

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ**

**Факультет кібербезпеки та програмної інженерії**

**Кафедра інженерії програмного забезпечення**

ДОПУСТИТИ ДО ЗАХИСТУ  
Завідувач кафедри

Катерина НЕСТЕРЕНКО

“ \_\_\_\_ ” \_\_\_\_\_ 2023 р.

**КВАЛІФІКАЦІЙНА РОБОТА  
(ПОЯСНЮВАЛЬНА ЗАПИСКА)**

**ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ  
МАГІСТРА**

**Тема:** “Методика та програмний засіб спостереження за економічною ефективністю компаній”

**Виконавець:** Чеснокова Катерина Сергіївна

**Керівник:** к.т.н доцент Колганова Олена Олегівна

**Нормоконтролер:** ассистент Кравченко Ольга Сергіївна

Київ 2023

# НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

**Факультет** кібербезпеки та програмної інженерії

**Кафедра** інженерії програмного забезпечення

**Освітній ступінь** магістр

**Спеціальність** 121 Інженерія програмного забезпечення

**Освітньо-професійна програма** «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Катерина НЕСТЕРЕНКО

" \_\_\_ " \_\_\_\_\_ 2023 р

## ЗАВДАННЯ

на виконання кваліфікаційної роботи студентки  
Чеснокової Катерини Сергіївни

1. Тема кваліфікаційної роботи: «Методика та програмний засіб спостереження за економічною ефективністю компаній»  
затверджена наказом ректора від 29.09.2023 р. № 1994/ст.
2. Термін виконання проекту: з 02.10.2022 р. по 26.12.2023 р.
3. Вихідні дані до роботи : програмний продукт розробити на основі платформи .NET Framework як Windows Service застосунок.
4. Зміст пояснювальної записки:
  1. Аналіз предметної області спостереження за економічною ефективністю компаній;
  2. Аналіз та формування вимог до розроблюваної системи. архітектура системи створення звітів;
  3. Реалізація програмного засобу спостереження за економічною ефективністю компаній.
  4. Результати роботи системи
5. Перелік обов'язкових слайдів презентації:
  1. Актуальність теми;
  2. Мета дослідження;
  3. Задачі, предмет та об'єкт дослідження;
  4. Аналіз існуючих рішень;
  5. Запропоноване рішення;
  6. Результати роботи системи;
  7. Висновки.

## 6. Календарний план-графік

№ пор.	Завдання	Термін виконання	Відмітка про виконання
1.	Розробка та затвердження графіка роботи	25.10.23 – 31.10.23	
2.	Підготовка та написання 1 розділу. Відсилка керівнику	25.10.23 – 31.10.23	
3.	Підготовка та написання 2 розділу. Відсилка керівнику	23.10.23 – 12.11.23	
4.	Підготовка та написання 3 розділу. Відсилка керівнику	13.11.23 – 3.12.23	
5.	Підготовка та написання 4 розділу. Відсилка керівнику	4.12.23 – 10.12.23	
6.	Редагування та друк пояснювальної записки, графічного матеріалу Відсилка ПЗ для перевірки на плагіат одним файлом.	13.12.23 – 19.12.23	
7.	Проходження нормо-контролю, перепліт пояснювальної записки. Отримання відгуку керівника. Підготовка презентації та тексту доповіді.	20.12.23 – 26.12.23	
8.	Передзахист (наявність віддрукованої ПЗ, презентації, позитивного відгуку керівника). При наявності цього приймається рішення про допуск к захисту дипломного проекту перед Екзаменаційною комісією. Що оформлюється протоколом засідання кафедри	18.12.23 – 24.12.23	
9.	Підготовка матеріалів для передачі секретарю ДЕК (ПЗ, CD-R з електронними копіями ПЗ, презентації, відгук керівника, рецензія) в папці	21.12.23 – 22.12.23	
10.	Захист дипломної роботи перед ЕК	26.12.23	

Дата видачі завдання 03.10.2023 р.

Керівник дипломної роботи:

к.т.н доцент Олена КОЛГАНОВА

Завдання прийняла до виконання:

Катерина ЧЕСНОКОВА

## РЕФЕРАТ

Пояснювальна записка до дипломної роботи «Методика та програмний засіб спостереження за економічною ефективністю компаній»: 61 сторінка, 27 рисунків, 1 таблиця, 16 використаних джерел.

ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ КОМПАНІЇ, ФІНАНСОВІ ПОКАЗНИКИ, ЕКОНОМІЧНИЙ АНАЛІЗ КОМПАНІЇ, ІНВЕСТИЦІЇ, ЗВІТ

**Об'єкт дослідження** – система створення звітів призначена для користування інвесторами.

**Мета дипломної роботи** – спрощення процесу аналізу економічної ефективності компаній інвесторами шляхом збору та систематизації інформації через запровадження системи створення звітів.

**Метод дослідження** – аналіз можливостей та обмежень технологій та фреймворків для побудови архітектури системи створення звітів, розробка такої системи та її використання для отримання структурованих даних.

**Результати роботи** можуть бути використані при розробці архітектури та побудови системи створення звітів для програмних засобів, особливо тих, що оперують великою кількістю даних.

Розробка та дослідження проводилися під управлінням операційної системи Windows 10. Розробка програми проводилася на мові програмування C# платформи .NET Framework, а також на платформі Angular.

## **ABSTRACT**

Explanatory note to the thesis "Methodology and software for monitoring the economic efficiency of companies": 61 p., 27 fig., 1 table., 16 information sources.

**ECONOMIC EFFICIENCY OF THE COMPANY, FINANCIAL INDICATORS,  
ECONOMIC ANALYSIS OF THE COMPANY, INVESTMENTS, REPORT**

**Property of development** – the reporting system that is intended for use by investors.

**Purpose** – simplifying the process of companies' economic efficiency analysis conducted by investors by collecting and systematizing information through the introduction of a reporting system.

## ЗМІСТ

ПЕРЕЛІК ПРИЙНЯТИХ СКОРОЧЕНЬ.....	8
ВСТУП .....	9
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ СПОСТЕРЕЖЕННЯ ЗА ЕКОНОМІЧНОЮ ЕФЕКТИВНІСТЮ КОМПАНІЙ .....	12
4.1. Аналіз предметної області.....	12
4.2. Аналіз основних фінансових показників компанії.....	16
4.3. Актуальність розробки програмного засобу спостереження за економічною ефективністю компаній та аналіз існуючих аналогів.....	20
Висновки .....	25
РОЗДІЛ 2 АНАЛІЗ ТА ФОРМУВАННЯ ВИМОГ ДО РОЗРОБЛЮВАННОЇ СИСТЕМИ. АРХІТЕКТУРА СИСТЕМИ СТВОРЕННЯ ЗВІТІВ .....	26
2.1. Аналіз вимог до системи спостереження за економічною ефективністю компаній .....	26
2.2. Функціональні вимоги .....	27
2.3. Нефункціональні вимоги.....	28
2.4. Архітектура системи створення звітів .....	30
Висновок .....	40
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАСОБУ СПОСТЕРЕЖЕННЯ ЗА ЕКОНОМІЧНОЮ ЕФЕКТИВНІСТЮ КОМПАНІЙ .....	41
3.1. Архітектура системи.....	41
3.2. Налаштування SSRS, створення шаблонів звіту .....	42
3.3. Користувацький інтерфейс .....	46
3.4. Підсистема створення звітів.....	49

Висновки .....	55
РОЗДІЛ 4 РЕЗУЛЬТАТИ РОБОТИ СИСТЕМИ .....	56
4.1. Результати роботи системи .....	56
4.2. Напрямки розширення функціоналу .....	61
Висновки .....	61
ВИСНОВКИ.....	62
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	63

## **ПЕРЕЛІК ПРИЙНЯТИХ СКОРОЧЕНЬ**

RDL – Report Definition Language

SSRS – SQL Server Reporting Services



## ВСТУП

У сучасному світі, де економіка стає все більш взаємопов'язаною та конкурентною, розуміння та ефективне використання ресурсів на корпоративному рівні стає ключовим завданням для ефективної діяльності. Наразі бізнес-середовище вимагає від компаній не лише виживання, але й активного розвитку та досягнення високих результатів. Економічні результати діяльності компаній визначаються низкою економічних показників, які відображають ефективність їхньої діяльності та впливають на їхню конкурентоспроможність і підтримку: це, зокрема, ефективність використання ресурсів, прибутковість, фінансова стійкість, здатність адаптуватися та пристосовуватися до внутрішнього та зовнішнього середовищ.

Швидкі темпи глобалізації та постійні зміни у сфері технологій змушують компанії постійно вдосконалюватися та пристосовуватися до нових ситуацій. У цьому контексті аналіз економічної ефективності стає стратегічним інструментом для розуміння, оцінки та оптимізації ресурсів для досягнення максимальних результатів. Важливо зазначити, що ефективність не є статичним поняттям, і для забезпечення постійного прогресу важливий систематичний аналіз і створення бізнес-плану.

Аналіз економічних показників діяльності компанії включає комплексний аналіз фінансових показників, операційних показників, контролю процесів та управлінського контролю. Врахування цих аспектів дозволяє отримати детальний опис ефективності діяльності компанії, а також виявити сильні та слабкі сторони, що є основою для прийняття подальших стратегічних рішень. В сучасному суспільстві соціальні та екологічні аспекти діяльності компаній також стали важливими факторами. Ефективність

компаній повинна бути збалансованою, забезпечуючи не лише фінансовий успіх, але й враховуючи потреби людей та навколишнього середовища.

Оцінка фінансових показників є важливою не тільки для менеджменту компаній, але і для інвесторів. Вони загалом не схильні до ризикових вкладень, тому аналіз компанії є для них важливим етапом перед прийняттям рішень. Цей аналіз передбачає вивчення безлічі факторів, таких як коефіцієнти прибутковості, ліквідності, платоспроможності та показники ефективності. Кожен із цих показників відіграє особливу роль у створенні детальної картини того, наскільки добре компанія використовує свої ресурси, керує своїми боргами та отримує прибуток для своїх інвесторів. Звіти служать важливим джерелом інформації, пропонуючи інвесторам панорамне уявлення про економічний стан компанії.

