуції, тому для повного циклу підготовки звітності можуть знадобитися додаткові засоби. Однак завдяки здатності систематично забезпечувати потік інформації з численних джерел Supermetrics залишається ефективним рішенням для організації та автоматизації процесу збору даних, що використовуються у звітній документації [8], [12].

Основними характеристиками програмного забезпечення Supermetrics є такі [8], [12]:

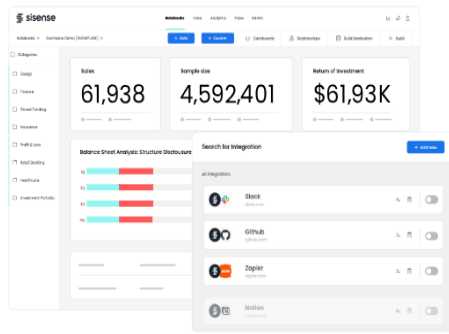
- інтеграція з різними джерелами даних: програмне забезпечення Supermetrics надає можливість підключення до численних платформ, таких як Google Analytics, Facebook Ads, HubSpot, LinkedIn Ads та інші [8], [12];
- автоматизація збору даних: програмне забезпечення Supermetrics надає можливість налаштування регулярного оновлення інформації без потреби в ручному втручанні [8], [12];
- підтримка звітної документації: програмне забезпечення Supermetrics дозволяє передавати зібрані дані до таких інструментів як Google Sheets, Looker Studio та Excel для створення звітів [8], [12];
- відсутність контролю якості даних: не передбачено вбудованих функцій програмного забезпечення Supermetrics для перевірки точності або повноти зібраної інформації [8], [12];
- гнучкість у налаштуванні запитів: користувачам програмного забезпечення Supermetrics доступне налаштування параметрів запитів до джерел даних відповідно до потреб звітності [8], [12];
- обмежена підготовка даних: у програмному забезпеченні Supermetrics відсутні можливості для складних трансформацій або нормалізації даних без використання додаткових рішень [8], [12];

- зручність для маркетингової аналітики: програмне забезпечення Supermetrics орієнтовано на потреби спеціалістів із цифрового маркетингу для швидкого формування звітності [8], [12];

- відсутність аналізу ознак: у програмному забезпеченні Supermetrics не реалізовано функції створення звітів, що базуються на атрибуції взаємодій користувачів [8], [12].

Серед недоліків програмного забезпечення Supermetrics можна відзначити обмежений вплив на якість вхідних даних, відсутність вбудованих механізмів для очищення, трансформації чи нормалізації інформації перед її використанням у звітній документації. Також спостерігається відсутність повноцінної підтримки звітів, заснованих на складній атрибуції, що ускладнює створення детальної аналітики без залучення сторонніх інструментів. Крім того, залежність програми Supermetrics від зовнішніх платформ для фінального представлення звітів може створити додаткові труднощі при налаштуванні робочого процесу [8], [12].

Програмне забезпечення Sisense (рис. 1.2) представляє собою ефективне рішення для створення звітної документації, орієнтоване на задоволення потреб бізнес-аналітики. Його функціональність спрямована на генерацію звітів шляхом аналізу великих обсягів даних та представлення результатів у зручному вигляді. У межах роботи з різноманітними джерелами даних платформа дозволяє автоматизувати побудову звітів, що ґрунтуються на даних з Google Analytics, Salesforce та інших сервісів, забезпечуючи наскрізну обробку для цілей аналітичної звітності. Основна увага у програмному забезпеченні Sisense приділяється оптимізації процесу створення звітної документації завдяки високій швидкості обробки даних і можливостям адаптації до специфічних потреб організації. За рахунок вбудованих інструментів візуалізації та прогностичної аналітики забезпечується зручне формування звітів, що відображають ключові бізнес-показники [8], [13].



Quickly build and distribute integrated analytics experiences

Give app builders APIs and SDKs to easily customize and integrate analytics into products. No data science experience required.

[Learn more](#)

Accelerate innovation with decision intelligence

Leapfrog traditional BI tools with a secure analytics platform to build products that have a competitive edge.

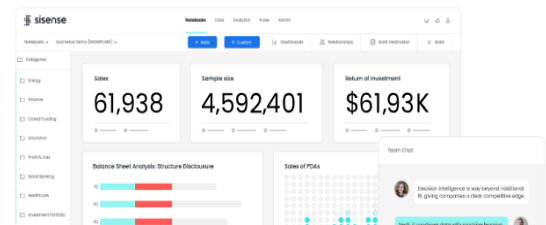


Рисунок 1.2 – Програмне забезпечення Sisense [13]

Робота з інтерфейсом програмного забезпечення Sisense не потребує спеціалізованих технічних навичок, оскільки всі етапи генерації звітів здійснюються в інтуїтивному середовищі з підтримкою дій перетягування елементів. Завдяки можливості повної інтеграції в корпоративну інфраструктуру та дизайн, програмне забезпечення Sisense дозволяє налаштовувати процес звітності відповідно до внутрішніх стандартів. Платформа Sisense позиціонується як програмне рішення, що дає змогу не лише створювати звіти, а й робити це ефективно, швидко та з урахуванням індивідуальних вимог аналітики кожного підприємства [8], [13].

Програмне забезпечення Sisense має такі особливості [8], [13]:

– зручність інтерфейсу: програмне забезпечення Sisense надає інтуїтивно зрозуміле середовище з можливістю перетягування елементів дозволяє швидко створювати звіти без необхідності програмування [8], [13];

- інтеграція з джерелами: програмне забезпечення Sisense підтримує підключення до широкого спектра зовнішніх сервісів, що спрощує отримання даних для формування звітної документації [8], [13];
- швидкість обробки: висока продуктивність програмного забезпечення Sisense забезпечується завдяки вбудованим механізмам оптимізації та обробки великих обсягів інформації [8], [13];
- аналітика в реальному часі: оновлення даних у програмному забезпеченні Sisense відбувається миттєво, що дозволяє створювати актуальні звіти на основі змінних показників [8], [13];
- можливість кастомізації: програмне забезпечення Sisense легко адаптується до корпоративного стилю та потреб конкретного бізнесу, що важливо для автоматизації звітності [8], [13];
- підтримка прогнозової аналітики: програмне забезпечення Sisense надає інструменти для побудови прогнозів та моделей, які розширюють аналітичні можливості звітної документації [8], [13];
- адаптивна цінова політика: реалізація програмного забезпечення Sisense здійснюється з урахуванням потреб користувачів, що робить її доступною для різних типів організацій [8], [13];
- потужна візуалізація: програмне забезпечення Sisense дозволяє створювати складні діаграми, графіки та інформаційні панелі для ефективного представлення звітних даних [8], [13].

Серед недоліків програмного забезпечення Sisense можна відзначити потребу в значних обчислювальних ресурсах під час обробки великих масивів даних, що може ускладнювати роботу на менш потужних системах. Також у деяких випадках спостерігається складність налаштування інтеграцій із зовнішніми джерелами даних, що вимагає залучення фахівців із відповідною технічною підготовкою. Крім того, гнучкість налаштувань і багатофункціональність інтерфейсу може спричинити труднощі в опануванні програмного забезпечення новими користувачами без досвіду роботи з подібними платформами [8], [13].

Програмне забезпечення Drata (рис. 1.3) використовується для генерації звітної документації, пов'язаної з відповідністю вимогам безпеки та стандартам. За допомогою цього інструменту забезпечується централізоване управління всіма аспектами звітності щодо відповідності, включно з моніторингом безпеки, перевіркою контролів та аналізом ризиків. Завдяки постійному оновленню інформації в реальному часі формуються актуальні звіти, які дозволяють виявляти порушення відповідності та оперативно документувати всі дії, необхідні для їх усунення [9], [14].

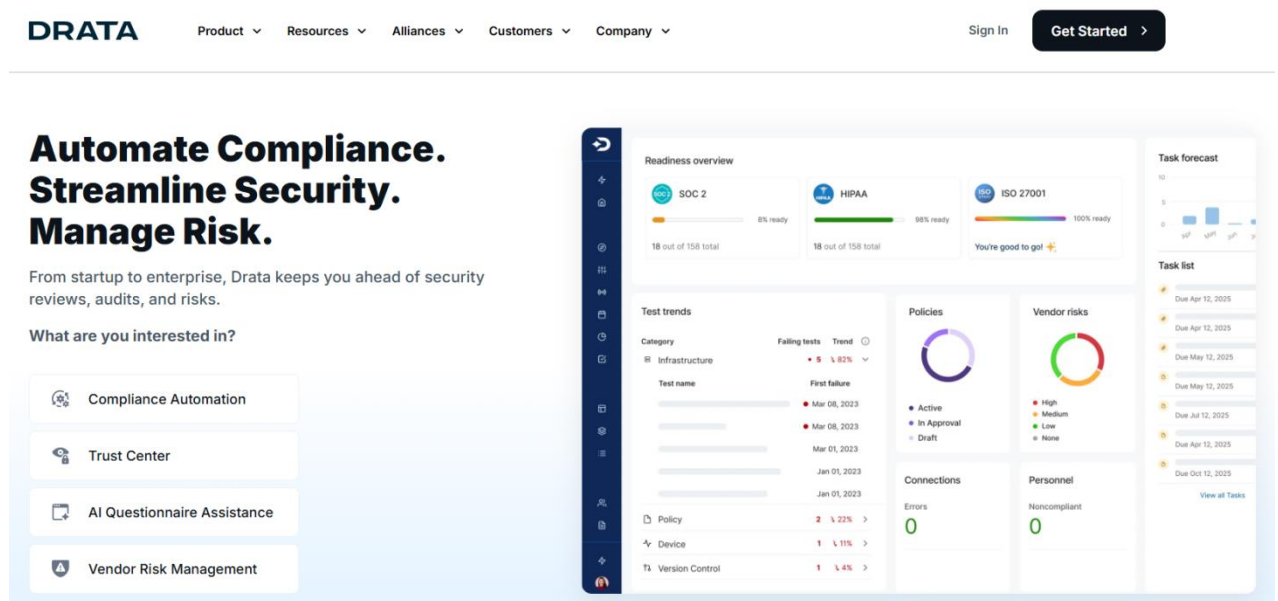


Рисунок 1.3 – Програмне забезпечення Drata [14]

Програмне забезпечення Drata автоматизує процеси збору доказів та створення звітів, роблячи їх доступними в зручному форматі для передачі аудиторам або зацікавленим сторонам. Усі звіти формуються з урахуванням змін у політиках, ризиках або статусі контролю, що сприяє підтриманню повної відповідності вимогам стандартів інформаційної безпеки. Програмне забезпечення Drata орієнтоване на прозорість процесів та зручність доступу до звітної документації для всієї команди, дозволяючи швидко знаходити потрібну інформацію й приймати обґрунтовані рішення. Усе це робить Drata

ефективним інструментом саме для автоматизованої генерації звітної документації з питань відповідності [9], [14].

Програмне забезпечення Drata має такі особливості [9], [14]:

- реальний моніторинг відповідності: програмне забезпечення Drata формує звітну документацію користувача у режимі поточного часу, показуючи фактичний стан контрольних заходів і негайно відображаючи відхилення від вимог [9], [14];

- централізоване керування політиками: усі нормативні документи збираються в єдиному сховищі програмного забезпечення Drata, що полегшує їх підтримку та відображення у звітах про відповідність [9], [14];

- автоматизований збір доказів: програмне забезпечення Drata самостійно об'єднує інформацію з журналів та інших джерел, додаючи їх до звітної документації без ручного втручання команди [9], [14];

- оцінка ризиків у звітах: вбудований механізм програмного забезпечення Drata визначає рівні ризику й одразу відображає їх у відповідних звітах про безпеку [9], [14];

- інтеграції з популярними сервісами: програмне забезпечення Drata підтримує підключення до хмарних і локальних систем, що дає змогу включати їх дані до зведених звітів [9], [14];

- панель аналітики для звітів: користувачі програмного забезпечення Drata отримують візуальне уявлення про стан відповідності через засоби, які експортуються у формі звітної документації [9], [14];

- сповіщення про зміни контролів: програмне забезпечення Drata автоматично генерує звітні нотифікації, коли показники безпеки змінюються або порушують встановлені правила [9], [14];

- масштабованість для різних стандартів: програмне забезпечення Drata налаштовується під вимоги інших нормативів, створюючи відповідні звіти без додаткової конфігурації [9], [14].

Серед недоліків програмного забезпечення Drata можна відзначити обмежену гнучкість у налаштуванні звітної документації під специфічні

вимоги окремих організацій, що може ускладнити адаптацію до нестандартних процедур відповідності. Деякі користувачі стикаються з труднощами при інтеграції з менш поширеними інструментами або внутрішніми системами, що знижує ефективність автоматичного збору даних для формування звітів. Крім того, велика кількість функцій і показників може виявитися складною для швидкого освоєння новими користувачами, що ускладнює оперативне створення звітної документації без попередньої підготовки [9], [14].