Система створення звітів є цінним інструментом, який спрощує процес збору та представлення важливих даних, пов'язаних з економічною ефективністю компанії, надаючи інвесторам інструменти для навігації у складних ситуаціях фінансових ринків. Розглядаючи фінансову історію компанії, включаючи й минулі показники, інвестори можуть проводити аналіз тенденцій і робити більш обґрунтовані прогнози щодо її майбутньої прибутковості. Крім того, система звітності сприяє оцінці ризиків, надаючи детальний огляд фінансового стану компанії.

Здатність збирати, аналізувати та експортувати інформацію в зручних для користувача форматах документів є важливою для полегшення ефективного прийняття рішень, сприяння співпраці та покращення загального розуміння економічної ефективності компанії з точки зору інвестора.

У даній роботі буде проведено аналіз метрик спостереження за економічною ефективністю діяльності компаній та розглянута роль систем створення звітів у оцінці ефективності компаній. Також в ході роботи буде

розроблена модель програмного засобу, що може бути використана для сторення звітів з метою аналізу економічних показників компанії.

# РОЗДІЛ 1

## АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ СПОСТЕРЕЖЕННЯ ЗА ЕКОНОМІЧНОЮ ЕФЕКТИВНІСТЮ КОМПАНІЙ

### 4.1. Аналіз предметної області

Економічна ефективність компаній – це міра здатності підприємства до досягнення максимальних результатів при мінімальних витратах [1]. Вона включає в себе оцінку різних аспектів фінансової та господарської діяльності компанії, таких як рентабельність, ліквідність, фінансова стійкість, ринкова цінність, інноваційність та багато інших. Ефективна компанія має здатність адаптуватися до мінливих ринкових умов і досягати поставлених цілей.

Поняття ефективності компанії включає в себе різні характеристики, оскільки воно охоплює ядро організаційної структури, яка була налаштована для максимальної ефективності. Найважливішою з цих характеристик є операційна ефективність, яка є ознакою ретельного управління ресурсами та збільшення продуктивності. Справді ефективний бізнес ретельно керує динамікою свого ланцюжка поставок та мінімізує відходи в усіх аспектах виробництва.

Розсудливість у фінансах також є важливою якістю ефективною структури компанії. Окрім отримання прибутку бізнес має впроваджувати розумне управління фінансами, зберігаючи баланс між винагородою та ризиком. Таке дбайливе поводження з грошима захищає бізнес від ринкових коливань і полегшує здійснення розрахункових витрат у сферах, де можливе розширення.

Включення найсучасніших технологій також є важливою характеристикою, підприємства використовують найсучасніші технології для підвищення своїх операційних можливостей. Вони використовують аналітику даних та автоматизацію для прийняття обґрунтованих рішень і оптимізації роботи.

Однак ефективність виходить за межі внутрішніх процесів і включає зв'язки із зовнішніми зацікавленими сторонами, а оцінка економічної ефективності компанії передбачає комплексний аналіз. Ключові зацікавлені сторони, яким зазвичай потрібно оцінити економічну ефективність компанії такі:

- акціонери, тобто інвестори, що володіють акціями компанії, які безпосередньо впливають на її економічні результати. Їх цікавить здатність компанії генерувати прибуток на інвестиції, що можна побачити у таких фінансових показниках як рентабельність інвестицій, рентабельність капіталу, тощо;
- потенційні інвестори, тобто фізичні особи або компанії, які розглядають можливість інвестування в компанію. Вони переглядають економічні показники щоб бути впевненими, що їхній капітал буде добре використаний і принесе задовільний прибуток;
- керівництво, яке відповідає за реалізацію планів щодо покращення економічних показників компанії. Вони аналізують показники продуктивності, показники витрат і розподіл ресурсів, щоб забезпечити оптимальну продуктивність і прибутковість;
- кредитори безпосередньо зацікавлені в економічних показниках компанії, оскільки це впливає на її здатність погашати свої борги. Вони аналізують грошовий потік та загальний фінансовий стан, щоб визначити життєздатність компанії;
- державні та регулюючі органи можуть контролювати економічні показники компанії задля забезпечення дотримання стандартів, фінансових правил і практики чесної конкуренції;
- конкуренти уважно спостерігають за економічною діяльністю один одного, щоб визначити сфери для вдосконалення чи інновацій.

Отже, можна сказати, що оцінка економічної діяльності компанії – це багатогранний процес, в якому ефективність компанії може розглядатися з різних сторін. Надалі питання економічної ефективності компанії буде розглядатися саме з точки зору інвесторів.

Оскільки інвестори орієнтуються у складному ландшафті фінансових ринків, розуміння фінансових показників компаній стає важливим для прийняття правильних рішень. Розглянемо детальніше концепції економічної ефективності з точки зору інвестора, поглянемо на їх аспекти та наслідки для прийняття інвестиційних рішень.

Економічна ефективність означає здатність компанії максимізувати виробництво, мінімізуючи при цьому витрати і досягаючи таким чином найвищого рівня продуктивності. З точки зору інвестора ця ефективність відображається в здатності компанії отримувати стабільний прибуток і створювати акціонерну вартість. Економічна ефективність включає як розподільну, так і продуктивну ефективність. Ефективність розподілу забезпечує оптимальний розподіл ресурсів для задоволення споживчого попиту, тоді як ефективність виробництва зосереджується на мінімізації витрат у процесі виробництва [6].

Операційна ефективність, підмножина економічної ефективності, також відіграє важливу роль у залученні інвесторів. Ефективна діяльність веде до економії коштів, підвищення прибутку та підвищення конкурентоспроможності. Інвестори часто ретельно перевіряють ланцюг постачання компанії, виробничі процеси та загальну операційну структуру, щоб оцінити потенціал довгострокової прибутковості. Здатність компанії адаптуватися до технологічного прогресу та впроваджувати ефективні операційні методи значною мірою сприяє її економічній ефективності [7].

Інвестори загалом не схильні до ризику, а економічна ефективність компанії тісно пов'язана зі стратегіями управління ризиками. Ефективні заходи зі зниження ризиків сприяють стабільній фінансовій діяльності та

зменшують ймовірність негативних подій, які можуть вплинути на довіру інвесторів. Компанії з надійною системою управління ризиками сприймаються як більш економічно ефективні, оскільки вони демонструють відданість збереженню акціонерної вартості.

У сучасному фінансовому кліматі стійкість стала ключовим фактором для інвесторів. Компанії, які дотримуються екологічної та соціальної етики, все більше приваблюють інвесторів, які стурбовані довгостроковою стабільністю своїх інвестицій. У цьому випадку економічні показники виходять за рамки фінансової системи, включаючи екологічний і соціальний вплив компанії, демонструючи комплексний підхід до створення вартості.

Конкурентні ринкові умови також впливають на економічні показники компанії. Інвестори оцінюють позицію компанії в своїй галузі, її частку на ринку та її здатність адаптуватися до мінливих умов ринку. Компанії, які демонструють глибоке розуміння ринкових тенденцій і використовують можливості, вважаються надійними та більш привабливими для інвестицій.

Отже, з точки зору інвестора, оцінка фінансових показників компанії – це комплексний захід, який включає аналіз фінансових показників, управління ризиками, стійкості та динаміки ринку. Добре розуміння цих аспектів дозволяє інвесторам приймати зважені рішення та інвестувати в компанії, які демонструють високі фінансові показники.

Процес оцінки економічної ефективності компанії включає в себе ряд етапів:

- 1) визначення ключових показників діяльності, які відображають фінансовий стан компанії. Це можуть бути показники прибутковості, ліквідності, фінансової стійкості, тощо;
- 2) збір та аналіз фінансово-економічних даних компанії, таких як фінансова звітність, бюджети, виробничі звіти, витрати та інші;

- 3) порівняльний аналіз отриманих даних з аналогічними цифрами або показниками конкурентів, що дозволяє побачити як компанія працює порівняно з галузевими стандартами;
- 4) виявлення тенденцій та закономірностей в роботі бізнесу. Можливо також використання статистичних моделей та методів прогнозування для передбачення майбутніх фінансових показників компаній.

На основі аналізу зібраних даних інвестори формують список переваг і недоліків компанії. Розглядаючи різні фактори, вони приходять до висновку про рівень ризику інвестування в цю організацію. Беручи до уваги ці аспекти, інвестори приймають обґрунтоване рішення, чи є конкретна компанія вигідною для їхнього інвестиційного портфеля.

#### **4.2. Аналіз основних фінансових показників компанії**

Для оцінки економічної ефективності використовуються різні методи та показники, але важливо зазначити, що вони можуть відрізнятися залежно від регіону, розміру компанії та її стратегічних цілей.

Інвестори покладаються на різноманітні фінансові показники для вимірювання економічної ефективності компанії. Аналіз акцій передбачає комплексну оцінку різних факторів для прийняття обґрунтованих інвестиційних рішень. Ось кілька ключових пунктів, на які звертають увагу інвестори:

- фінансова звітність, що включає звіт про прибутки та збитки (аналіз доходів, витрат і прибутковості), бухгалтерський баланс (оцінка активів, зобов'язань і власного капіталу), звіт про рух грошових коштів (оцінка здатності компанії генерувати готівку та управляти нею);



- прибуток і дивіденди, а саме історичне та прогнозоване зростання прибутків компанії, історія дивідендів та коефіцієнт виплат, якщо отримання доходу є пріоритетом;
- галузь і ринкові умови, адже важливо розуміти галузеві тенденції та динаміку, що впливають на компанію, й мати можливість оцінити позицію компанії відносно конкурентів;
- конкурентне позиціонування, тобто частка ринку та конкурентні переваги компанії;
- ризики, наприклад ризики для галузі, геополітичні, регуляторні та економічні ризики;
- потенціал компанії для майбутнього зростання;
- різні фінансові коефіцієнти, такі як коефіцієнт ціна-прибуток, коефіцієнт ціна-балансова вартість та інші;
- кредитні плечі та рівень боргу компанії. Варто розглядати співвідношення боргу до власного капіталу та коефіцієнт покриття відсотків;
- макроекономічні фактори, такі як процентні ставки, інфляція тощо;
- підхід компанії до технологій та інновацій, її здатність адаптуватися до змін ринкових тенденцій;
- прихильність компанії до соціальної та екологічної відповідальності. Останнім часом потреба компаній у дотриманні соціальних та екологічних стандартів значно зросла, оскільки це може вплинути на імідж бренду в довгостроковій перспективі;
- рейтинги та рекомендації аналітиків.