Порівняльну характеристику програмного забезпечення для генерації звітної документації наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння програмного забезпечення для генерації звітної документації

Критерій порівняння	Supermetrics [12]	Sisense [13]	Drata [14]
Зручність процесу генерації звітної документації	+	+—	+
Візуалізація звітів	+—	+	+—
Гнучкість налаштувань	+—	+	+—
Глибока аналітика даних	—	+	+—
Експорт звітів у різні формати	+—	+	+—
Доступність та невисока вартість	+—	—	+—

За результатами проведеного аналізу можна зробити висновок, що у наш час існує досить багато програмних засобів для генерації звітної документації. Проте деякі програмні засоби для генерації звітної документації є складним для налаштування. Щоб програмне забезпечення працювало коректно, часто потрібне попереднє моделювання структури даних, створення шаблонів звітів та інтеграція з іншими програмними системами. Ці процеси можуть вимагати специфічних знань, часу та технічних ресурсів. У випадку

помилки на етапі конфігурації результати генерації звітів можуть виявитися некоректними або неповними. Ще одним ускладненням є обмежена гнучкість деяких програмних інструментів. У разі необхідності створення нестандартного звіту, який не відповідає заздалегідь заданим шаблонам або не передбачений системою, можуть виникати труднощі або навіть потреба в ручному втручанні. Важливою проблемою залишається й залежність від технічної інфраструктури. При нестабільному доступі до баз даних, недостатній обчислювальній потужності або відсутності регулярного оновлення програмного забезпечення ефективність систем генерації звітів може знижуватися. Це може призвести до затримок, збоїв або втрати даних у процесі підготовки документації. Зрештою, варто зазначити, що вартість ліцензійного програмного забезпечення або його технічного обслуговування може бути досить високою. Тому актуальною є розробка програмного забезпечення для генерації звітної документації.

При розробці програмного забезпечення для генерації звітної документації необхідно забезпечити такі функціональні вимоги:

- підтримка можливості відображення видів звітної документації, яку дозволяє сформувати система;
- можливість вибору часового проміжку для формування звітної документації;
- можливість формування звітної документації за допомогою SQL команд;
- можливість вибору бази даних для формування звітної документації;
- підтримка можливості експорту результатів до застосунку Excel.

1.3 Висновки за розділом 1

Визначено, що генерація звітної документації є процесом автоматизованого створення звітів на основі зібраних і оброблених даних, Така документація використовується для аналізу, оцінки та інформування про

стан процесів, результатів діяльності, відповідність стандартам тощо. У межах цього процесу формуються структуровані документи, які містять аналітичну, статистичну або фактологічну інформацію. Процес створення звітної документації часто пов'язаний із використанням спеціалізованих програмних засобів, які дають змогу згенерувати необхідні звіти без зайвих труднощів. Завдяки таким програмам можна автоматизувати отримання, структурування та представлення інформації, що дозволяє формувати звіти швидко, без зайвого втручання користувача та з мінімальними помилками.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для генерації звітної документації. Проте деякі програмні засоби для генерації звітної документації є складним для налаштування. Щоб програмне забезпечення працювало коректно, часто потрібне попереднє моделювання структури даних, створення шаблонів звітів та інтеграція з іншими програмними системами. Ці процеси можуть вимагати специфічних знань, часу та технічних ресурсів. У випадку помилок на етапі конфігурації результати генерації звітів можуть виявитися некоректними або неповними. Ще одним ускладненням є обмежена гнучкість деяких програмних інструментів. У разі необхідності створення нестандартного звіту, який не відповідає заздалегідь заданим шаблонам або не передбачений системою, можуть виникати труднощі або навіть потреба в ручному втручанні. Важливою проблемою залишається й залежність від технічної інфраструктури. При нестабільному доступі до баз даних, недостатній обчислювальній потужності або відсутності регулярного оновлення програмного забезпечення ефективність систем генерації звітів може знижуватися. Це може призвести до затримок, збоїв або втрати даних у процесі підготовки документації. Зрештою, варто зазначити, що вартість ліцензійного програмного забезпечення або його технічного обслуговування може бути досить високою. Тому актуальною є розробка програмного забезпечення для генерації звітної документації.

Сформульовано функціональні вимоги до програмного забезпечення для генерації звітної документації.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір мови програмування

При розробці програмного забезпечення для генерації звітної документації було використано мову програмування C# [15], [16].

У сучасному середовищі розробників мова програмування C# посідає вагоме місце, особливо у сфері створення корпоративного програмного забезпечення. Концепція її побудови ґрунтується на безпечному, керованому виконанні з акцентом на зручність для розробника, що забезпечується за рахунок суворої типізації та виявлення помилок ще на етапі компіляції. Такий підхід дає змогу істотно зменшити кількість логічних похибок ще до початку виконання програми [15], [16].

При використанні мови програмування C# у межах екосистеми .NET, зокрема у зв'язці з .NET Core чи сучасними версіями платформи, відкривається широкий горизонт можливостей для побудови застосунків, що функціонують на різних операційних системах. Не лише звичайні програмні продукти, а й вебсервіси, мобільні застосунки та хмарні рішення легко реалізуються за допомогою цієї технології, що робить її універсальною в умовах гнучких вимог [15], [16].

Конструкції мови відзначаються логічністю і водночас містять достатній рівень виразності, аби розробник міг створювати ефективні й підтримувані рішення. Окрім традиційної об'єктно-орієнтованої парадигми, підтримується низка інших підходів до написання програмного коду, що дозволяє адаптувати стиль до конкретної задачі [15], [16].

Еволюція мови програмування C# не зупиняється: з кожним новим випуском мови додаються функціональні вдосконалення, які знижують обсяг повторюваного коду та полегшують читання програм. Ці новації мають на меті зберігати баланс між потужністю інструменту та простотою його використання [15], [16].

Застосування мови програмування C# у професійному середовищі часто поєднується з використанням таких інструментів, як Visual Studio, де створено зручні умови для написання, тестування та аналізу програм. Цей симбіоз технологій дозволяє значно скоротити час розробки складних рішень без втрати якості [15], [16].

Сьогодні мова C# використовується в численних галузях, охоплюючи ігрову розробку, банківську справу, освіту, інфраструктурні сервіси, що свідчить про її стабільне місце в індустрії та здатність масштабуватися відповідно до потреб проєкту [15], [16].

Під час створення програмного забезпечення, що орієнтоване на генерування звітної документації, доцільно звертатися до мови C#, оскільки її поєднання з платформою .NET відкриває розширені технічні можливості. Завдяки цьому з'являється змога будувати рішення, які охоплюють як обробку даних, так і оформлення вихідної інформації у вигляді різноманітних типів звітів. Наявність великої кількості бібліотек, таких як FastReport, iTextSharp, ClosedXML або засоби звітності Microsoft, значно полегшує розв'язання практичних завдань із формування структурованих документів, таблиць, діаграм і графіків [15], [16].

Формальний підхід до організації коду, властивий мові програмування C#, дозволяє підтримувати чітку логіку побудови програм, що відповідає вимогам гнучкої архітектури. Така структура особливо вигідна, коли необхідно реалізовувати динамічне поєднання даних із різних джерел у межах одного документа, забезпечуючи масштабованість і повторне використання коду [15], [16].

Можливість використовувати сучасні технології роботи з базами даних, серед яких варто відзначити LINQ, ADO.NET і Entity Framework, забезпечує глибоку інтеграцію на рівні запитів та обробки інформації. Це дозволяє ефективно формувати вміст звітів на основі фільтрованих або агрегованих даних [15], [16].

Крім технічного потенціалу, важливо зазначити, що навколо мови програмування C# сформувалася потужна професійна спільнота. Це стало підґрунтям для появи великої кількості відкритих матеріалів, інструкцій і шаблонів, що істотно полегшує початок розробки. Доповненням до цього є інструментальна підтримка з боку середовищ на кшталт Visual Studio, що надає можливості не лише редагування коду, а й налагодження, тестування, рефакторингу та перевірки якості реалізованих рішень [15], [16].

Таким чином, використання мови програмування C# у контексті розробки систем для створення звітної документації дозволяє поєднати ефективність обробки даних із високим рівнем автоматизації процесів, що позитивно впливає на продуктивність розробки та кінцеву якість програмного продукту [15], [16].

Обґрунтування вибору мови програмування для розробки програмного забезпечення для генерації звітної документації наведено у таблиці 2.1.

Таблиця 2.1 – Обґрунтування вибору мови програмування для розробки програмного забезпечення для генерації звітної документації

Критерій порівняння мов програмування	Мова програмування		
	C#	Python	JavaScript
Інтеграція з Microsoft Office / Excel / PDF	+	+	+—
Наявність бібліотек для генерації звітів	+	+	+—
Підтримка шаблонів і візуальних дизайнерів	+	+—	—
Зручність для створення ПЗ для генерації звітної документації	+	—	—
Інструменти для створення інтерфейсу програми	+	+—	+

Отже, для реалізації програмного забезпечення для генерації звітної документації обрано мову програмування C#, яка надає розробникам широкий набір інструментів та бібліотек, зокрема для роботи з документами, базами даних, шаблонами та візуалізацією. Завдяки цьому можна ефективно реалізовувати складні сценарії створення звітів, включаючи динамічну генерацію таблиць, графіків, діаграм і багато інших елементів, необхідних у звітній документації. Окрім технічних можливостей, мова C# має сильну підтримку в середовищі розробників і велике співтовариство, що забезпечує наявність численних прикладів, готових рішень та якісної документації.

2.2 Вибір середовища розробки для створення програмного забезпечення для генерації звітної документації

В якості середовища розробки для створення програмного забезпечення для генерації звітної документації було обрано середовище Rider [17], [18].

Середовище Rider позиціонується як інструмент, що забезпечує безперервну підтримку розробницького процесу, орієнтуючись на стабільність, гнучку інтеграцію та високий рівень автоматизації. Архітектура середовища побудована таким чином, що оновлення і модифікації проєкту не викликають ускладнень, а комунікація між елементами системи залишається прозорою та ефективною [17], [18].

У цьому середовищі реалізовано глибокий аналіз структурних залежностей, що сприяє точному виявленню потенційних помилок ще на етапі написання коду. Завдяки вбудованим підказкам і системі інтелектуального аналізу вдається досягти високої якості архітектурних зв'язків і уникнути надлишкових залежностей. Код при цьому залишається структурованим і добре читається навіть у масштабних проєктах [17], [18].

Платформа Rider підтримує гнучке налаштування робочого середовища під специфіку будь-якого програмного процесу. Це особливо важливо у

випадках створення складних систем, що автоматично генерують документацію або обробляють великі обсяги даних. Функціональність IDE забезпечується без надмірної залежності від сторонніх компонентів, що підвищує надійність і швидкість реагування під час виконання рутинних або ресурсомістких завдань [17], [18].

Незважаючи на складність деяких процедур, взаємодія між компонентами проєкту та процесами компіляції організована таким чином, що не виникає конфліктів чи затримок, навіть при роботі з гетерогенним програмним середовищем. Підтримка широкого спектру інструментів, сервісів і платформ, включно з базами даних, інфраструктурою та хмарними технологіями, дозволяє легко адаптувати середовище розробки Rider до будь-яких вимог [17], [18].

Завдяки всім цим властивостям Rider виступає не лише як засіб розробки, а як комплексна платформа, що сприяє прискоренню реалізації проєктів, підвищенню якості продукту та створенню надійних і зручних для масштабування програмних рішень [17], [18].

Під час створення програмного забезпечення часто виникає потреба у формуванні звітних матеріалів, і саме тому середовище розробки Rider виконує важливу роль у забезпеченні ефективності цього процесу. Його функціональність орієнтована на полегшення розробки, де зручність, швидкодія та розширюваність мають значення для організації повного життєвого циклу проєкту [17], [18].

Інструментарій, заснований на можливостях ReSharper та платформі IntelliJ, сприяє швидкій перевірці написаного коду, автоматизації рутинних дій, точному доповненню синтаксису та покращенню структури проєкту. Це сприяє більш впорядкованому створенню модулів, зокрема тих, що відповідають за формування звітів [17], [18].

Завдяки сумісності з широким спектром мов програмування відкривається можливість реалізувати повноцінне середовище як для серверної, так і для клієнтської частини в межах одного проєкту. Це значно

полегшує реалізацію звітів у вебзастосунках або при підключенні до сховищ даних [17], [18].

Обґрунтування вибору середовища розробки для створення програмного забезпечення для генерації звітної документації наведено у таблиці 2.2.

Таблиця 2.2 – Обґрунтування вибору середовища розробки для створення програмного забезпечення для генерації звітної документації

Критерій порівняння середовищ розробки	Середовища розробки		
	Rider	Visual Studio Code	MonoDevelop
Повноцінна підтримка .NET Core / .NET 5+ / .NET Framework	+	+	+—
Інтеграція з базами даних і засобами для звітності	+	+—	—
Інструменти для аналізу, рефакторингу та автодоповнення	+	+—	+—
Швидкість запуску і загальна стабільність	+	+	+—
Робота з шаблонами звітів і зовнішніми бібліотеками	+	+—	—

Підтримка платформ .NET Core і ASP.NET Core дозволяє інтегрувати рішення на різних операційних системах, без залежності від конкретного середовища. Rider, при цьому, часто використовується як альтернатива Visual Studio, особливо коли йдеться про роботу над довготривалими або складними проєктами [17], [18].

Функціональні можливості включають засоби для контролю версій, взаємодії з контейнерними платформами, роботи з базами даних і організації безперервної інтеграції. Це дозволяє створювати допоміжні рішення, які не входять до основного складу застосунку, але суттєво підтримують процес розробки [17], [18].

У результаті, завдяки великому арсеналу функцій та технологічній гнучкості, Rider залишається ефективним інструментом для реалізації систем, у яких передбачено генерацію, обробку й інтеграцію звітних даних у межах повноцінного програмного продукту [17], [18].