Розглянемо більш детально фінансові коефіцієнти. Фінансові коефіцієнти — це інструменти, за допомогою яких інвестори оцінюють привабливість акцій компанії шляхом порівняння її поточної ринкової

вартості з різними фінансовими показниками. Ці показники допомагають інвесторам визначити, чи акція переоцінена, недооцінена або оцінена справедливо.

До таких фінансових метрик належать [8]:

- співвідношення ціна-прибуток (P/E) – це відношення ринкової ціни на акцію до прибутку на акцію. Високий коефіцієнт P/E може вказувати на те, що інвестори мають високі очікування щодо майбутнього зростання доходів, тоді як низький коефіцієнт P/E може свідчити про недооцінку;
- співвідношення ціна-балансова вартість (P/B) – це відношення ринкової ціни за акцію до балансової вартості за акцію, тобто до вартості чистих активів компанії. Мультиплікатор P/B може вказувати на коректність поточної вартості акцій, у фінансовій спільноті прийнято трактувати показники таким чином: значення P/B більше ніж 2 з великою ймовірністю свідчить, що це підприємство переоцінено, якщо P/B від 1 до 2 – вартість акцій справедливо оцінена, P/B менше ніж 1 – підприємство недооцінене;
- дивідендна дохідність – це відношення річного дивіденду на акцію до ринкової ціни на акцію. Він представляє річний дивідендний дохід у відсотках від поточної ринкової ціни. Вища дивідендна дохідність може вказувати на кращу можливість отримання прибутку для інвестора;
- коефіцієнт виплати дивідендів, що дорівнює відношенню дивіденду на акцію до прибутку на акцію. Вказує частку прибутку, що виплачується як дивіденди. Нижчий коефіцієнт може означати, що компанія зберігає більше прибутку для зростання;

- прибуток, тобто відношення прибутку на акцію до ринкової ціни на акцію. Обернений до співвідношення P/E, він представляє прибуток, отриманий у відсотках від ринкової ціни. Це допомагає оцінити відносну привабливість заробітку порівняно з ринковою ціною;
- рентабельність власного капіталу (ROE) – це відношення чистого прибутку до акціонерного капіталу, показує здатність компанії отримувати прибуток від акціонерного капіталу. Вище значення ROE, як правило, більш бажане;
- співвідношення боргу до власного капіталу, де нижчий коефіцієнт вказує на нижчий фінансовий леверидж;
- вільний грошовий потік представляє собою готівку, згенеровану компанією, яка доступна для розподілу між інвесторами, погашення боргу або реінвестування;
- співвідношення ціни до прибутку до зростання (PEG) – вираховується як відношення P/E до річного темпу зростання прибутку. Коефіцієнт PEG нижче 1 може вказувати на недооцінку;
- зростання продажів вимірює відсоткове збільшення доходу компанії за певний період. Постійне та стійке зростання продажів часто вважається позитивним.

Важливо зазначити, що жоден показник чи фактор інвесторам не слід розглядати окремо. Вони повинні використовувати комбінацію цих факторів, щоб отримати повне розуміння фінансового стану компанії, потенціалу зростання та загальної інвестиційної привабливості.

Тож можна зробити висновок, що аналіз економічної ефективності компаній вимагає комплексного підходу, врахування різних факторів та використання різноманітних методів та інструментів. Диверсифікація та

періодична переоцінка інвестицій також є важливими компонентами надійної інвестиційної стратегії.

Враховуючи обсяг і складність інформації, яка використовується для аналізу, автоматизація процесу збору даних є важливою. Це звільняє час і ресурси, які можна використовувати для аналізу результатів, а застосування спеціальних програм і інструментів дозволяє правильно організувати обсяг даних, забезпечуючи точність і швидкість аналізу.

#### **4.3. Актуальність розробки програмного засобу спостереження за економічною ефективністю компаній та аналіз існуючих аналогів**

Для здійснення спостереження за економічною ефективністю компаній можуть використовуватися спеціалізовані програмні засоби, які дозволяють автоматизувати збір і аналіз фінансових даних, створювати звіти та аналітичні інструменти для прийняття рішень. Деякі з них можуть використовувати штучний інтелект для прогнозування та моделювання різних сценаріїв.

Основною метою автоматизації спостереження за економічною ефективністю компаній є забезпечення швидкого та точного доступу до фінансової інформації, а також підвищення рівня аналізу та прогнозування. Конкретні цілі автоматизації можуть включати зменшення людських помилок у процесі збору та обробки фінансових даних, а також збільшення швидкості створення фінансових звітів і аналітичних інструментів. Метою є й покращення моніторингу фінансових показників та реакція на негативні тенденції в реальному часі, а також покращення прийняття рішень та управління процесами на основі точного аналізу.

Автоматизація спостереження за економічною ефективністю компаній надає безліч переваг, наприклад:

- підвищення продуктивності, адже автоматизовані системи швидше збирають, обробляють та аналізують фінансову

інформацію, що дозволяє працівникам більше часу приділяти стратегічному аналізу та прийняттю рішень;

- мінімізація ризику людських помилок завдяки автоматичній обробці даних;
- автоматизовані системи забезпечують миттєвий доступ до оновленої інформації, дозволяючи оперативно реагувати на зміни в умовах ринку;
- зменшення витрат на обробку та аналіз даних завдяки оптимізації ресурсів.

На ринку вже існують різні програмні засоби для спостереження за економічною ефективністю компаній. Розглянемо декілька програмних засобів, які відзначаються на ринку, їх основні функції та недоліки:

1) FINVIZ – це комплексний фінансовий веб-сайт, що пропонує інструменти для інвесторів і трейдерів. Безкоштовна версія платформи надає доступ до потужного скринінгу акцій, інтерактивних діаграм з інструментами технічного аналізу та теплових карт для швидкого огляду секторів і галузей. Крім того, FINVIZ збирає фінансові новини та аналіз. Для користувачів, яким потрібні розширені функції та дані в реальному часі, є версія Elite, доступна за підпискою.

Веб-сайт має інтуїтивно зрозумілий дизайн, що дозволяє користувачам налаштовувати свою інформаційну панель, зберігати налаштування та експортувати діаграми. Хоча базові функції експорту доступні для усіх користувачів, передплатники версії Elite можуть користуватися додатковими параметрами експорту та розширеними функціями.

Сайт надає користувачам також простір для обговорення тенденцій ринку та конкретних акцій. До недоліків FINVIZ можна віднести те, що деякі дані у реальному часі можуть бути обмежені для користувачів без підписки. Вони також можуть зіткнутися із затримкою даних, що може вплинути на точність і своєчасність їхніх торгових рішень.



Рис. 1.1. Користувацький інтерфейс веб-сайту «FINVIZ»

2) Seeking Alpha – це веб-сайт фінансового аналізу, який пропонує перегляд і аналіз акцій, облігацій та інших інвестицій. Він містить статті та думки окремих інвесторів, а також професійних аналітиків. Seeking Alpha пропонує як безкоштовний, так і преміальний контент. Основна інформація та деякі статті доступні безкоштовно, а для преміум-контенту може знадобитися підписка. Функція експорту даних обмежена для користувачів без підписки, але платні підписники можуть мати додаткові параметри для експорту вмісту чи даних. Загалом Seeking Alpha надає інвесторам платформу для обміну та обговорення інвестиційних ідей. Він містить аналіз, новини та думки щодо широкого спектру фінансових інструментів.

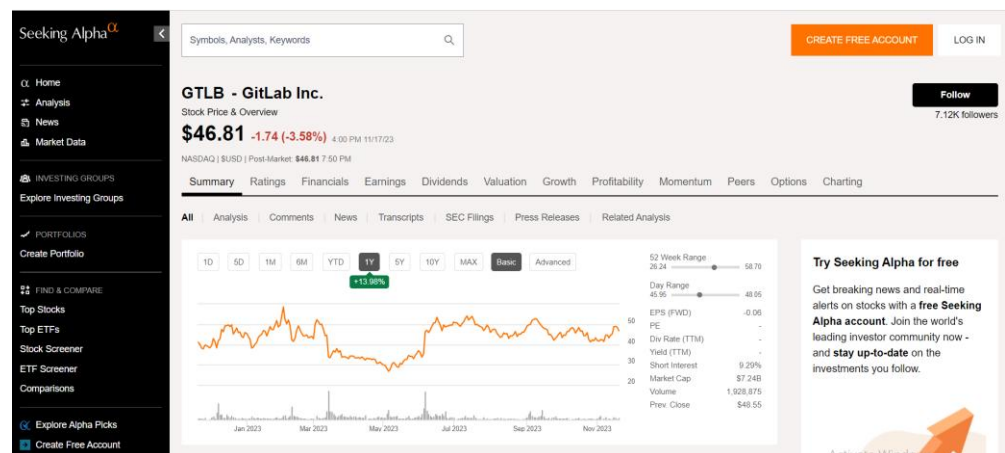


Рис. 1.2. Користувацький інтерфейс веб-сайту «Seeking Alpha»

Оскільки Seeking Alpha покладається на внески широкого кола людей, якість і достовірність вмісту можуть відрізнятись. Користувачам слід бути обережними та виконувати додаткові дослідження, покладаючись на інформацію, надану учасниками. Велика кількість статей і думок на сайті може бути надлишковою для деяких користувачів, особливо тих, хто тільки починає інвестувати.

3) Trading View — це веб-платформа, яка надає графіки та соціальні мережі для активних трейдерів та інвесторів. Вона охоплює різні фінансові ринки, включаючи акції, форекс, криптовалюти тощо. TradingView пропонує як безкоштовні, так і платні аккаунти. Основні інструменти для створення діаграм і аналізу доступні безкоштовно, тоді як для більш розширених функцій і даних у реальному часі може знадобитися підписка. Основні функції платформи включають інтерактивні діаграми, інструменти технічного аналізу, соціальні мережі для трейдерів і можливість ділитися торговими ідеями та стратегіями. TradingView дозволяє користувачам експортувати діаграми та аналіз. Параметри експорту можуть включати збереження діаграм як зображень або експорт даних у різних форматах. Деякі розширені функції експорту можуть бути доступні платним підписникам.