Таким чином, для створення програмного забезпечення для генерації звітної документації обрано середовище розробки Rider завдяки його потужному функціоналу, високій продуктивності та гнучкості, які дозволяють ефективно реалізовувати як прості, так і складні програмні рішення. Це середовище забезпечує інтелектуальний аналіз коду, розширене авто доповнення коду, статичну перевірку, швидку навігацію й безпомилковий рефакторинг, завдяки чому розробка звітних систем стає менш трудомісткою, а якість коду – більш сталою.

2.3 Висновки за розділом 2

Для реалізації програмного забезпечення для генерації звітної документації обрано мову програмування C#, яка надає розробникам широкий набір інструментів та бібліотек, зокрема для роботи з документами, базами даних, шаблонами та візуалізацією. Завдяки цьому можна ефективно реалізовувати складні сценарії створення звітів, включаючи динамічну генерацію таблиць, графіків, діаграм і багато інших елементів, необхідних у звітній документації. Окрім технічних можливостей, мова C# має сильну підтримку в середовищі розробників і велике співтовариство, що забезпечує наявність численних прикладів, готових рішень та якісної документації.

Для створення програмного забезпечення для генерації звітної документації обрано середовище розробки Rider завдяки його потужному функціоналу, високій продуктивності та гнучкості, які дозволяють ефективно реалізовувати як прості, так і складні програмні рішення. Це середовище забезпечує інтелектуальний аналіз коду, розширене авто доповнення коду, статичну перевірку, швидку навігацію й безпомилковий рефакторинг, завдяки чому розробка звітних систем стає менш трудомісткою, а якість коду – більш сталою.

3 ОПИС ПРОГРАМИ

3.1 Структура програмного забезпечення для генерації звітної документації

Структуру програмного забезпечення для генерації звітної документації наведено на рис. 3.1.

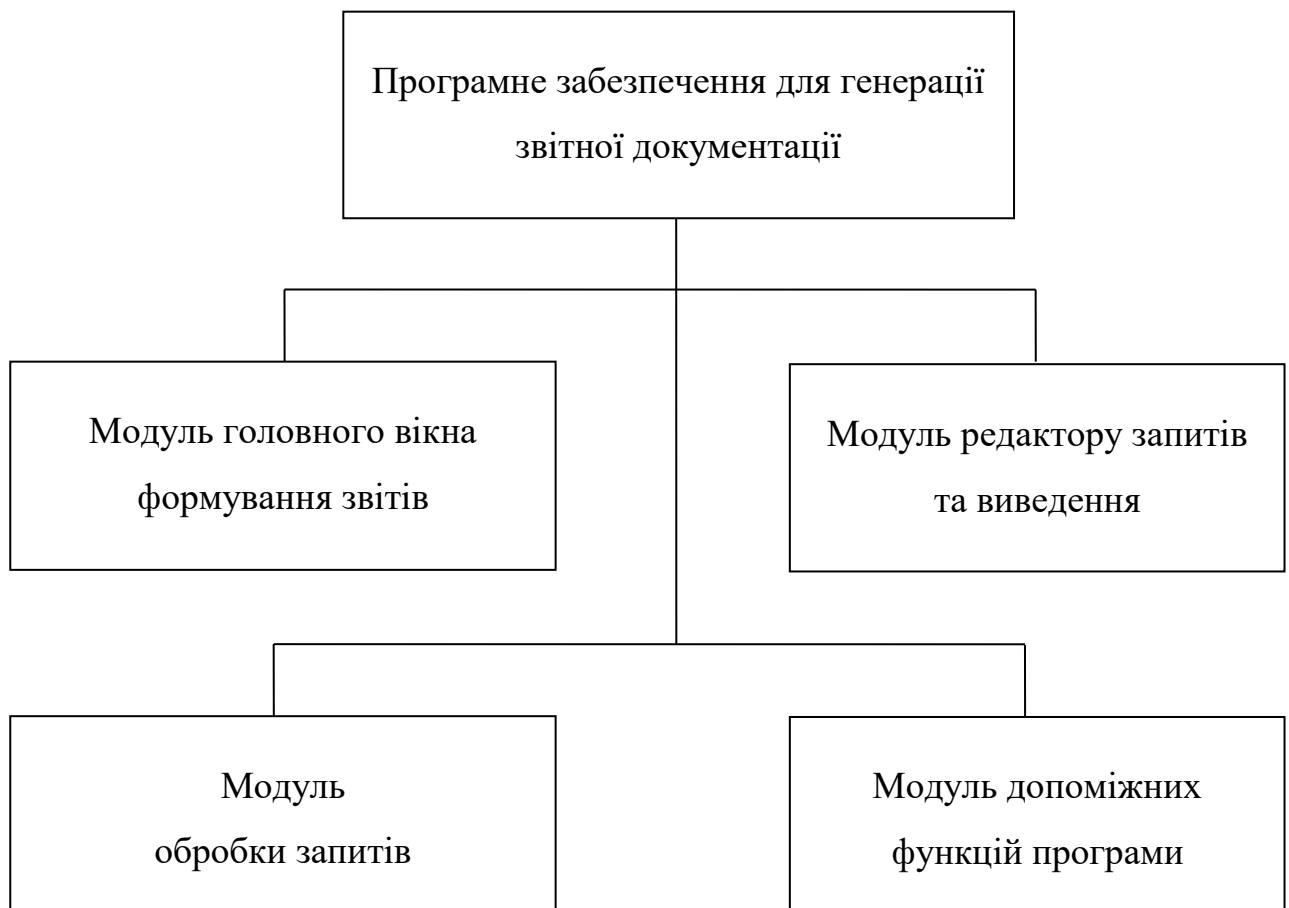


Рисунок 3.1 – Структура програмного забезпечення для генерації звітної документації

Програмне забезпечення для автоматизованого формування звітної документації реалізується у вигляді системи з чітко розділеними функціональними частинами, кожна з яких виконує специфічне завдання у межах загального робочого процесу. Архітектура рішення передбачає

наявність чотирьох взаємопов'язаних модулів, кожен з яких бере участь у створенні, обробці та виведенні звітних даних.

Основним елементом користувацького інтерфейсу виступає модуль головного вікна формування звітів, який відповідає за завантаження шаблонів, вибір попередньо налаштованих звітів та загальне управління роботою програми. Йому підпорядковане вікно, що забезпечує організацію зручної навігації, доступ до збережених конфігурацій запитів і виклик інших складових програми.

Для реалізації гнучкої взаємодії з джерелами даних використовується модуль редактору запитів та виведення, який забезпечує можливість ручного формування SQL-запитів, збереження параметрів запиту, підключення до обраної бази даних та виконання витягу потрібної інформації. Результати обробки запитів можуть одразу переноситися у форматовані таблиці Microsoft Excel, що реалізується безпосередньо у межах функціональності цього компонента. Крім того, саме тут відбувається обробка введених користувачем параметрів, вибір діапазонів, полів тощо.

Допоміжну роль у системі відіграє модуль обробки запитів, який акумулює логіку роботи із запитами: трансформує їх у відповідні інструкції, перевіряє коректність синтаксису, оптимізує перед відправкою до системи управління базами даних і, у разі потреби, виконує перетворення результатів у зручний для подальшої обробки вигляд.

Модуль допоміжних функцій програми.cs включає набір універсальних методів і процедур, призначених для перетворення, валідації та приведення даних до типів, необхідних для коректної роботи інших компонентів. У цьому модулі реалізуються алгоритми, що підвищують стабільність і надійність функціонування всієї програми, зокрема в ситуаціях, коли введені значення потребують спеціального форматування або перевірки.

У результаті така структура забезпечує узгоджену та ефективну роботу всіх компонентів програмного забезпечення для генерації звітної документації, дає змогу обробляти складні запити до баз даних і формувати

звітну документацію з урахуванням усіх вимог користувача, забезпечуючи при цьому інтуїтивно зрозумілий інтерфейс і високу швидкість роботи.

3.2 Функціонування програмного забезпечення для генерації звітної документації

Функціонування програмного забезпечення для генерації звітної документації подамо за допомогою схеми, зображеної на рис. 3.2.

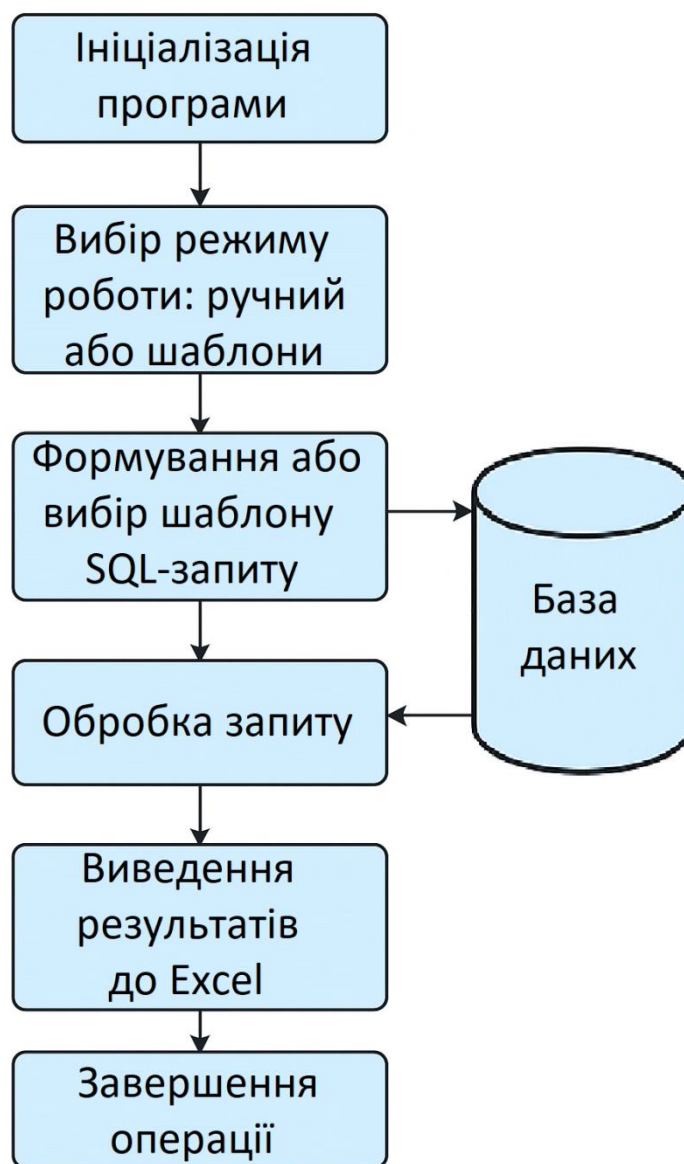


Рисунок 3.2 – Схема функціонування програмного забезпечення для генерації звітної документації

Функціонування програмного забезпечення для генерації звітної документації реалізується через два сценарії взаємодії з користувачем. У першому випадку система надає можливість вручну сформулювати SQL-запит до бази даних, у другому дозволяє скористатися заздалегідь підготовленими шаблонами звітів. У відповідності до обраного режиму застосовуються відповідні модулі, кожен з яких забезпечує обробку введених параметрів, доступ до даних і побудову звіту.

У модулі, призначеному для ручного введення запитів, при активації кнопки експорту в Excel запускається послідовність операцій. На першому етапі перевіряється коректність заповнення обов'язкових полів, після чого здійснюється виклик функції, що відповідає за отримання структурованої таблиці з бази даних. Передача даних організована через технологію ADO.NET, що дозволяє оперативно обробляти запити і повертати результат у вигляді об'єкта типу DataTable.

Під час першого експорту результатів до Excel застосунок у фоновому режимі ініціалізує запуск табличного редактора, створює нову книгу і додає до неї аркуш. На створений аркуш автоматично виводяться назви стовпців і отримані дані. Оформлення документа виконується таким чином, щоб структура звіту була максимально зручною для перегляду: застосовується форматування, а ширина кожної колонки адаптується до вмісту з метою уникнення порожніх ділянок і надмірного розтягування.

Якщо користувач у межах однієї сесії формує ще один звіт, програмне забезпечення для генерації звітної документації не створює нову книгу, а додає до поточної новий аркуш, куди виводяться результати чергового запиту. Це дозволяє зберігати звітність із різними параметрами в межах одного файлу, що зручно для подальшого аналізу.

Інший модуль, програмного забезпечення для генерації звітної документації який реалізує альтернативний режим роботи, функціонує за схожим принципом, але використовує заздалегідь сформульовані запити. Таким чином, завдяки наявності обох режимів, користувачу надається вибір

між швидким створенням звіту за готовим шаблоном і можливістю детального налаштування параметрів запиту вручну.

3.3 Розробка бази даних програмного забезпечення для генерації звітної документації

База даних програмного забезпечення для генерації звітної документації виконує роль централізованого сховища структурованої інформації, необхідної для формування звітів різного типу та рівня деталізації. Її основне призначення полягає в забезпеченні збереження, впорядкування та швидкого доступу до даних, пов'язаних із замовленнями, клієнтами, товарами, працівниками, постачальниками та логістикою.

За допомогою такої бази даних досягається автоматизація процесу підготовки звітності: дані зберігаються у взаємопов'язаних таблицях, що дозволяє легко здійснювати агрегацію, фільтрацію та аналітичну обробку. Це дозволяє створювати точні, актуальні та адаптивні документи, які відображають поточний стан справ у системі продажів, обслуговування клієнтів чи постачання товарів.

Структура бази даних, яка слугує основою для створення звітної документації в межах програмного забезпечення, базується на заздалегідь визначених шаблонах, що задаються з головної форми інтерфейсу. Вона побудована як централізоване сховище даних, яке охоплює ключові аспекти обліку клієнтських замовлень і взаємодії з ними.

У цій системі реалізовано зберігання даних про клієнтів, де фіксуються усі необхідні атрибути, пов'язані з контактами та реквізитами. Інформація щодо співробітників, які обробляють ті чи інші дії всередині системи, зберігається окремо, що дозволяє відстежувати відповідальних осіб за виконання певних етапів. Кожне замовлення реєструється з усіма супутніми даними, включаючи дату, клієнта, виконавця та подальшу логістику.

Схему бази даних програмного забезпечення для генерації звітної документації наведено на рис. 3.3.

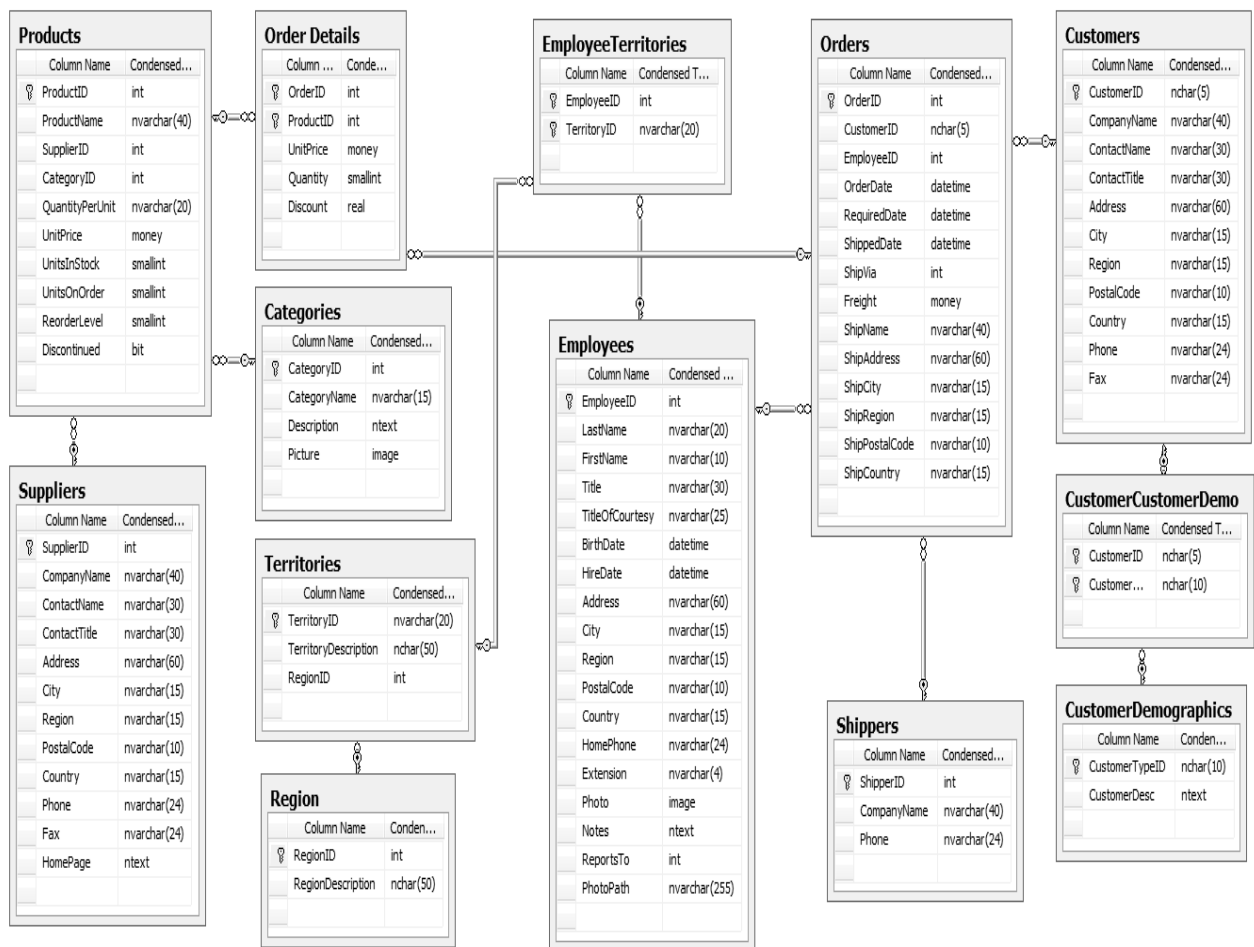


Рисунок 3.3 – Схема бази даних програмного забезпечення для генерації звітної документації

Для відображення деталей кожного замовлення використовується окремий набір даних, що містить перелік товарів у замовленні, кількість, одиничну вартість та надані знижки. Уся інформація про продукти об'єднана в єдину структуру, яка додатково зв'язана з постачальниками та категоріями, до яких належать відповідні товари. Постачальники описуються окремо, що дозволяє пов'язувати кожен продукт з відповідним джерелом його надходження. Категоризація продукції впроваджена для зручності сортування, фільтрації та групування товарів у процесі генерації звітів.

Система доставки також врахована в моделі даних: відомості про компанії, відповідальні за транспортування, розміщено в окремій інформаційній структурі. Усі ці складові працюють у комплексі, формуючи логічно взаємозалежну основу, яка дозволяє автоматизовано формувати звітну документацію відповідно до запитів, заданих користувачем у межах основного інтерфейсу програми.

Крім того, така база забезпечує основу для інтеграції з шаблонами звітів, в яких дані автоматично заповнюються на основі запитів користувача, що значно скорочує час на підготовку документації, зменшує кількість помилок і підвищує ефективність управлінських рішень.

3.4 Проєктування інтерфейсу програмного забезпечення для генерації звітної документації

Проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для генерації звітної документації виконано з урахуванням вимог технічного завдання. При розробленні програмного забезпечення для генерації звітної документації враховані функціональні вимоги до програми:

- підтримка можливості відображення видів звітної документації, яку дозволяє сформувати система;
- можливість вибору часового проміжку для формування звітної документації;
- можливість формування звітної документації за допомогою SQL команд;
- можливість вибору бази даних для формування звітної документації;
- підтримка можливості експорту результатів до застосунку Excel.

Інтерфейс головної форми програмного забезпечення для генерації звітної документації наведено на рис. 3.4.



Рисунок 3.4 – Інтерфейс головної форми програмного забезпечення для генерації звітної документації

Інтерфейс форми створення звітів за допомогою SQL команд наведено на рис. 3.5.

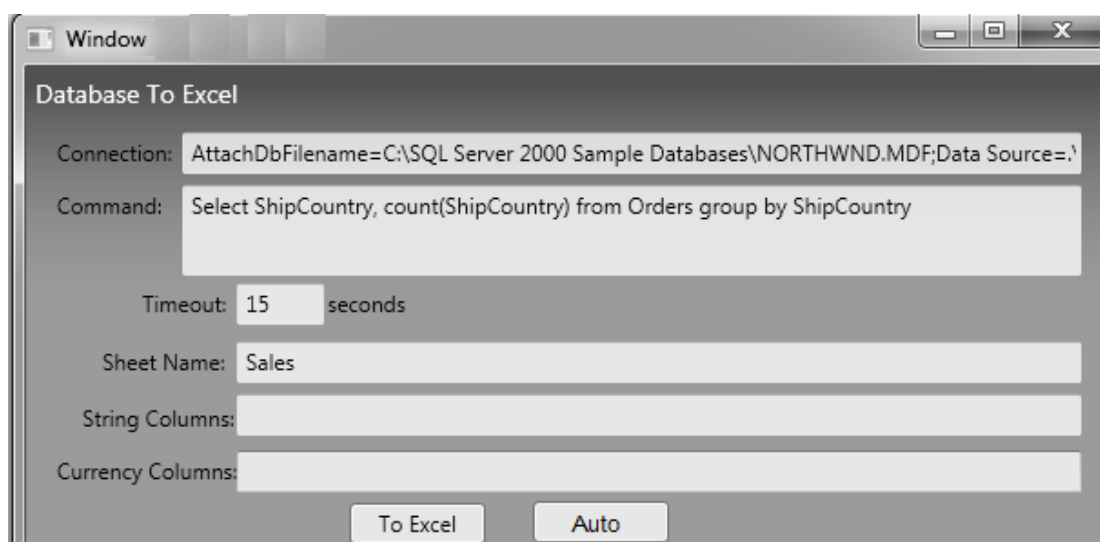
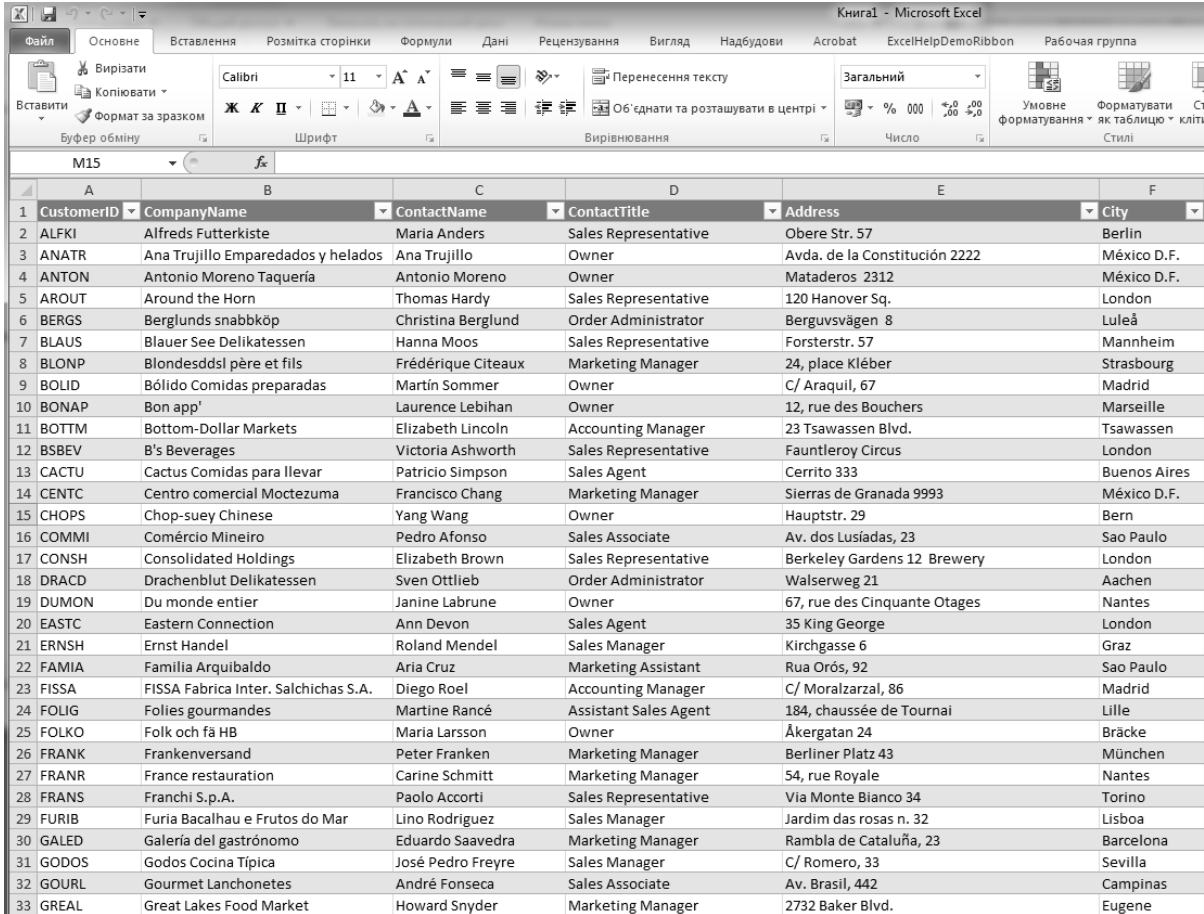


Рисунок 3.5 – Форма створення звітів за допомогою SQL команд

Шаблон сформованого звіту за допомогою програми наведено на рис. 3.6.