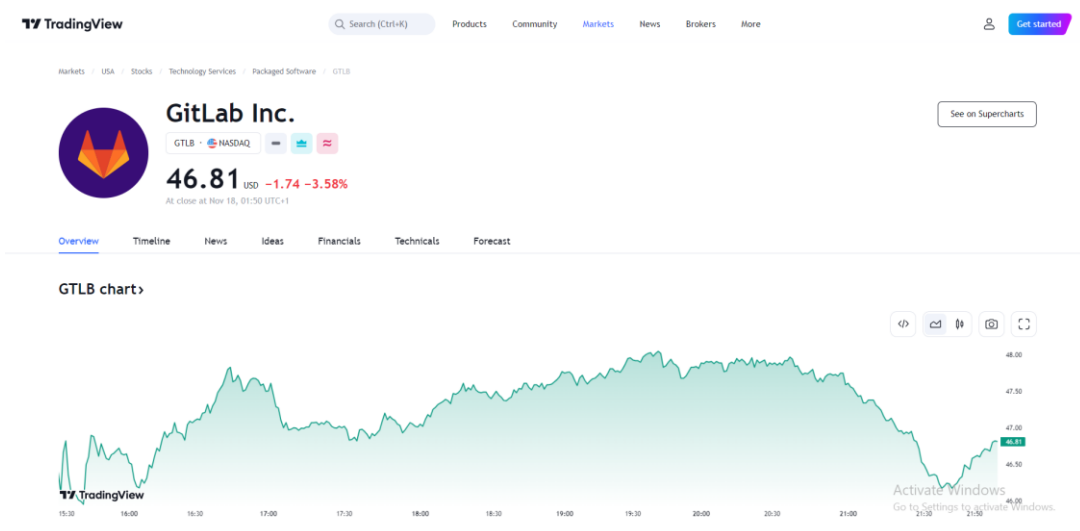


Рис. 1.3. Користувацький інтерфейс веб-сайту «TradingView»

TradingView в першу чергу зосереджений на технічному аналізі, і він може не надавати стільки широких інструментів фундаментального аналізу, скільки платформи, спеціально розроблені для фундаментального аналізу. Також, незважаючи на те, що TradingView охоплює широкий спектр ринків, деякі користувачі можуть виявити, що певні ринки або певні фінансові інструменти мають обмежену глибину або покриття порівняно з більш спеціалізованими платформами. Разом з тим користувачі без платного акаунту можуть зіткнутися із затримкою даних.

Можна зробити висновок, що сучасні інструменти фінансового моніторингу дозволяють інвесторам контролювати фінансову діяльність компаній та надають непоганий функціонал навіть без платної підписки. Однак у той час як ці програми надають користувачам інструменти для аналізу в реальному часі та визначення тенденцій, можливість вилучення та експорту повних даних компанії у документи різних форматів залишається недостатньо розробленою.

Переваги функції експорту звітів [9]:

- експорт даних компанії у документ дозволяє інвесторам виконувати більш глибокий аналіз даних, що дозволяє створити комплексний звіт, який сприяє глибшому розумінню економічної ситуації;
- забезпечує створення добре впорядкованої та архівної документації. Інвестори можуть вести структурований облік фінансового аналізу, уможливаючи ретроспективний аналіз, порівняльні дослідження та довгострокове стратегічне планування;
- експортні звіти є ефективним інструментом спілкування для інвесторів, які беруть участь у спільному прийнятті рішень. Можливість обмінюватися повними документами між колегами



або фінансовими консультантами покращує спілкування та колективне розуміння;

- можливості звітування підтримують дотримання нормативних вимог і стандартів аудиту. Інвестори можуть збирати та представляти фінансову інформацію у форматі структурованого документа для систематичного аналізу.

Підсумовуючи, інтеграція функції експорту звітів у інструменти фінансового моніторингу є важливим удосконаленням, яке може значно підвищити їх корисність для інвесторів. Тому варто приділяти увагу впровадженню надійної функції експорту звітів, щоб надати інвесторам більше інструментів для прийняття обґрунтованих та ефективних рішень.

### **Висновки**

Оцінка економічної ефективності компанії необхідна багатьом учасникам ринку, в тому числі інвесторам. Для прийняття рішення про купівлю акцій вони покладаються на різноманітні фінансові показники, при тому жоден показник чи фактор інвесторам не слід розглядати окремо. Це призводить до необхідності аналізу великої кількості інформації, який можна спростити за допомогою експортування звітів. Зважаючи на те, що програмних засобів для спостереження за економічною ефективністю компаній, що включають в себе функцію створення звітів, небагато, можна зробити висновок, що розробка такої функції є актуальною.

## РОЗДІЛ 2

### АНАЛІЗ ТА ФОРМУВАННЯ ВИМОГ ДО РОЗРОБЛЮВАННОЇ СИСТЕМИ. АРХІТЕКТУРА СИСТЕМИ СТВОРЕННЯ ЗВІТІВ

#### **2.1. Аналіз вимог до системи спостереження за економічною ефективністю компаній**

У цьому розділі детально розглядаються аспекти, пов'язані з визначенням та формулюванням вимог до системи створення звітів. Аналіз цих вимог є важливим кроком у розробці ефективного інструменту звітності, оскільки вона визначає якість і точність отриманих результатів.

Повний список вимог до системи звітності залежить від різних факторів та може змінюватись в залежності від потреб користувачів конкретного програмного продукту. Також система звітності може не бути закладена в архітектуру застосунку на етапі проектування, а додана на більш пізніх етапах розвитку. В такому випадку на вимоги також можуть впливати архітектура застосунку, технології та фреймворки, що вже в ньому використовуються.

Загалом користувачі, що бажають створити звіт, повинні мати змогу обирати об'єм даних, що увійдуть у звіт, на користувацькому інтерфейсі. Важливо також надати користувачам можливість обрати формат звіту та забезпечити можливість показувати або приховувати нецікаву для користувача інформацію у звіті. Створення звітів має бути можливим як з персональних комп'ютерів та ноутбуків, так і з мобільних пристроїв. Користувацький інтерфейс повинен бути зручним у використанні незалежно від розміру екрану.

Система створення звітів може бути реалізована за допомогою різних технологій та фреймворків, в даній роботі надалі будуть детально

розглядатися два підходи: звіти на основі Angular та Microsoft SQL Server Reporting Services (SSRS).

## **2.2. Функціональні вимоги**

Функціональні вимоги відносяться до конкретних функцій і можливостей, якими повинна володіти система програмного забезпечення, щоб задовольняти потреби своїх користувачів і досягати поставлених цілей. Функціональні вимоги мають вирішальне значення для керування процесом розробки, гарантуючи, що програмне забезпечення відповідає потребам своїх користувачів та загальним цілям проекту. Ці вимоги є основою для розробки, впровадження та тестування програмних систем.

Як було зазначено вище, вимоги до системи звітності можуть різнитися в залежності від очікувань користувачів та обмежень застосунку, однак основні вимоги до систем створення звітів такі:

- автентифікація та авторизація користувача;
- можливість підключення до різних джерел даних, наприклад бази даних, прикладного програмного інтерфейсу (API), тощо;
- інтуїтивно зрозумілий і зручний інтерфейс;
- багатий набір можливостей візуалізації (схеми, графіки, карти);
- експорт звітів у різні формати, зокрема PDF, Excel, Microsoft Word, Microsoft PowerPoint;
- ефективна обробка великих наборів даних та масштабованість для обробки зростаючих обсягів даних;
- формування журналу системних подій.

### 2.3. Нефункціональні вимоги

Нефункціональні вимоги в розробці програмного забезпечення стосуються характеристик програмної системи, які не пов'язані з певною функціональністю чи поведінкою. Це набір специфікацій, які описують робочі можливості та обмеження системи.

Нефункціональні вимоги поділяють на такі типи:

- операційні вимоги – це безпека, надійність, продуктивність, тощо;
- вимоги до інтерфейсу – включають в себе вимоги до апаратних інтерфейсів, інтерфейси взаємодії з іншими системами та/або пристроями та користувачем;
- апаратні та програмні вимоги – апаратні та програмні платформи, необхідні для роботи застосунку.

Нефункціональні вимоги до системи створення звітів наведені у таблиці 2.1.

Таблиця 2.1

Тип вимоги	Вимога	Характеристика
Операційні вимоги	Безпека	Система повинна забезпечувати безпеку даних, доступ до даних можливий лише після авторизації у системі
	Керування помилками	Усі помилки повинні бути записані, наприклад до .log файлів
	Правила перевірки	Усі введені користувачем дані необхідно перевіряти на відповідність до очікуваного формату

Вимоги до інтерфейсу	Вимоги до інтерфейсу користувача	Користувацький інтерфейс має бути зручним та інтуїтивно зрозумілим, приємним для ока та без нагромодження елементів
Апаратні та програмні вимоги для створення звітів на основі Angular	Апаратні вимоги до сервера	Сервер для 100 і більше підключень. Підключення до Інтернету
	Програмні вимоги до сервера	Операційна система: Microsoft Windows 7 / Microsoft Windows 8 / Microsoft Windows 10/ Linux (64 біт / 86 біт)
Апаратні та програмні вимоги для створення звітів на основі SSRS	Апаратні вимоги до сервера	Мінімум 6 ГБ вільного місця на жорсткому диску, доступ до Інтернету, принаймні 4 ГБ вільної пам'яті і її слід збільшувати в міру збільшення розміру бази даних. Частота процесора 2,0 ГГц або вище, тип процесора x64: AMD Opteron, AMD Athlon 64, Intel Xeon з підтримкою Intel EM64T, Intel Pentium IV з підтримкою EM64T
	Програмні вимоги до сервера	Операційна система: Windows 10 TH1 1507 або новіша/Windows Server 2016 або новіша. Мінімальна операційна система включає мінімальну версію .NET Framework.
Апаратні та програмні вимоги до клієнта		Встановлений браузер, доступ до Інтернету

## 2.4. Архітектура системи створення звітів

Архітектура системи створення звітів може дуже різнитися від застосунку до застосунку. Вона визначає не тільки її технічні характеристики, функціональність, надійність і гнучкість, але й здатність задовольняти конкретні потреби користувачів і вимоги бізнес-середовища. В цьому підрозділі будуть розглянені фактори, що впливають на вибір архітектури, а також деякі варіанти архітектурних рішень.

Отже, структура систем звітності може значно відрізнятися в різних застосунках через вимоги, специфікації та контексти, в яких вони функціонують. Нижче наведено фактори, які можуть впливати на варіації архітектури системи звітності.