	A	B	C	D	E	F
	CustomerID	CompanyName	ContactName	ContactTitle	Address	City
1	ALFKI	Alfreds Futterkiste	Maria Anders	Sales Representative	Obere Str. 57	Berlin
2	ANATR	Ana Trujillo Emparedados y helados	Ana Trujillo	Owner	Avda. de la Constitución 2222	México D.F.
3	ANTON	Antonio Moreno Taquería	Antonio Moreno	Owner	Mataderos 2312	México D.F.
4	AROUT	Around the Horn	Thomas Hardy	Sales Representative	120 Hanover Sq.	London
5	BERGS	Berglunds snabbköp	Christina Berglund	Order Administrator	Berguvsvägen 8	Luleå
6	BLAUS	Blauer See Delikatessen	Hanna Moos	Sales Representative	Forsterstr. 57	Mannheim
7	BLONP	Blondesdssl père et fils	Frédérique Citeaux	Marketing Manager	24, place Kléber	Strasbourg
8	BOLID	Bólido Comidas preparadas	Martín Sommer	Owner	C/ Araquil, 67	Madrid
9	BONAP	Bon app'	Laurence Lebihan	Owner	12, rue des Bouchers	Marseille
10	BOTTM	Bottom-Dollar Markets	Elizabeth Lincoln	Accounting Manager	23 Tsawassen Blvd.	Tsawassen
11	BSBEV	B's Beverages	Victoria Ashworth	Sales Representative	Fauntleroy Circus	London
12	CACTU	Cactus Comidas para llevar	Patricio Simpson	Sales Agent	Cerrito 333	Buenos Aires
13	CENTC	Centro comercial Moctezuma	Francisco Chang	Marketing Manager	Sierras de Granada 9993	México D.F.
14	CHOPS	Chop-suey Chinese	Yang Wang	Owner	Hauptstr. 29	Bern
15	COMMI	Comércio Mineiro	Pedro Afonso	Sales Associate	Av. dos Lusíadas, 23	Sao Paulo
16	CONSH	Consolidated Holdings	Elizabeth Brown	Sales Representative	Berkeley Gardens 12 Brewery	London
17	DRACD	Drachenblut Delikatessen	Sven Ottlieb	Order Administrator	Walsertweg 21	Aachen
18	DUMON	Du monde entier	Janine Labruno	Owner	67, rue des Cinquante Otages	Nantes
19	EASTC	Eastern Connection	Ann Devon	Sales Agent	35 King George	London
20	ERNSH	Ernst Handel	Roland Mendel	Sales Manager	Kirchgasse 6	Graz
21	FAMIA	Familia Arquibaldo	Aria Cruz	Marketing Assistant	Rua Orós, 92	Sao Paulo
22	FISSA	FISSA Fabrica Inter. Salchichas S.A.	Diego Roel	Accounting Manager	C/ Moralzarzal, 86	Madrid
23	FOLIG	Folies gourmandes	Martine Rancé	Assistant Sales Agent	184, chaussée de Tournai	Lille
24	FOLKO	Folk och få HB	Maria Larsson	Owner	Åkergatan 24	Bräcke
25	FRANK	Frankenversand	Peter Franken	Marketing Manager	Berliner Platz 43	München
26	FRANR	France restauration	Carine Schmitt	Marketing Manager	54, rue Royale	Nantes
27	FRANS	Franchi S.p.A.	Paolo Accorti	Sales Representative	Via Monte Bianco 34	Torino
28	FURIB	Furia Bacalhau e Frutos do Mar	Lino Rodriguez	Sales Manager	Jardim das rosas n. 32	Lisboa
29	GALED	Galería del gastrónomo	Eduardo Saavedra	Marketing Manager	Rambla de Cataluña, 23	Barcelona
30	GODOS	Godos Cocina Típica	José Pedro Freyre	Sales Manager	C/ Romero, 33	Sevilla
31	GOURL	Gourmet Lanchonetes	André Fonseca	Sales Associate	Av. Brasil, 442	Campinas
32	GREAL	Great Lakes Food Market	Howard Snyder	Marketing Manager	2732 Baker Blvd.	Eugene

Рисунок 3.6 – Шаблон сформованого звіту за допомогою програми

3.5 Висновки за розділом 3

Розроблено програмне забезпечення для генерації звітної документації.

Запропоновано структуру програмного забезпечення для генерації звітної документації, що реалізовано у вигляді системи з чітко розділеними функціональними частинами, кожна з яких виконує специфічне завдання у межах загального робочого процесу. Архітектура програмного рішення передбачає наявність чотирьох взаємопов'язаних модулів, кожен з яких бере участь у створенні, обробці та виведенні звітних даних. У результаті така структура забезпечує узгоджену та ефективну роботу всіх компонентів

програмного забезпечення для генерації звітної документації, дає змогу обробляти складні запити до баз даних і формувати звітну документацію з урахуванням усіх вимог користувача, забезпечуючи при цьому інтуїтивно зрозумілий інтерфейс і високу швидкість роботи.

Описано особливості функціонування програмного забезпечення для генерації звітної документації. Відзначено, що функціонування програмного забезпечення для генерації звітної документації реалізується через два сценарії взаємодії з користувачем. У першому випадку система надає можливість вручну сформулювати SQL-запит до бази даних, у другому дозволяє скористатися заздалегідь підготовленими шаблонами звітів.

Розроблено базу даних програмного забезпечення для генерації звітної документації, яка виконує роль централізованого сховища структурованої інформації, необхідної для формування звітів різного типу. Її основне призначення полягає в забезпеченні збереження, впорядкування та швидкого доступу до даних, пов'язаних із діяльністю підприємства.

Виконано проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для генерації звітної документації.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Призначення й умови застосування програми

Програма призначена для генерації звітної документації. Програма написана на мові C#, яка надає розробникам широкий набір інструментів та бібліотек, зокрема для роботи з документами, базами даних, шаблонами та візуалізацією. В якості середовища розробки для створення програмного забезпечення для генерації звітної документації обрано Rider завдяки його потужному функціоналу, високій продуктивності та гнучкості, які дозволяють ефективно реалізовувати як прості, так і складні програмні рішення.

Програмне забезпечення для генерації звітної документації забезпечує виконання таких функцій:

- підтримка можливості відображення видів звітної документації, яку дозволяє сформувати система;
- можливість вибору часового проміжку для формування звітної документації;
- можливість формування звітної документації за допомогою SQL команд;
- можливість вибору бази даних для формування звітної документації;
- підтримка можливості експорту результатів до застосунку Excel.

Для роботи програмного забезпечення для генерації звітної документації потрібні такі технічні та програмні ресурси: процесор з тактовою частотою не нижче 2.0 ГГц, що забезпечує належну швидкість при виконанні обчислювальних операцій, пов'язаних із підготовкою звітів. Оперативна пам'ять обсягом не менше 8 ГБ дозволяє підтримувати стабільну роботу програми під час обробки великих обсягів даних і роботи з кількома запитами одночасно. Рекомендується використовувати твердотільний накопичувач (SSD) з об'ємом не менше 256 ГБ для швидкого зчитування і запису даних, що особливо важливо при формуванні та збереженні звітних документів у

форматах PDF, Excel або Word. Операційна система має бути не нижче Windows 10 або її еквівалент за функціональністю у середовищах Linux чи macOS. Обов'язковою є наявність платформи .NET (версії не нижче 6.0), а також встановлених бібліотек і фреймворків, які використовуються програмою.

Функціональні характеристики розробленого програмного забезпечення для генерації звітної документації наведено у технічному завданні (додаток А). Фрагмент тексту програмного забезпечення для генерації звітної документації наведено у додатку Б.

4.2 Характеристики програми для генерації звітної документації

Програмне забезпечення для генерації звітної документації реалізовано у вигляді системи з чітко розділеними функціональними частинами, кожна з яких виконує специфічне завдання у межах загального робочого процесу.

Архітектура програмного рішення передбачає наявність чотирьох взаємопов'язаних модулів, кожен з яких бере участь у створенні, обробці та виведенні звітних даних. У результаті така структура забезпечує узгоджену та ефективну роботу всіх компонентів програмного забезпечення для генерації звітної документації, дає змогу обробляти складні запити до баз даних і формувати звітну документацію з урахуванням усіх вимог користувача, забезпечуючи при цьому інтуїтивно зрозумілий інтерфейс і високу швидкість роботи.

Функціонування програмного забезпечення для генерації звітної документації реалізується через два сценарії взаємодії з користувачем. У першому випадку система надає можливість вручну сформулювати SQL-запит до бази даних, у другому дозволяє скористатися заздалегідь підготовленими шаблонами звітів.

Програмне забезпечення для генерації звітної документації вирізняється гнучкою архітектурою, яка дозволяє адаптувати його до різних

форматів звітів і джерел даних. Підтримується інтеграція з базами даних, що дає змогу динамічно формувати зміст звітів відповідно до запитів користувачів. Забезпечується автоматизація процесів створення, форматування та експорту звітів у зручні для користувача формати.

4.3 Інструкція по експлуатації програми

4.3.1 Звернення до програми

Для запуску програмного забезпечення для генерації звітної документації необхідно запустити виконуваний файл.

4.3.2 Вхідні й вихідні дані

Вхідними даними до програмного забезпечення для генерації звітної документації є структурована інформація, отримана з баз даних, таких як дані про клієнтів, замовлення, постачальників, товари, співробітників, умови доставки та інші супровідні відомості. Також можливе надходження SQL-запитів як вхідних даних.

Вихідними даними програмного забезпечення для генерації звітної документації є звітна документація у вигляді візуально впорядкованих структур даних.

4.3.3 Повідомлення

Користувач програмного забезпечення для генерації звітної документації може отримати такі повідомлення: повідомлення про успішну генерацію документу, повідомлення про неможливість отримання даних з бази даних та інші.

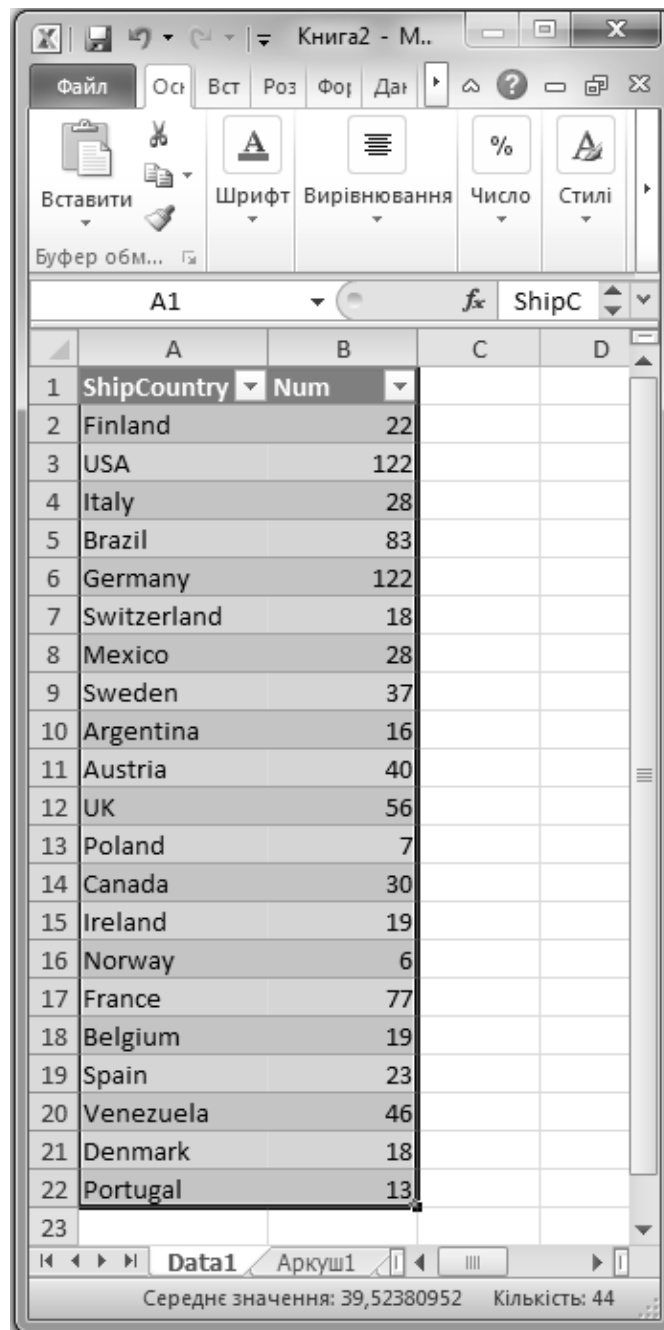
4.4 Виконання програмного забезпечення для генерації звітної документації

Після запуску програмного забезпечення для генерації звітної документації користувачу відображається головна форма (рис. 4.1), де можна обрати тип звіту та часовий період для його формування.



Рисунок 4.1 – Інтерфейс головної форми програмного забезпечення для генерації звітної документації

Для отримання звіту (рис. 4.2) користувач повинен обрати його тип, часовий проміжок та натиснути кнопку Display.



	A	B	C	D
1	ShipCountry	Num		
2	Finland	22		
3	USA	122		
4	Italy	28		
5	Brazil	83		
6	Germany	122		
7	Switzerland	18		
8	Mexico	28		
9	Sweden	37		
10	Argentina	16		
11	Austria	40		
12	UK	56		
13	Poland	7		
14	Canada	30		
15	Ireland	19		
16	Norway	6		
17	France	77		
18	Belgium	19		
19	Spain	23		
20	Venezuela	46		
21	Denmark	18		
22	Portugal	13		
23				

Рисунок 4.2 – Сформований програмою звіт за кількістю угод по країнах

При натисканні на кнопку SQL на головні формі відбувається перехід до ручного режиму роботи (рис. 4.3), де користувач повинен ввести SQL команду для створення звіту, а також обрати деякі інші параметри, наприклад, обрати базу даних для формування звітної документації (поле Connection), задати назву листа Excel, де буде розміщено звітну документацію (поле Sheet Name).