При проектуванні спершу важливо зрозуміти чи буде система створення звітів функціонувати як самостійний інструмент, чи як складова більшої системи. Якщо система звітності розробляється як окремий продукт, то така система може використовувати власні механізми зберігання даних, генерації звітів та інші компоненти. В такому випадку майже немає обмеження при виборі технологій та фреймворків, адже система незалежна.

Якщо ж система звітування є складовою більшої системи, особливо коли необхідно інтегрувати звітні можливості безпосередньо в існуючі бізнес-системи, виникають деякі обмеження при проектуванні. Сюди можна віднести, наприклад, обмеження на використання конкретної бази даних, мови програмування, розробку під конкретну операційну систему, тощо.

На вибір того, як система буде реалізована, також впливає й те, які запити на генерацію звітів вона буде приймати: внутрішні або зовнішні. Внутрішні запити – це ті, що приходять від самої системи. В такому випадку можна назвати систему закритою, адже усі процеси, такі як створення запиту на генерацію, генерація, відображення та інші маніпуляції зі звітом, виконуються в одному сервісі. Така система може бути вбудована,

наприклад, у веб-застосунок і не мати відкритого інтерфейсу для взаємодії з іншими компонентами.

На противагу цьому підходу можна створити систему, що буде приймати запити на генерацію звіту від різних сервісів або навіть від різних застосунків. Така система може бути реалізована різноманітними способами, розглянемо деякі з них:

- Windows Service для .NET Framework або Worker Service для .NET. Різниця цих двох типів застосунків у підтримці різних операційних систем: Windows Service використовується виключно на операційних системах Microsoft Windows, а Worker Service можна налаштовувати для роботи з Windows, Linux та macOS. Загалом обидва сервіси призначені для безперервної роботи у фоновому режимі. Після встановлення сервіси можна налаштувати на автоматичний запуск разом із операційною системою. Вони можуть працювати необмежений час, періодично виконуючи завдання, реагувати на системні події або виконувати інші операції без втручання користувача;
- Web API, що зазвичай працює у відповідь на вхідні запити. Сам Web API може постійно працювати в серверному середовищі, але він не працює так само, як служба Windows або Worker: він не має статусу та працює на основі кожного запиту.

Також якщо система створення звітів має приймати запити з різних сервісів або застосунків, використання брокерів повідомлень, таких як RabbitMQ або Amazon Simple Queue System (SQS), може стати у нагоді. Брокери повідомлень часто використовуються в розподілених системах для забезпечення зв'язку між роз'єднаними компонентами. Причини для їх використання включають [12]:

- спрощення асинхронного зв'язку між різними компонентами системи. Замість того, щоб компоненти безпосередньо викликали

функції або API один одного, вони надсилають повідомлення до спільної черги;

- розділення (від англ. «decoupling») виробників та споживачів повідомлень. Це значить, що системи не мають знати про існування одна одної для організації спільної роботи;
- забезпечення надійності та стійкості системи. Якщо система створення звітів тимчасово недоступна, повідомлення можна зберігати в черзі, доки не з'явиться можливість їх обробити. Це гарантує, що повідомлення не будуть втрачені та що система зможе впоратися з тимчасовими стрибками навантаження;
- брокери повідомлень підтримують масштабування, дозволяючи кільком екземплярам системи, що працюють паралельно, виконувати обробку повідомлень одночасно. Це дозволяє легко масштабувати різні частини системи незалежно одна від одної;
- брокери повідомлень можуть розподіляти навантаження між кількома екземплярами системи, забезпечуючи ефективну обробку кожного повідомлення. Таке балансування навантаження допомагає уникнути перевантажень і покращує загальну продуктивність;
- покращення відмовостійкості, що забезпечується завдяки режимам підтвердження та повторної спроби. Систему, що надіслала запит, можна сповістити про успішну обробку повідомлення, а в разі невдачі надіслати повідомлення на обробку повторно;
- полегшення взаємодії між різними технологіями та платформами. Компоненти, реалізовані різними мовами або запущені на різних платформах, можуть спілкуватися через загальний протокол обміну повідомленнями;



- брокер повідомлень гарантує, що повідомлення обробляються в порядку надходження, що може бути важливо у деяких випадках.

Загалом обмін повідомленнями в чергах забезпечує надійний і масштабований спосіб обміну даними між різними частинами розподіленої системи. Вони необхідні для створення масштабованих, слабозв'язаних архітектур, особливо в ситуаціях, коли компоненти повинні взаємодіяти без прямих залежностей.

Також для системи створення звітів є важливим вибір технології для створення шаблонів звітів. Шаблоном в даному випадку є попередньо визначена структура або макет, який визначає спосіб представлення даних у звіті. Він служить набором інструкцій для системи звітності про те, як упорядковувати та формувати інформацію під час створення документів. Шаблони звітів відіграють важливу роль у стандартизації форми та структури звітів, забезпечуючи узгодженість між різними звітами, створеними системою. Для створення шаблону можна використовувати широкий спектр мов та технологій, наприклад Report Definition Language (RDL), JasperReports, HTML, Angular, тощо. При виборі важливо звернути увагу на переваги та недоліки кожної технології. Розглянемо їх на прикладі RDL та Angular.

RDL – це мова на основі XML, яка використовується службами звітування SSRS для визначення звітів. Особливості її використання такі [13]:

- звіти RDL тісно інтегровані з SQL Server, що дозволяє легко створювати звіти з баз даних SQL Server, однак це також означає обмеження їх використання з іншими базами даних. Одночасно з цим звіти RDL можуть підключатися до різних джерел даних, включаючи XML, OLE DB та інші;
- RDL підтримує використання параметрів, значення яких користувачі можуть вводити під час виконання, що робить звіти більш динамічними та гнучкими;

- для створення RDL можна використовувати спеціалізовані програми або розширення у Visual Studio, що візуалізують шаблон і тим самим дозволяють легко вносити зміни;
- RDL підтримує різноманітні елементи візуалізації даних, такі як таблиці, діаграми та графіки, хоча можливості дизайну можуть бути не такими розширеними, як деякі інші інструменти звітності;
- SSRS дозволяє планувати виконання та доставку звітів, гарантуючи, що користувачі отримають оновлені звіти через визначені проміжки часу;
- створення та проектування звітів RDL може вимагати навчання, особливо для розробників, які не знайомі з XML або середовищем SSRS;
- RDL не дозволяє перевикористання елементів шаблону одного звіту в іншому. Це призводить до того, що розробники витрачають більше часу на створення та підтримку звітів. При виникненні необхідності внесення змін у повторюваний компонент необхідно буде вручну оновлювати кожен визначений звіт, що може бути непростим та трудомістким завданням;
- розширене налаштування звітів RDL може вимагати глибокого розуміння архітектури SSRS і структури XML;
- оновлення або перехід до новішої версії SSRS може вимагати коригування наявних звітів RDL, особливо коли вводяться нові функції.

Таким чином, звіти RDL є потужним та інтегрованим рішенням для середовищ SQL Server, але їх складність і залежність від конкретних технологій можуть створити проблеми у деяких обставинах.

При виборі фреймворку Angular для створення шаблонів звітів слід звернути увагу на такі його особливості [14]:

- Angular дотримується архітектури на основі компонентів, що дозволяє перевикористовувати компоненти у різних звітах. Це створює більш зручну структуру звітів, що може значно полегшити їх підтримку;
- Angular надає потужну систему ін'єкцій залежностей, що може покращити тестування та зручність обслуговування коду;
- Angular вирішує багато проблем із сумісністю між браузерами, забезпечуючи узгоджену роботу в різних браузерах;
- Angular має багату екосистему сторонніх бібліотек та інструментів. Це означає, що розробникам доступний широкий спектр ресурсів і розширень для покращення звіту;
- на відміну від RDL звітів, що інтегровані з SSRS, для Angular звітів необхідно створити свою реалізацію служб, які будуть завантажувати дані до звіту та генерувати його;
- створення простих для розробки та підтримки звітів на Angular передбачає деяку початкову конфігурацію та налаштування проекту, що вимагає глибоких знань фреймворку від команди розробки;
- час від часу Angular випускає оновлення, які можуть містити критичні зміни та вимагатимуть зусиль для оновлення існуючих проектів.

Вирішуючи, чи використовувати Angular для шаблонів звітів, важливо враховувати конкретні вимоги системи звітності, набір навичок команди розробників і загальні цілі проекту. Для більших і складніших програм звітності з фокусом на інтерактивності Angular може стати хорошим вибором. Для простіших проектів може бути більш придатним легший фреймворк або бібліотека.

Також при проектуванні системи звітності слід звертати увагу на вихідні формати звітів. Найбільш популярні формати включають PDF,

Microsoft Excel, Microsoft Word, Microsoft PowerPoint. Однак список форматів, звичайно, залежить від потреб користувачів, тому відрізнятиметься від системи до системи. Деякі технології, наприклад SSRS, мають вбудовані розширення для експорту звітів у різні формати. Якщо вбудованих розширень технологія не має, то можна використовувати сторонні розширення.

Не останню роль при проектуванні також грає вартість системи. Сюди можуть входити витрати на ліцензії, розширення, бази даних, сервери, тощо. SSRS, наприклад, не потребує додаткової ліцензії, адже він є частиною SQL Server пакету. При використанні Angular додатково необхідно буде витратити кошти на ліцензії сторонніх бібліотек для перетворення документів з PDF до інших форматів.

Отже, можна зробити висновок, що в сучасному бізнес-середовищі різноманітність і складність вимог до системи звітності визначається не тільки різноманітністю бізнес-сфер, але й унікальністю кожної організації. Фактори, що визначають архітектуру таких систем, настільки різноманітні, що відмінності можна побачити навіть між схожими програмами.

Розглянемо детальніше приклади архітектур на основі SSRS та Angular. SSRS – це серверна платформа звітності, розроблена Microsoft. Вона є частиною набору інструментів Microsoft SQL Server і надає повний набір функцій для створення, керування та розгортання звітів. SSRS спочатку було опубліковано як надбудову з SQL Server 2000, а потім Microsoft додала та продовжує додавати нові функції до всіх версій платформи. Остання версія на даний момент – SQL Server 2022 Reporting Services.

Для створення SSRS звітів необхідні такі компоненти [15]:

- сервер звітів;
- сервер звітів і тимчасова база даних сервера звітів;
- Report Builder.

Розглянемо їх детальніше. Сервер звітів є основним механізмом SSRS, оскільки всі процеси, завдання та запити обробляються саме ним. Сервер звітів встановлюється як Windows Service, тобто фоновий сервіс.

SSRS базується на трирівневій архітектурі (рис. 2.1), що включає: середній рівень, рівень даних і рівень презентації. Середній рівень включає компоненти служби Windows і розширення. SSRS підтримує наступні типи розширень: розширення безпеки, розширення обробки даних, розширення обробки звітів, розширення візуалізації та розширення доставки. Однак розширень безпеки, обробки даних і рендерингу достатньо для запуску сервісу. На рівні даних відбувається взаємодія бази даних і джерел даних сервера звітів. Клієнтські програми та вбудовані або спеціальні інструменти розміщуються на презентаційному рівні. Цей рівень включає веб-сервіс, веб-портал, менеджер конфігурації служби звітування, дизайнер звітів.

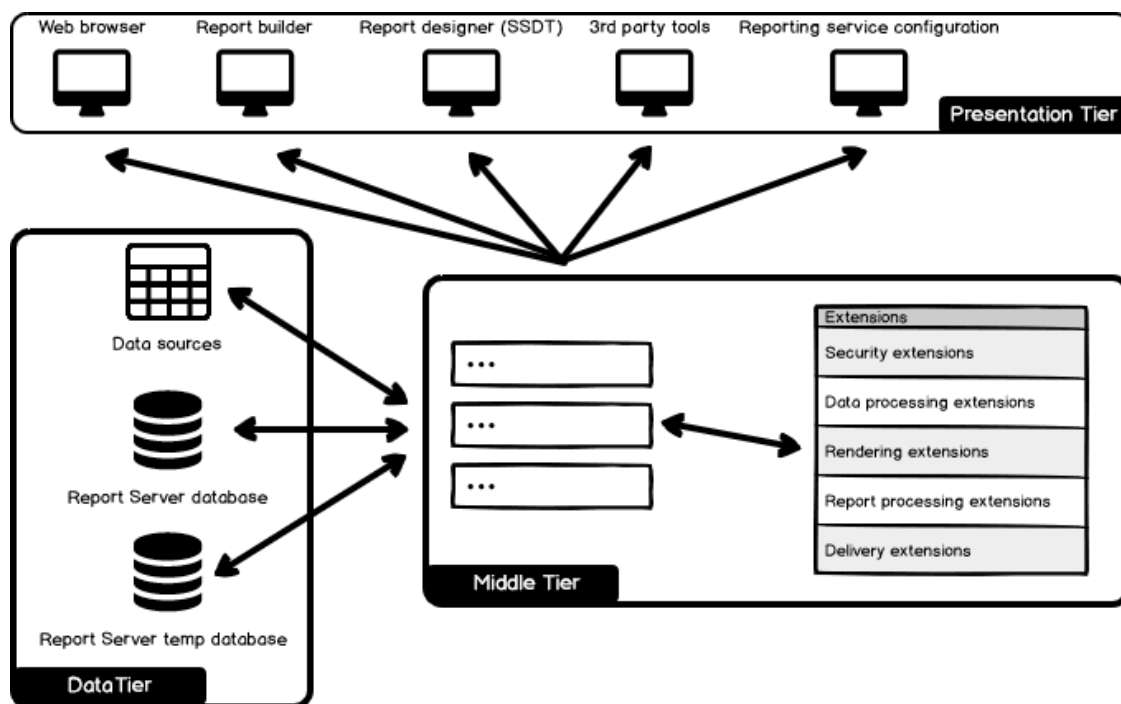


Рис. 2.1. Базова архітектура SSRS

Для кожного розгортання SSRS потрібні дві бази даних: ReportServer та ReportServerTempDB. Вони зберігають інформацію про користувачів,

визначення звітів і метадані звітів, журнали виконання та інше. Report Builder, в свою чергу, – це легкий інструмент для розробки звітів. Потік обробки звітів зображений на рис. 2.2.

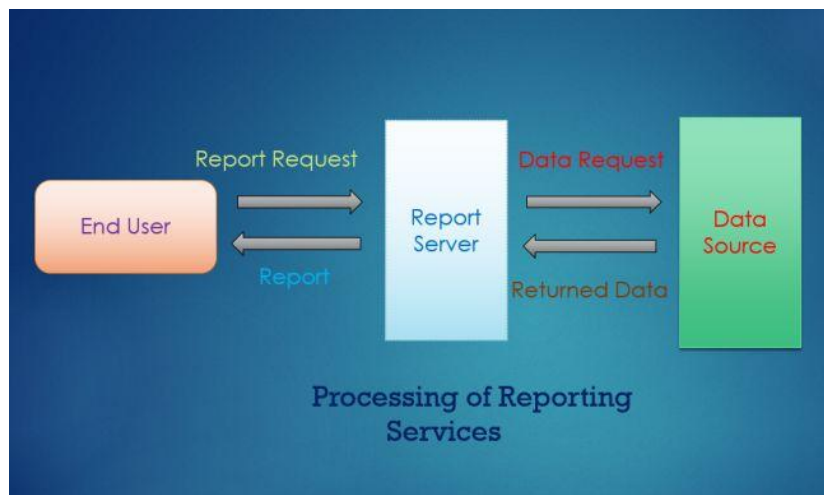


Рис. 2.2. Потік обробки звітів у SSRS

Angular, на відміну від SSRS, не має вбудованих функцій для створення звітів, а може слугувати лише для створення компонентних шаблонів. В такому випадку з'являється необхідність створення власних сервісів, що будуть відповідати за підготовку даних, складання звіту та його експорт у бажані формати.

Розглянемо приклад архітектури такої системи. У даному випадку розглянемо варіант, коли необхідно вбудувати систему звітності в уже існуючу більшу систему, що має доступ до даних. В даному випадку основною системою буде веб-додаток, а сервіс створення звітів буде встановлений на сервер як фоновий процес Worker Service. При цьому важливо зазначити, що Angular проект має бути також встановлений та запущений на сервері. В такому випадку можна побудувати наступний потік створення звітів (рис. 2.3):

- 1) веб-додаток відправляє запит на створення звіту до брокера повідомлень;

- 2) генератор отримує повідомлення та починає процес створення звіту;
- 3) генератор створює екземпляр браузера, наприклад CefSharp, що відкриває сторінку Angular застосунку за посиланням, переданим у запиті;
- 4) при відкритті сторінки Angular застосунку збирається звіт, на даному етапі він ще є веб сторінкою. Дані до звіту завантажуються через Web API веб-додатку;
- 5) браузер експортує сторінку у документ PDF формату;
- 6) при необхідності використовуються додаткові бібліотеки для переведення документу в інші формати;
- 7) генератор звітів зберігає створений документ у вказаному сховищі, такому, наприклад, як папка на сервері чи база даних.
- 8) веб-додаток завантажує звіт зі сховища та надає його користувачу.

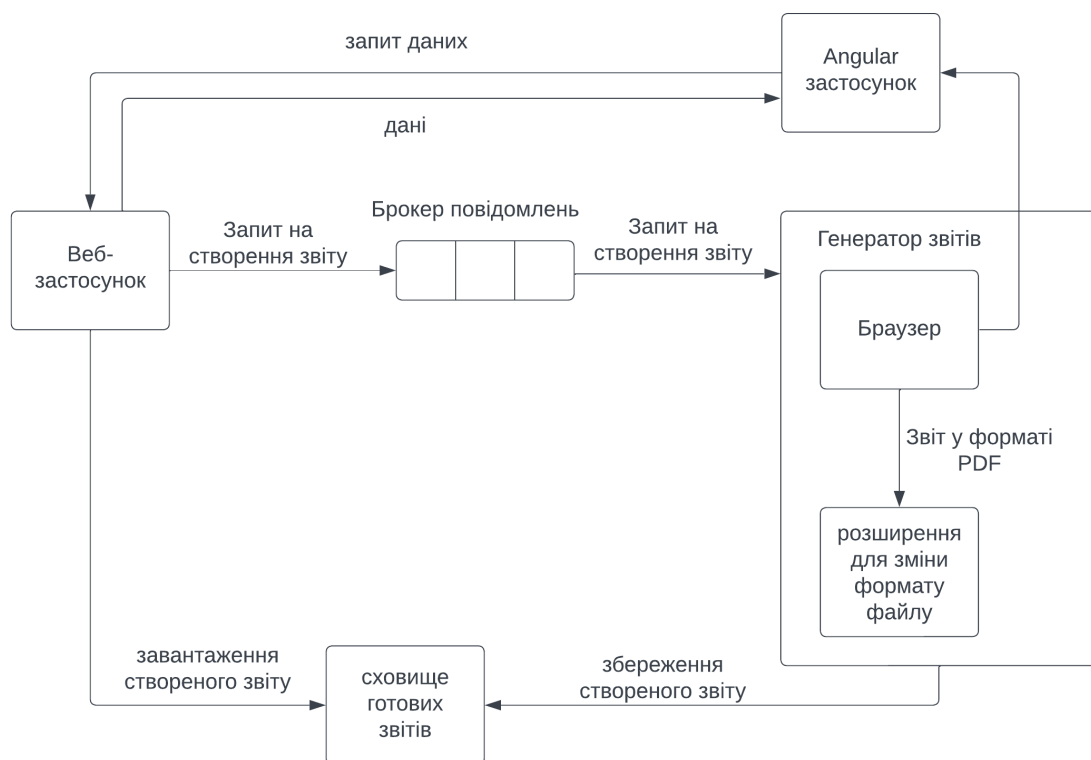


Рис. 2.3. Приклад архітектури системи звітування на основі Angular

## **Висновок**

У розділі були визначені функціональні та нефункціональні вимоги до системи створення звітів, детально розглянені фактори, що впливають на вибір архітектури, а також проаналізовані найбільш підходящі для реалізації розроблюваної системи технології та практики, використання яких дозволить створити таку систему, що буде водночас ефективною та простою у підтримці.



## РОЗДІЛ 3

### РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАСОБУ СПОСТЕРЕЖЕННЯ ЗА ЕКОНОМІЧНОЮ ЕФЕКТИВНІСТЮ КОМПАНІЙ

#### 3.1. Архітектура системи

В даній роботі буде проведено моделювання роботи систем звітності обох підходів, SSRS та Angular, що будуть об'єднані в одному Windows Service застосунку. Вибір системи, що буде створювати звіт, буде здійснюватись з користувацького інтерфейсу для кожного окремого запиту. Користувацький інтерфейс буде виконаний як Windows Presentation Foundation (WPF) застосунок. Для передачі повідомлень буде використаний брокер повідомлень RabbitMQ. Діаграма розроблюваної системи наведена на рис. 3.1.