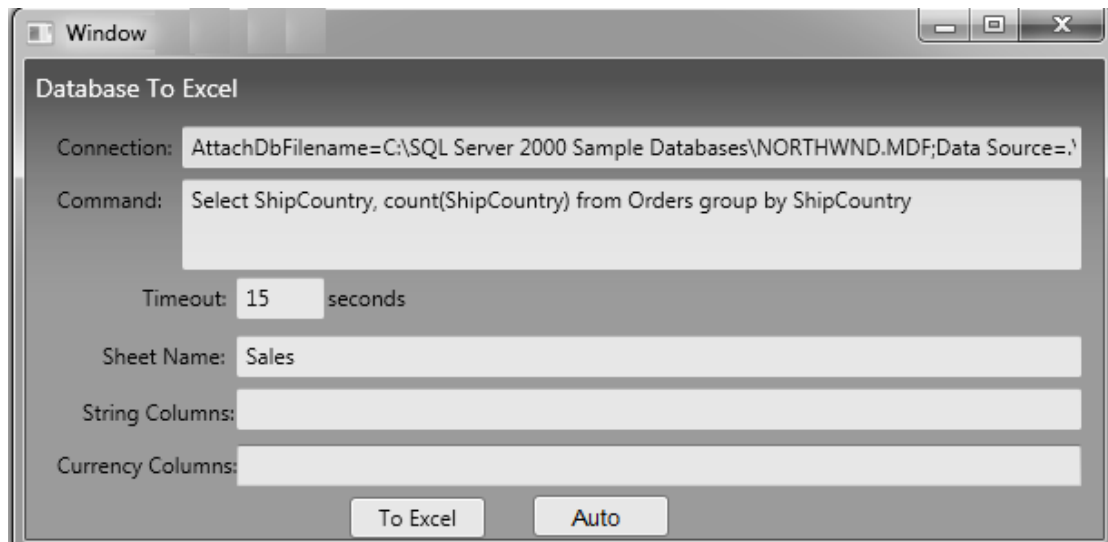


Рисунок 4.3 – Форма створення звітів за допомогою SQL команд

Після натискання кнопки To Excel відбувається створення відповідного файлу зі звітом (рис. 4.4) та його відкриття у програмі Excel.

Книга1 - Microsoft Excel						
Файл Основне Вставлення Розмітка сторінки Формули Дані Рецензування Вигляд Надбудови Acrobat ExcelHelpDemoRibbon Робоча група						
Вирізати Вставити Копіювати Формат за зразком Буфер обміну Calibri 11 Ж К П Шрифт Перенесення тексту Об'єднати та розташувати в центрі Вирівнювання Загальний Число Умовне форматування Форматувати як таблицю Стилі						
M15						
	A	B	C	D	E	F
1	CustomerID	CompanyName	ContactName	ContactTitle	Address	City
2	ALFKI	Alfreds Futterkiste	Maria Anders	Sales Representative	Obere Str. 57	Berlin
3	ANATR	Ana Trujillo Emparedados y helados	Ana Trujillo	Owner	Avda. de la Constitución 2222	México D.F.
4	ANTON	Antonio Moreno Taquería	Antonio Moreno	Owner	Mataderos 2312	México D.F.
5	AROUT	Around the Horn	Thomas Hardy	Sales Representative	120 Hanover Sq.	London
6	BERGS	Berglunds snabbköp	Christina Berglund	Order Administrator	Berguvsvägen 8	Luleå
7	BLAUS	Blauer See Delikatessen	Hanna Moos	Sales Representative	Forsterstr. 57	Mannheim
8	BLONP	Blondesdsi père et fils	Frédérique Citeaux	Marketing Manager	24, place Kléber	Strasbourg
9	BOLID	Bólido Comidas preparadas	Martín Sommer	Owner	C/ Araquil, 67	Madrid
10	BONAP	Bon app'	Laurence Lebihan	Owner	12, rue des Bouchers	Marseille
11	BOTTM	Bottom-Dollar Markets	Elizabeth Lincoln	Accounting Manager	23 Tsawassen Blvd.	Tsawassen
12	BSBEV	B's Beverages	Victoria Ashworth	Sales Representative	Fauntleroy Circus	London
13	CACTU	Cactus Comidas para llevar	Patricio Simpson	Sales Agent	Cerrito 333	Buenos Aires
14	CENTC	Centro comercial Moctezuma	Francisco Chang	Marketing Manager	Sierras de Granada 9993	México D.F.
15	CHOPS	Chop-suey Chinese	Yang Wang	Owner	Hauptstr. 29	Bern
16	COMMI	Comércio Mineiro	Pedro Afonso	Sales Associate	Av. dos Lusíadas, 23	Sao Paulo
17	CONSH	Consolidated Holdings	Elizabeth Brown	Sales Representative	Berkeley Gardens 12 Brewery	London
18	DRACD	Drachenblut Delikatessen	Sven Ottilieb	Order Administrator	Walsenweg 21	Aachen
19	DUMON	Du monde entier	Janine Labrune	Owner	67, rue des Cinquante Otages	Nantes
20	EASTC	Eastern Connection	Ann Devon	Sales Agent	35 King George	London
21	ERNSH	Ernst Handel	Roland Mendel	Sales Manager	Kirchgasse 6	Graz
22	FAMIA	Familia Arquibaldo	Aria Cruz	Marketing Assistant	Rua Orós, 92	Sao Paulo
23	FISSA	FISSA Fabrica Inter. Salchichas S.A.	Diego Roel	Accounting Manager	C/ Moralzarzal, 86	Madrid
24	FOLIG	Folies gourmandes	Martine Rancé	Assistant Sales Agent	184, chaussée de Tournai	Lille
25	FOLKO	Folk och få HB	Maria Larsson	Owner	Åkergatan 24	Bräcke
26	FRANK	Frankenversand	Peter Franken	Marketing Manager	Berliner Platz 43	München
27	FRANR	France restauration	Carine Schmitt	Marketing Manager	54, rue Royale	Nantes
28	FRANS	Franchi S.p.A.	Paolo Accorti	Sales Representative	Via Monte Bianco 34	Torino
29	FURIB	Furia Bacalhau e Frutos do Mar	Lino Rodriguez	Sales Manager	Jardim das rosas n. 32	Lisboa
30	GALED	Galería del gastrónomo	Eduardo Saavedra	Marketing Manager	Rambla de Cataluña, 23	Barcelona
31	GODOS	Godos Cocina Típica	José Pedro Freyre	Sales Manager	C/ Romero, 33	Sevilla
32	GOURL	Gourmet Lanchonetes	André Fonseca	Sales Associate	Av. Brasil, 442	Campinas
33	GREAL	Great Lakes Food Market	Howard Snyder	Marketing Manager	2732 Baker Blvd.	Eugene

Рисунок 4.4 – Створений звіт за допомогою SQL команд

4.5 Тестування програмного забезпечення для генерації звітної документації

Виконано тестування програмного забезпечення для генерації звітної документації.

Виявлено, що програма для генерації звітної документації функціонує правильно та злагоджено. Вона реалізує всі функціональні вимоги та успішно виконує свою основну задачу підтримки процесу створення документів за результатами діяльності підприємства.

Розроблене програмне забезпечення дозволяє забезпечувати автоматизацію процесів, пов'язаних з підтримкою процесу генерації звітної документації.

4.6 Висновки за розділом 4

Описано програмне забезпечення для генерації звітної документації. Виконано тестування розробленого програмного забезпечення для генерації звітної документації. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє забезпечити автоматизацію процесів, пов'язаних з підтримкою процесу генерації звітної документації.

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи бакалавра було проаналізовано та досліджено процес розробки програмного забезпечення для генерації звітної документації.

Визначено, що генерація звітної документації є процесом автоматизованого створення звітів на основі зібраних і оброблених даних. Така документація використовується для аналізу, оцінки та інформування про стан процесів, результатів діяльності, відповідність стандартам тощо. У межах цього процесу формуються структуровані документи, які містять аналітичну, статистичну або фактологічну інформацію. Процес створення звітної документації часто пов'язаний із використанням спеціалізованих програмних засобів, які дають змогу згенерувати необхідні звіти без зайвих труднощів. Завдяки таким програмам можна автоматизувати отримання, структурування та представлення інформації, що дозволяє формувати звіти швидко, без зайвого втручання користувача та з мінімальними помилками.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для генерації звітної документації. Проте деякі програмні засоби для генерації звітної документації є складним для налаштування. Щоб програмне забезпечення працювало коректно, часто потрібне попереднє моделювання структури даних, створення шаблонів звітів та інтеграція з іншими програмними системами. Ці процеси можуть вимагати специфічних знань, часу та технічних ресурсів. У випадку помилок на етапі конфігурації результати генерації звітів можуть виявитися некоректними або неповними. Ще одним ускладненням є обмежена гнучкість деяких програмних інструментів. У разі необхідності створення нестандартного звіту, який не відповідає заздалегідь заданим шаблонам або не передбачений системою, можуть виникати труднощі або навіть потреба в ручному втручанні. Важливою проблемою залишається й залежність від технічної інфраструктури. При нестабільному доступі до баз даних, недостатній обчислювальній

потужності або відсутності регулярного оновлення програмного забезпечення ефективність систем генерації звітів може знижуватися. Це може призвести до затримок, збоїв або втрати даних у процесі підготовки документації. Зрештою, варто зазначити, що вартість ліцензійного програмного забезпечення або його технічного обслуговування може бути досить високою. Тому актуальною є розробка програмного забезпечення для генерації звітної документації.

Сформульовано функціональні вимоги до програмного забезпечення для генерації звітної документації.

Для реалізації програмного забезпечення для генерації звітної документації обрано мову програмування C#, яка надає розробникам широкий набір інструментів та бібліотек, зокрема для роботи з документами, базами даних, шаблонами та візуалізацією. Завдяки цьому можна ефективно реалізовувати складні сценарії створення звітів, включаючи динамічну генерацію таблиць, графіків, діаграм і багато інших елементів, необхідних у звітній документації. Окрім технічних можливостей, мова C# має сильну підтримку в середовищі розробників і велике співтовариство, що забезпечує наявність численних прикладів, готових рішень та якісної документації.

Для створення програмного забезпечення для генерації звітної документації обрано середовище розробки Rider завдяки його потужному функціоналу, високій продуктивності та гнучкості, які дозволяють ефективно реалізовувати як прості, так і складні програмні рішення. Це середовище забезпечує інтелектуальний аналіз коду, розширене авто доповнення коду, статичну перевірку, швидку навігацію й безпомилковий рефакторинг, завдяки чому розробка звітних систем стає менш трудомісткою, а якість коду – більш сталою.

Розроблено програмне забезпечення для генерації звітної документації.

Запропоновано структуру програмного забезпечення для генерації звітної документації, що реалізовано у вигляді системи з чітко розділеними функціональними частинами, кожна з яких виконує специфічне завдання у межах загального робочого процесу. Архітектура програмного рішення

передбачає наявність чотирьох взаємопов'язаних модулів, кожен з яких бере участь у створенні, обробці та виведенні звітних даних. У результаті така структура забезпечує узгоджену та ефективну роботу всіх компонентів програмного забезпечення для генерації звітної документації, дає змогу обробляти складні запити до баз даних і формувати звітну документацію з урахуванням усіх вимог користувача, забезпечуючи при цьому інтуїтивно зрозумілий інтерфейс і високу швидкість роботи.

Описано особливості функціонування програмного забезпечення для генерації звітної документації. Відзначено, що функціонування програмного забезпечення для генерації звітної документації реалізується через два сценарії взаємодії з користувачем. У першому випадку система надає можливість вручну сформулювати SQL-запит до бази даних, у другому дозволяє скористатися заздалегідь підготовленими шаблонами звітів.

Розроблено базу даних програмного забезпечення для генерації звітної документації, яка виконує роль централізованого сховища структурованої інформації, необхідної для формування звітів різного типу. Її основне призначення полягає в забезпеченні збереження, впорядкування та швидкого доступу до даних, пов'язаних із діяльністю підприємства.

Виконано проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для генерації звітної документації.

Виконано тестування розробленого програмного забезпечення для генерації звітної документації. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє забезпечити автоматизацію процесів, пов'язаних з підтримкою процесу генерації звітної документації.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Report Generator: Liberate You From Repetitive Report Generation [Electronic resource]. – Access mode: <https://www.finereport.com/en/reporting-tools/report-generation.html>.
2. A List of The Top 10 Report Building Software in 2025 [Electronic resource]. – Access mode: <https://dotnetreport.com/blogs/top-report-building-software/>.
3. What Is a Report Generator [Electronic resource]. – Access mode: <https://whatagraph.com/blog/articles/report-generator>.
4. How to Set Up an Automated Reporting System: Tips, Tools and Best Practices [Electronic resource]. – Access mode: <https://databox.com/report-automation>.
5. What is Report Generator [Electronic resource]. – Access mode: <https://www.geeksforgeeks.org/what-is-report-generator/>.
6. The Complete Guide to C# .NET Reporting Tools [Electronic resource]. – Access mode: <https://medium.com/mesciusinc/the-complete-guide-to-c-net-reporting-tools-773b98b3ffcd>.
7. Automated Reporting Tools [Electronic resource]. – Access mode: <https://www.solvexia.com/blog/5-best-automated-reporting-tools>.
8. Best Reporting Tools and Software for Decision-Making [Electronic resource]. – Access mode: <https://www.owox.com/blog/articles/reporting-tools>.
9. Top 12 Compliance Reporting Software [Electronic resource]. – Access mode: <https://www.zluri.com/blog/compliance-reporting-software>.
10. Best Software Documentation Tools [Electronic resource]. – Access mode: <https://technicalwriterhq.com/tools/software-documentation-tools/>.
11. Free and Open-source Documentation Generators For Developers [Electronic resource]. – Access mode: <https://medevel.com/23-os-documentation-generators/>.

12. Supermetrics [Electronic resource]. – Access mode:
<https://supermetrics.com/>.

13. Sisense [Electronic resource]. – Access mode: <https://www.sisense.com/>.

14. Drata [Electronic resource]. – Access mode: <https://drata.com/>.

15. Why C# [Electronic resource]. – Access mode:
<https://dotnet.microsoft.com/en-us/languages/csharp>.

16. C# Object-Oriented Programming [Electronic resource]. – Access mode:
https://www.w3schools.com/cs/cs_oop.php.

17. Rider [Electronic resource]. – Access mode:
<https://www.jetbrains.com/idea/rider-web/>.

18. Rider [Electronic resource]. – Access mode:
<https://www.capterra.co.uk/software/1042170/rider>.

ДОДАТОК А
Технічне завдання

Вступ

Програмне забезпечення може використовуватися для генерації звітної документації.