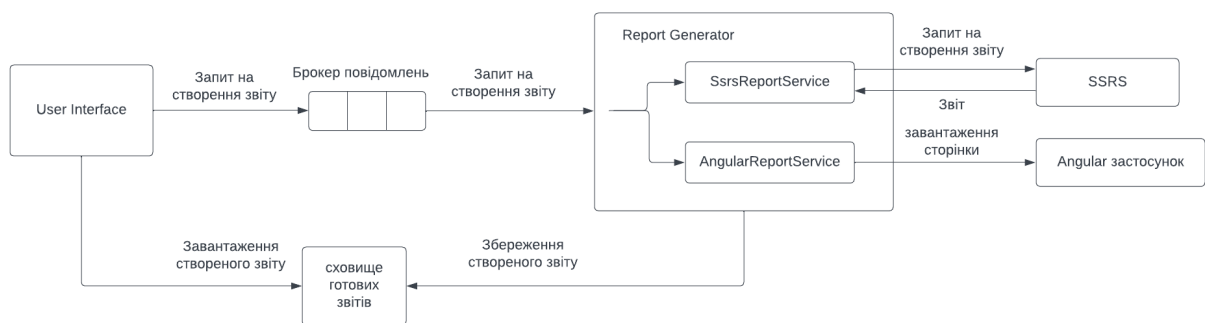


Рис. 3.1. Діаграма загальної архітектури системи звітності

В даній реалізації в звітах будуть використані статичні дані. Джерелом даних у випадку SSRS буде вбудований набір даних (від англ. «embedded data set»), Angular застосунок буде читати дані з XML файлів. Створені звіти будуть зберігатися у визначеній директорії.

Для розробки користувацького інтерфейсу та сервісу створення звітів буде використовуватись мова програмування C# платформи .NET Framework 4.6.1, для реалізації Angular застосунку буде використовувється платформа

Angular версії 17.0.5. Для створення звітів також буде використовуватись SQL Server 2016 та SSRS 2016.

### 3.2. Налаштування SSRS, створення шаблонів звіту

Для створення звітів на початку необхідно встановити SQL Server та SSRS відповідної версії, після чого можна перейти до налаштування сервісів створення звітів. Для цього необхідно використати Report Server Configuration Manager (RSCM), що буде встановлений разом із SSRS.

При першому відкритті RSCM запитує до якого сервера користувач бажає під'єднатись, в даному випадку це localhost. Наступним кроком необхідно під'єднатися до SQL Server баз даних ReportServer та ReportServerTempDB, що є одними з обов'язкових складових SSRS. Для підключення є дві опції: створити нові бази або під'єднатися до вже існуючих. В даному випадку створюємо нові бази, після налаштування їх можна побачити у SQL Server Management Studio (SSMS) (рис. 3.2).

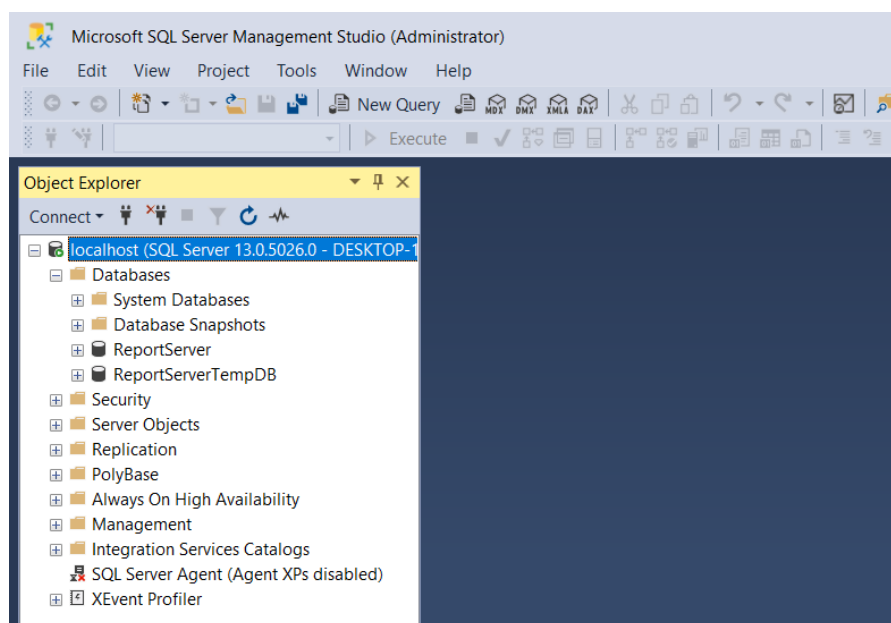


Рис. 3.2. Створені ReportServer та ReportServerTempDB бази даних

При необхідності можливо також змінити такі налаштування як сервіс акаунт, URL веб сервісу, тобто адресу для розгортання звітів на сервері, URL веб порталу, що використовується для керування звітами SSRS, джерелами

даних, параметрами звітів, тощо. Ці налаштування заповнюються автоматично значеннями за замовчуванням.

На цьому основні налаштування серверу закінчено, для перевірки необхідно відкрити веб портал (рис. 3.3).

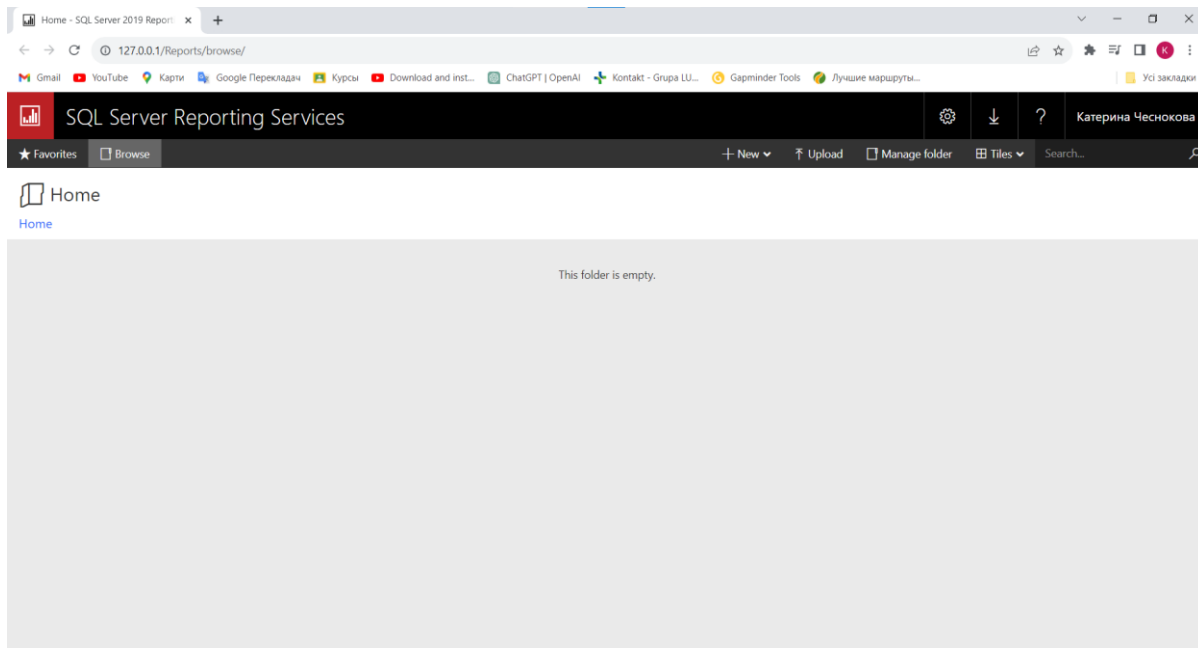


Рис. 3.3. Інтерфейс веб порталу SSRS

Наступним кроком необхідно створити та завантажити до SSRS веб порталу шаблон звіту RDL (рис. 3.4), де визначається спосіб завантаження даних та зовнішній вигляд звіту. Як зазначалося раніше, дані завантажуються з вбудованого набору даних (рис. 3.5).