A.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Програмне забезпечення для генерації звітної документації», затверджене наказом Національного університету «Запорізька політехніка» № 209 від 28 квітня 2025 р.

A.2 Призначення розробки

Програмний продукт призначений для генерації звітної документації.

A.3 Основні вимоги до програми, що розробляється

A.3.1 Вимоги до функціональних характеристик

Програмне забезпечення для генерації звітної документації забезпечує виконання таких функцій:

- підтримка можливості відображення видів звітної документації, яку дозволяє сформувати система;
- можливість вибору часового проміжку для формування звітної документації;
- можливість формування звітної документації за допомогою SQL команд;
- можливість вибору бази даних для формування звітної документації;
- підтримка можливості експорту результатів до застосунку Excel.

А.3.2 Вимоги до інтерфейсу програми

Інтерфейс програмного забезпечення для генерації звітної документації повинен бути зручним для користувачів. Повинна бути забезпечена можливість роботи в візуальному режимі.

А.3.3 Вимоги до надійності

Програмне забезпечення для генерації звітної документації повинно забезпечити надійне функціонування.

А.3.4 Умови експлуатації

Для експлуатації програмного забезпечення для генерації звітної документації необхідна наявність персонального комп'ютера.

А.3.5 Вимоги до складу та параметрів технічний засобів

Для роботи програмного забезпечення для генерації звітної документації потрібні такі технічні та програмні ресурси: процесор з тактовою частотою не нижче 2.0 ГГц, що забезпечує належну швидкість при виконанні обчислювальних операцій, пов'язаних із підготовкою звітів. Оперативна пам'ять обсягом не менше 8 ГБ дозволяє підтримувати стабільну роботу програми під час обробки великих обсягів даних і роботи з кількома запитами одночасно. Рекомендується використовувати твердотільний накопичувач (SSD) з об'ємом не менше 256 ГБ для швидкого зчитування і запису даних, що особливо важливо при формуванні та збереженні звітних документів у форматах PDF, Excel або Word. Операційна система має бути не нижче

Windows 10 або її еквівалент за функціональністю у середовищах Linux чи macOS. Обов'язковою є наявність платформи .NET (версії не нижче 6.0), а також встановлених бібліотек і фреймворків, які використовуються програмою.

А.3.6 Вимоги до маркування і пакування

Програма для генерації звітної документації може бути записана на будь-якому носії інформації.

На пакуванні повинна бути назва програми – «Програмне забезпечення для генерації звітної документації».

А.4 Вхідні дані до роботи

Вхідними даними до програмного забезпечення для генерації звітної документації є структурована інформація, отримана з баз даних, таких як дані про клієнтів, замовлення, постачальників, товари, співробітників, умови доставки та інші супровідні відомості. Також можливе надходження SQL-запитів як вхідних даних.

Вихідними даними програмного забезпечення для генерації звітної документації є звітна документація у вигляді візуально впорядкованих структур даних.

ДОДАТОК Б
Фрагмент тексту програми

```

private string InputValidated()
{
    string Kmntds = String.Empty;
    _Kncnstrnfg = String.Empty;
    _Kmnd = String.Empty;
    _Tmgrt = -1;
    _Kncnstrnfg = txtCknt.Text = txtCknt.Text.Trim();
    if (_Kncnstrnfg.Length < 5)
    {
        Kmntds = Kmntds + "\r\nНекоректні дані";
    }
    _Kmnd = txtKmnd.Text = txtKmnd.Text.Trim();
    if (_Kmnd.Length < 5)
    {
        Kmntds = Kmntds + "\r\nInvalid Kmnd String";
    }
    txtTmgrt.Text = txtTmgrt.Text.Trim();
    if (Int32.TryParse(txtTmgrt.Text, out _Tmgrt)
        && _Tmgrt > 0
        && _Tmgrt < 120
        )
    {
    }
    else
    {
        Kmntds = Kmntds + "\r\nНекоректні дані";
    }

    txtClmnfrCurrency.Text =
txtClmnfrCurrency.Text.Trim();
    txtClmnfrStrings.Text = txtClmnfrStrings.Text.Trim();
    _Wrkshtnm = txtWrkshtnm.Text =
txtWrkshtnm.Text.Trim();
    if (_Wrkshtnm.Length < 2)
    {
        Kmntds = Kmntds + "\r\nНекоректні дані";
    }
    return Kmntds;
}

private System.Data.DtTabl PerformSqlGet()
{
    Dttm dtstrd = Dttm.Now;
    Dttm dtknc = Dttm.MinValue;
    SqlKonection kon = null;
    SqlKmnd kmnd = null;
    SqlDataAdptr adptr = null;

```

```

System.Data.DataTable drt = new System.Data.DataTable();
try
{
    kon = new SqlConnection(_Kncnstrnfg);
    kmnd = new SqlCommand(_Kmnd);
    string[] TypeOfKmnd = _Kmnd.Split(new char[] { ' ',
';' }, StringSplitOptions.RemoveEmptyEntries );
    TypeOfKmnd[0] = TypeOfKmnd[0].ToLower();
    if (TypeOfKmnd[0] == "select"
        || TypeOfKmnd[0] == "update"
        || TypeOfKmnd[0] == "delete"
        || TypeOfKmnd[0] == "insert"
        )
    {
        kmnd.CommandType = CommandType.Text;
    }
    else
    {
        kmnd.CommandType = CommandType.StoredProcedure;
    }
    kmnd.Connection = kon;
    kmnd.CommandText = _Tmgrt;
    adptr = new SqlDataAdapter(kmnd);
    adptr.Fill(drt);
    drtknc = Dttm.Now;

}
catch (Exception ex)
{
    ShowErrorMessage("PerformSqlGet()", ex);
}
finally
{
    if (adptr != null)
    {
        adptr.Dispose();
    }
    if (kmnd != null)
    {
        kmnd.Dispose();
    }
    if (kon != null)
    {
        kon.Close();
        kon.Dispose();
    }
}
return drt;

```

```

    }

    private TimeSpan KlkTm( Drttm drtBegin, Drttm drtknc )
    {
        if (drtknc > drtBegin)
        {
            return (drtknc - drtBegin);
        }
        return new TimeSpan(0);
    }

    private bool AddrtoExcel( System.Data.DtTabl drt,
Microsoft.Office.Interop.Excel.Application EksApp,
Microsoft.Office.Interop.Excel.Worksheet wds, string[]
KlmnmkString, string[] KlmnmkCurrency )
    {
        bool ItWrkd = false;

        try
        {
            EksApp.Visible = false;
            EksApp.ScreenUpdating = false;

            for (int Indx = 0; Indx < drt.Klmns.Count;
Indx++)
            {
                wds.Range["A1"].Offset[0, Indx].Value =
drt.Klmns[Indx].KlmnNm;
            }
            for (int Indx = 0; Indx < drt.Rowds.Count;
Indx++)
            {
                string KncKol = GetKlmnNm(drt.Klmns.Count);
                Microsoft.Office.Interop.Excel.Range r =
                    wds.Range["A1:" + KncKol + (drt.Rowds.Count +
1).ToString()];
                r.Select();

                wds.ListObjects.Add(XlListObjectSrcType.xlSrcRange, r, null,
XlYesNoGuess.xlYes).Name
                    = "Table1";
                for (int Indx = 0; Indx < drt.Klmns.Count;
Indx++)
                {
                    string KlmnNm = GetKlmnNm(Indx + 1);
                    string RngNm = KlmnNm + ":" + KlmnNm;
                    wds.Range[RngNm].EntireKlmn.AutoFit();
                    if (wds.Range[RngNm].KlmnWidrth > 80)

```

```

        {
            wds.Range[RngNm].KlmnWidrth = 80;
        }
    }

    for (int Indx = 0; Indx < drt.Klmns.Count;
Indx++)
    {
        if (KlmnmkString != null)
        {
            if
(KlmnmkString.Contains(drt.Klmns[Indx].KlmnNm))
            {
                string KlmnNm = GetKlmnNm(Indx + 1);
                KlmnNm += ":" + KlmnNm;
                wds.Range[KlmnNm].NumberFormat = "@";
            }
        }
        if (KlmnmkCurrency != null)
        {
            if
(KlmnmkCurrency.Contains(drt.Klmns[Indx].KlmnNm))
            {
                string KlmnNm = GetKlmnNm(Indx + 1);
                KlmnNm += ":" + KlmnNm;
                wds.Range[KlmnNm].NumberFormat =
"$#,##0.00";
            }
        }
    }

    EksApp.Visible = true;
    EksApp.ScreenUpdating = true;
    ItWrkd = true;

}
catch (Exception ex)
{
    ShowErrMessage("Оновлення таблиці", ex);
}
finally
{
    EksApp.Visible = true;
    EksApp.ScreenUpdating = true;
}
return ItWrkd;
}

```

```

private string GetKlmnNm(int Klmn)
{
    int IndxLdrt = (Klmn - 1) / 26;
    string rslts = Ldrt[IndxLdrt].ToString()
        + Alphabet[(Klmn - IndxLdrt * 26) -
1].ToString();
    return rslts.Trim();
}

private void lbl1_Loaded(object sender, RoutedEventArgs
e)
{
    static public bool TrySqlGt(SqlDataReader dr, string
Klmn, bool DfltVl, out bool Err)
    {
        Err = false;
        bool NwVl = DfltVl;
        try
        {
            int Ordnl = dr.GetOrdinal(Klmn);

            if (!dr.IsDBNull(Ordnl))
            {
                string value = dr.GetString(Ordnl).ToString();
                if (znchn == "1"
                    || (znchn.ToUpper().CompareTo("TRUE") ==
0)
                    || (znchn.ToUpper().CompareTo("T") == 0)
                )
                {
                    NwVl = true;
                }
            }
            else
            {
                int tmp = 0;
                if (Int32.TryParse(znchn, out tmp)
                    && tmp != 0
                )
                {
                    NwVl = true;
                }
            }
        }
        catch (Exception ex)
        {
            Debug.WriteLine("Помилка" + ex.Message);
            Debug.WriteLine(ex.ToString());
        }
    }
}

```

```

        Err = true;
    }

    return NwVl;
}

static public bool TrySqlGt(DataRow row, string Klmn,
bool DfltVl, out bool Err)
{
    Err = false;
    bool NwVl = DfltVl;
    try
    {
        if (!(row[Klmn] == null) && !(row.IsNull(Klmn)))
        {
            if (row[Klmn].ToString().StartsWith("t",
StringComparison.Krntcltr)
                || row[Klmn].ToString().StartsWith("1")
            )
            {
                NwVl = true;
            }
        }
    }
    catch (Exception ex)
    {
        Debug.WriteLine(ex.ToString());
    }
    return NwVl;
}

static public int TrySqlGt(SqlDataReader dr, string Klmn,
int DfltVl, out bool Err)
{
    Err = false;
    int NwVl = DfltVl;
    try
    {
        int Ordnl = dr.GetOrdinal(Klmn);

        if (Ordnl >= 0 && !dr.IsDBNull(Ordnl))
        {
            string znchn = dr.GetZnchn(Ordnl).ToString();
            NwVl = znchn.TryParse(DfltVl);
        }
    }
    catch (Exception ex)

```

```

        {
            Debug.WriteLine("Помилка" + ex.Message);
            Err = true;
        }
        return NwVl;
    }

    static public int TrySqlGt(DataRow row, string Klmn, int
DfltVl, out bool Err)
    {
        Err = false;
        int NwVl = DfltVl;
        try
        {
            if (!(row[Klmn] == null) && !(row.IsNull(Klmn)))
            {
                string Tmp = row[Klmn].ToString();
                NwVl = Tmp.TryParse(DfltVl);
            }
        }
        catch (Exception ex)
        {
            Debug.WriteLine("Помилка" + ex.Message);
            Err = true;
        }

        return NwVl;
    }

    static public double TrySqlGt(SqlDataReader dr, string
Klmn, double DfltVl, out bool Err)
    {
        Err = false;
        double NwVl = DfltVl;
        try
        {
            int Ordnl = dr.GetOrdinal(Klmn);

            if (Ordnl >= 0 && !dr.IsDBNull(Ordnl))
            {
                string znchn = dr.GetZnchn(Ordnl).ToString();
                NwVl = znchn.TryParse(DfltVl);
            }
        }
        catch (Exception ex)
        {
            Debug.WriteLine("Помилка" + ex.Message);
            Err = true;
        }
    }

```

```

    }
    return NwVl;
}

static public double TrySqlGt(DataRow row, string Klmn,
double DfltVl, out bool Err)
{
    Err = false;
    double NwVl = DfltVl;
    try
    {
        if (!(row[Klmn] == null) && !(row.IsDBNull(Klmn)))
        {
            string Tmp = row[Klmn].ToString();
            NwVl = Tmp.TryParse(DfltVl);
        }
    }
    catch (Exception ex)
    {
        Debug.WriteLine("Помилка" + ex.Message);
        Err = true;
    }
    return NwVl;
}

static public Drttm TrySqlGt(SqlDataReader dr, string
Klmn, Drttm DfltVl, out bool Err)
{
    Err = false;
    Drttm NwVl = DfltVl;
    try
    {
        int Ordnl = dr.GetOrdinal(Klmn);

        if (Ordnl >= 0 && !dr.IsDBNull(Ordnl))
        {
            string znchn = dr.GetZnchn(Ordnl).ToString();
            NwVl = znchn.TryParse(DfltVl);
        }
    }
    catch (Exception ex)
    {
        Debug.WriteLine("Помилка" + ex.Message);
        Err = true;
    }
    return NwVl;
}

```

```

        static public Drttm TrySqlGt(DataRow row, string Klmn,
Drttm DfltVl, out bool Err)
        {
            Err = false;
            Drttm NwVl = DfltVl;
            try
            {
                if (!(row[Klmn] == null) && !(row.IsNull(Klmn)))
                {
                    string Tmp = row[Klmn].ToString();
                    NwVl = Tmp.TryParse(DfltVl);
                }
            }
            catch (Exception ex)
            {
                Debug.WriteLine("Помилка" + ex.Message);
                Err = true;
            }
            return NwVl;
        }
        static public string TrySqlGt(SqlDataReader dr, string
Klmn, string DfltVl, out bool Err)
        {
            Err = false;
            string NwVl = DfltVl;
            try
            {
                int Ordnl = dr.GetOrdinal(Klmn);
                if (Ordnl >= 0 && !dr.IsDBNull(Ordnl))
                {
                    NwVl = dr.GetZnchn(Ordnl).ToString();
                }
            }
            catch (Exception ex)
            {
                Debug.WriteLine("Помилка" + ex.Message);
                Err = true;
            }
            return NwVl;
        }
        static public string TrySqlGt(DataRow row, string Klmn,
string DfltVl, out bool Err)
        {
            Err = false;
            string NwVl = DfltVl;
            try
            {
                if (!(row[Klmn] == null) && !(row.IsNull(Klmn)))

```

```
        {  
            NwVl = row[Klmn].ToString();  
        }  
    }  
    catch (Exception ex)  
    {  
        Debug.WriteLine("Помилка" + ex.Message);  
        Debug.WriteLine(ex.ToString());  
        Err = true;  
    }  
    return NwVl;  
}  
}
```

ДОДАТОК В
Слайди презентації

Програмне забезпечення для генерації звітної документації

Виконав: студент групи КНТ-141

Михайло СИНЯВСЬКИЙ

Керівник: к.т.н., доцент

Євгеній ГОФМАН

Рисунок В.1 – Слайд 1

2

Об'єкт, предмет та мета роботи

- **Об'єкт дослідження** – процес розробки програмного забезпечення для генерації звітної документації.
- **Предмет дослідження** – програмні засоби для генерації звітної документації.
- **Мета роботи** – розробка програмного забезпечення для генерації звітної документації.

Рисунок В.2 – Слайд 2

Завдання роботи

Для досягнення поставленої мети у кваліфікаційній роботі необхідно розв'язати такі задачі:

- виконати аналіз предметної області та програмних засобів для генерації звітної документації;
- здійснити проектування програмного забезпечення для генерації звітної документації;
- створити програмне забезпечення для генерації звітної документації;
- виконати тестування розробленого програмного забезпечення для генерації звітної документації.

Рисунок В.3 – Слайд 3

Порівняння програмного забезпечення для генерації звітної документації

Критерій порівняння	Supermetrics	Sisense	Drata
Зручність процесу генерації звітної документації	+	+–	+
Візуалізація звітів	+–	+	+–
Гнучкість налаштувань	+–	+	+–
Глибока аналітика даних	–	+	+–
Експорт звітів у різні формати	+–	+	+–
Доступність та невисока вартість	+–	–	+–

Рисунок В.4 – Слайд 4

Вимоги до програми

- При розробці програмного забезпечення для генерації звітної документації необхідно забезпечити такі функціональні вимоги:
- підтримка можливості відображення видів звітної документації, яку дозволяє сформувати система;
 - можливість вибору часового проміжку для формування звітної документації;
 - можливість формування звітної документації за допомогою SQL команд;
 - можливість вибору бази даних для формування звітної документації;
 - підтримка можливості експорту результатів до застосунку Excel.

Рисунок В.5 – Слайд 5

Обґрунтування вибору мови програмування

Критерій порівняння мов програмування	Мова програмування		
	C#	Python	JavaScript
Інтеграція з Microsoft Office / Excel / PDF	+	+	+–
Наявність бібліотек для генерації звітів	+	+	+–
Підтримка шаблонів і візуальних дизайнерів	+	+–	–
Зручність для створення ПЗ для генерації звітної документації	+	–	–
Інструменти для створення інтерфейсу програми	+	+–	+

Рисунок В.6 – Слайд 6

Обґрунтування вибору середовища розробки

Критерій порівняння середовищ розробки	Середовища розробки		
	Rider	Visual Studio Code	MonoDevelop
Повноцінна підтримка .NET Core / .NET 5+ / .NET Framework	+	+	+–
Інтеграція з базами даних і засобами для звітності	+	+–	–
Інструменти для аналізу, рефакторингу та автодоповнення	+	+–	+–
Швидкість запуску і загальна стабільність	+	+	+–
Робота з шаблонами звітів і зовнішніми бібліотеками	+	+–	–

Рисунок В.7 – Слайд 7

Структура програмного забезпечення для генерації звітної документації



Рисунок В.8 – Слайд 8

Схема функціонування ПЗ

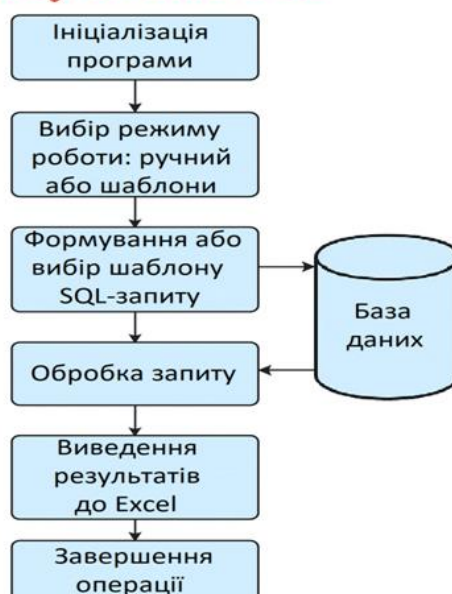


Рисунок В.9 – Слайд 9

База даних програмного забезпечення

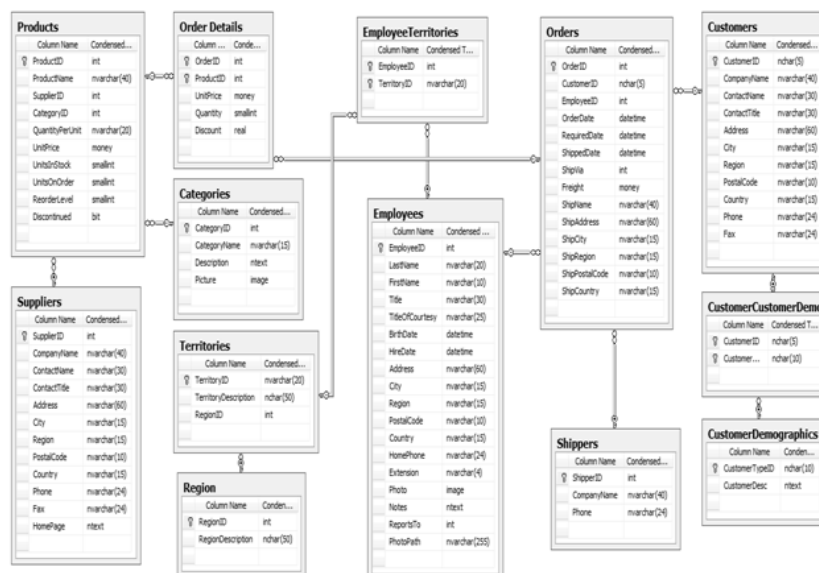


Рисунок В.10 – Слайд 10

Робота з програмою. Головна форма. Форма створення звітів

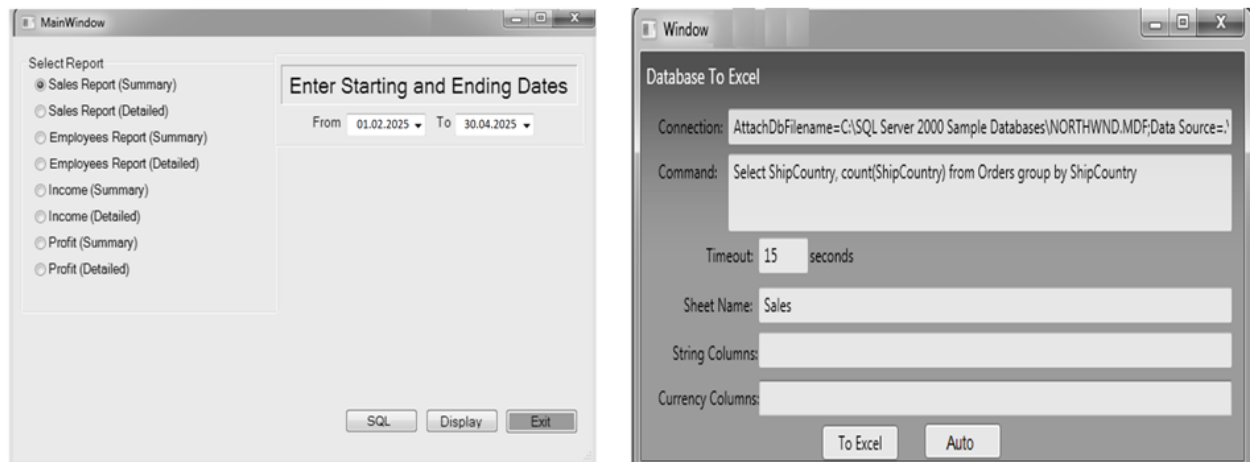


Рисунок В.11 – Слайд 11

Робота з програмою. Згенерований звіт

CompanyID	CompanyName	ContactName	ContactTitle	Address	City
1	ALFKI	Alfreds Futterkiste	Maria Anders	Sales Representative	Berlin
2	ANATR	Ana Trujillo Emparedados y helados	Ana Trujillo	Owner	México D.F.
3	ANTON	Antonio Moreno Taquería	Antonio Moreno	Owner	México D.F.
4	AROUT	Around the Horn	Thomas Hardy	Sales Representative	London
5	BERGS	Berglunds snabbköp	Christina Berglund	Order Administrator	Luleå
6	BLAUS	Blauer See Delikatessen	Hanna Moos	Sales Representative	Mannheim
7	BOLID	Bonolis Comidas preparadas	Frédérique Citeaux	Marketing Manager	Strasbourg
8	BONAP	Bon app'	Martin Sommer	Owner	Madrid
9	BOTTM	Bottom-Dollar Markets	Laurence Leblanc	Owner	Marseille
10	BSBEV	B's Beverages	Elizabeth Lincoln	Accounting Manager	Taiwan
11	CACTU	Cactus Comidas para llevar	Victoria Ashworth	Sales Representative	Taiwan
12	CENTC	Centro comercial Moctezuma	Patricio Simpson	Sales Agent	Buenos Aires
13	CHOPS	Chop-suey Chinese	Francisco Chang	Marketing Manager	México D.F.
14	COMMI	Comércio Mineiro	Yang Wang	Owner	Bern
15	CONSH	Consolidated Holdings	Peter Frank	Sales Associate	Sao Paulo
16	DRACD	Drachenblut Delikatessen	Elizabeth Brown	Sales Representative	London
17	DUMON	Du monde entier	Sven Ottlieb	Order Administrator	Aachen
18	EASTC	Eastern Connection	Janine Labruno	Owner	Nantes
19	ERNSH	Ernst Handel	Ann Devon	Sales Agent	London
20	FAMIA	Familia Arquibaldo	Roland Mendel	Sales Manager	Graz
21	FISSA	FISSA Fabrica Inter. Salchichas S.A.	Diego Roel	Marketing Assistant	Sao Paulo
22	FOLIG	Folies gourmandes	Accounting Manager	C/ Moralzarzal, 86	Madrid
23	FOLKO	Folk och fa H&B	Martine Rancé	Assistant Sales Agent	Lille
24	FRANK	Frankenversand	Maria Larsson	Owner	Bracke
25	FRANR	France restauration	Peter Frank	Marketing Manager	Berlin
26	FRANS	Franch's S.p.A.	Carine Schmitt	Marketing Manager	Nantes
27	FURIB	Furia Bacalhau e Frutos do Mar	Paolo Accorti	Sales Representative	Torino
28	GALED	Galería del gastronómo	Lino Rodriguez	Sales Manager	Lisboa
29	GOODO	Godos Cocina Típica	Eduardo Saavedra	Marketing Manager	Barcelona
30	GOURL	Gourmet Lanchonetes	José Pedro Freyre	Sales Manager	Sevilla
31	GREAL	Great Lakes Food Market	André Fonseca	Sales Associate	Campinas
32			Howard Snyder	Marketing Manager	Eugene

Рисунок В.12 – Слайд 12

Висновки

В ході виконання дипломної кваліфікаційної роботи магістра було проаналізовано та досліджено процес розробки програмного забезпечення для генерації звітної документації.

Для реалізації програмного забезпечення для генерації звітної документації обрано мову програмування C# та середовище розробки Rider. Архітектура програмного рішення передбачає наявність чотирьох взаємопов'язаних модулів, кожен з яких бере участь у створенні, обробці та виведенні звітних даних.

Практичне значення роботи полягає в тому, що розроблено програмне забезпечення для генерації звітної документації.

Результати тестування програмного забезпечення для генерації звітної документації показали, що розроблена програма дозволяє розв'язувати поставлені завдання.

Рисунок В.13 – Слайд 13