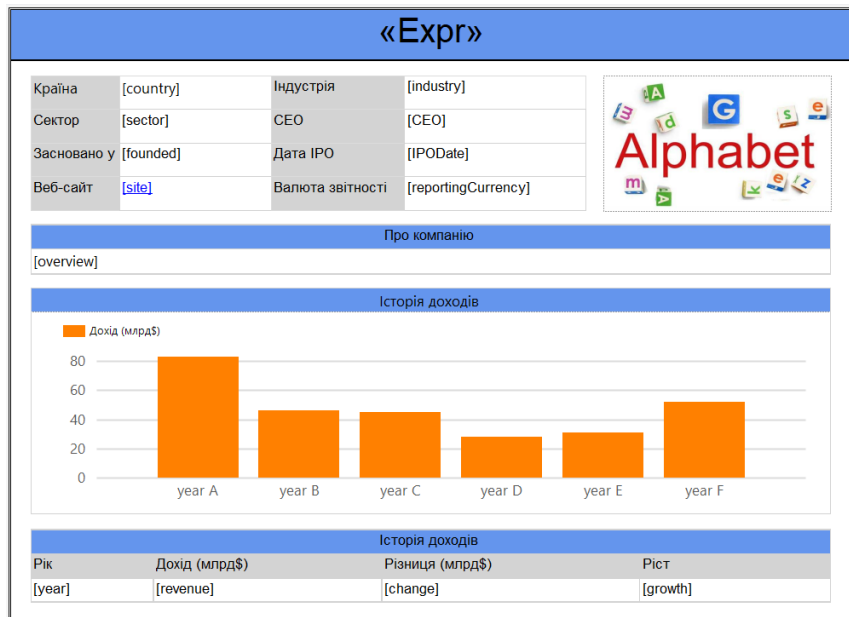


Рис. 3.4. Шаблон звіту

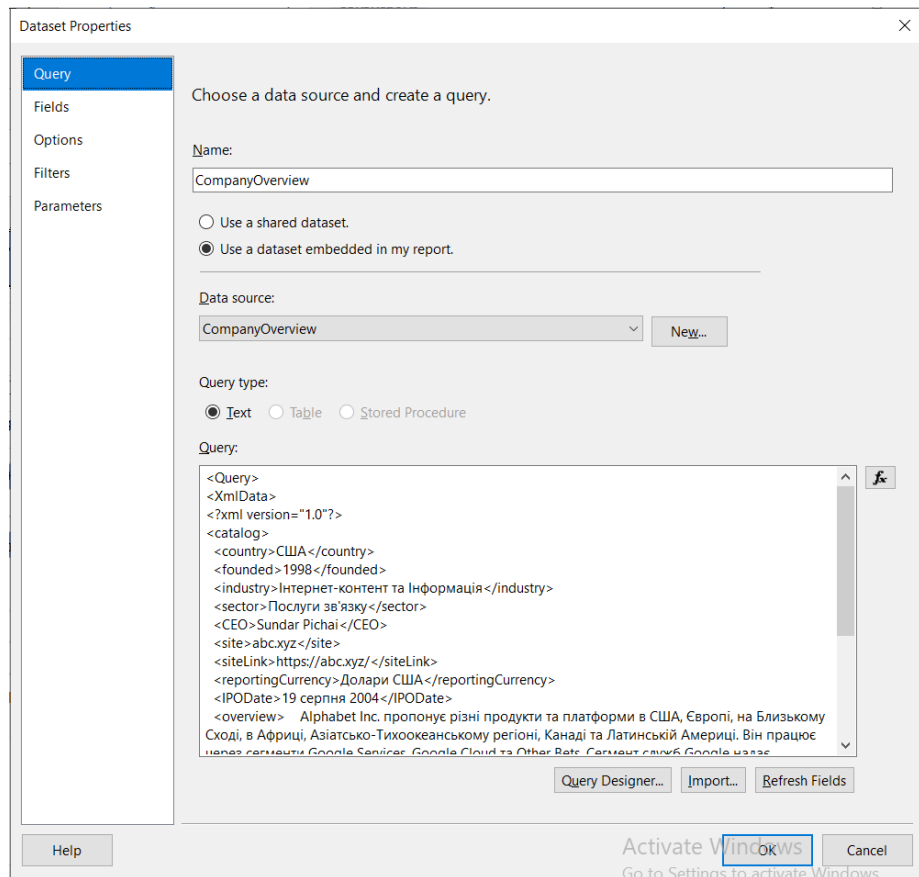


Рис. 3.5. Вбудований набір даних RDL файлу

Для включення можливостей контролю обсягу даних у звіті використовуються параметри, значення яких передаються під час створення звіту (рис. 3.6). Ці параметри використовуються у виразах «Hidden», що визначаються для кожного окремого елемента. Ці вирази визначають необхідність приховання елементів звіту: true – елемент необхідно приховати, false – елемент повинен бути видимим для користувача.

Angular шаблон побудований у відповідності із загальною архітектурою (рис. 3.7). Для реалізації шаблону створено два компоненти: report-page визначає загальне розташування елементів на сторінці звіту та її вигляд, revenue-chart описує стовпчасту діаграму. Також застосунок містить моделі даних та клас для читання даних з файлів XML.

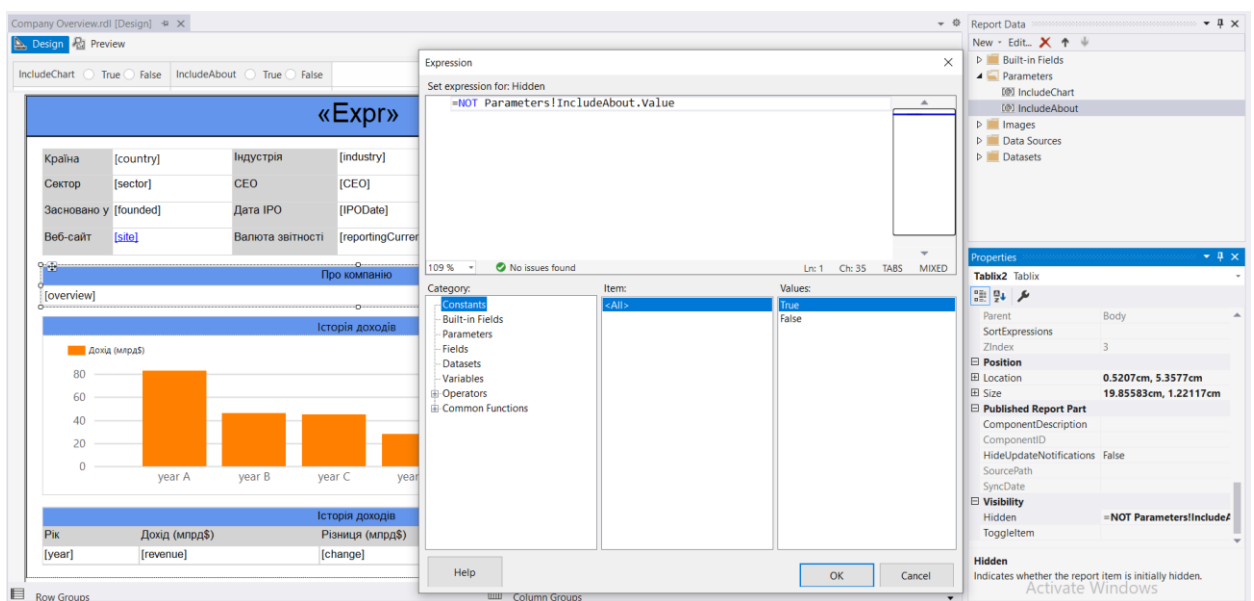


Рис. 3.6. Використання параметру «IncludeAbout»

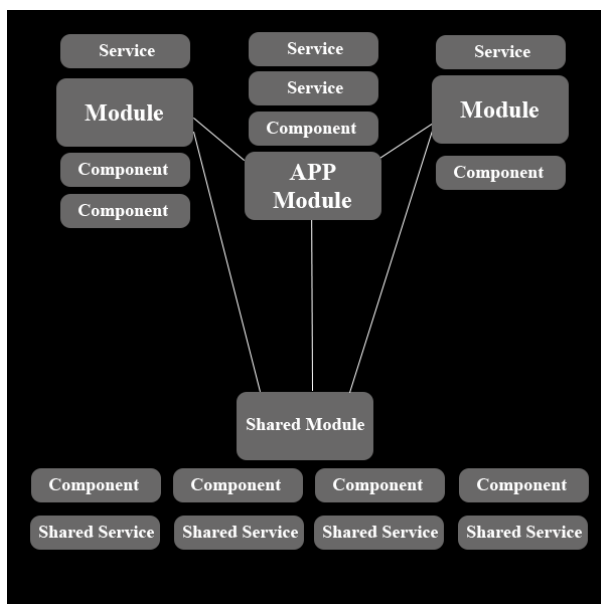


Рис. 3.7. Загальна архітектура Angular застосунків

### 3.3. Користувацький інтерфейс

Для реалізації користувацького інтерфейсу було обрано WPF застосунок - графічну підсистему, розроблену Microsoft для створення десктоп програм Windows. WPF є частиною платформи .NET і забезпечує уніфіковану модель програмування для створення насичених інтерактивних інтерфейсів користувача.

Застосунок має два вікна. Перше з них відображає список доступних для створення звітів. Разом зі створенням цього вікна відбувається перевірка робочого стану Angular застосунку. У випадку, якщо він не запущений, виконується команда «ng serve» у директорії Angular застосунку для його запуску (лістинг 3.1).

Лістинг 3.1.

#### Запуск Angular застосунку

```
private bool StartAngularHttpServer()
{
    string command = "ng serve";

    Process process = new Process();
```

```

    process.StartInfo.WorkingDirectory =
ConfigurationManager.AppSettings["AngularProjectPath"];
    process.StartInfo.FileName = "cmd.exe";
    process.StartInfo.Arguments = "/C" + command;
    process.StartInfo.UseShellExecute = false;
    process.StartInfo.RedirectStandardOutput = false;
    process.StartInfo.RedirectStandardInput = false;
    process.StartInfo.CreateNoWindow = false;

    bool started = process.Start();
    return started;
}

```

Друге вікно містить інтерфейс для звіту «Огляд компанії» (рис. 3.8). Користувач може виключити секцію «Про компанію» та діаграму зі звіту, обрати формат документу та систему створення звітів, що буде використовуватись. Доступні формати звіту для SSRS – PDF, MS Word, MS Excel, для Angular – тільки PDF.

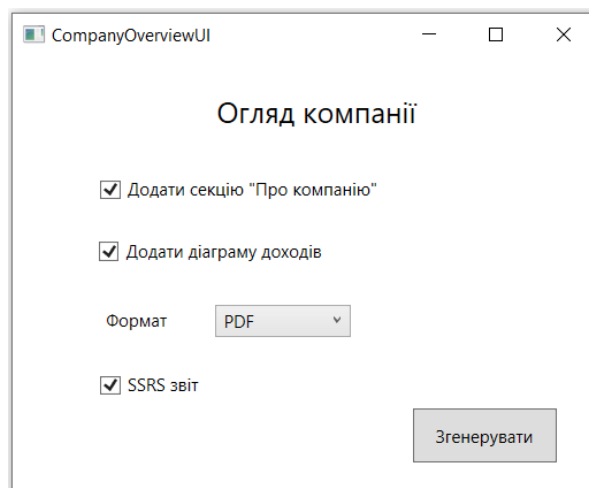


Рис. 3.8. Інтерфейс для створення звіту «Огляд компанії»

При натисканні кнопки «Згенерувати» формується запит на створення звіту та надсилається до брокера повідомлень RabbitMQ.

Лістинг 3.2.

## Клас запиту на створення звіту

```
public class ReportRequest
{
    public string TemplateName { get; set; }
    public DateTime Time { get; set; }
    public bool IsHtml { get; set; }
    public string FileName { get; set; }
    public List<ReportParameter> ReportParameters { get; set; }
    public ReportFormat Format { get; set; }
    public string SSRSReportPath { get; set; }
    public string AngularReportPath { get; set; }
}
```

Для коректної роботи RabbitMQ його необхідно встановити та запустити на комп'ютері (рис. 3.9). За замовчуванням він працює на порту 15672 і не потребує спеціальних налаштувань для початку роботи. Черга «Reporting Queue», що використовується в системі, створюється автоматично при записі першого повідомлення.

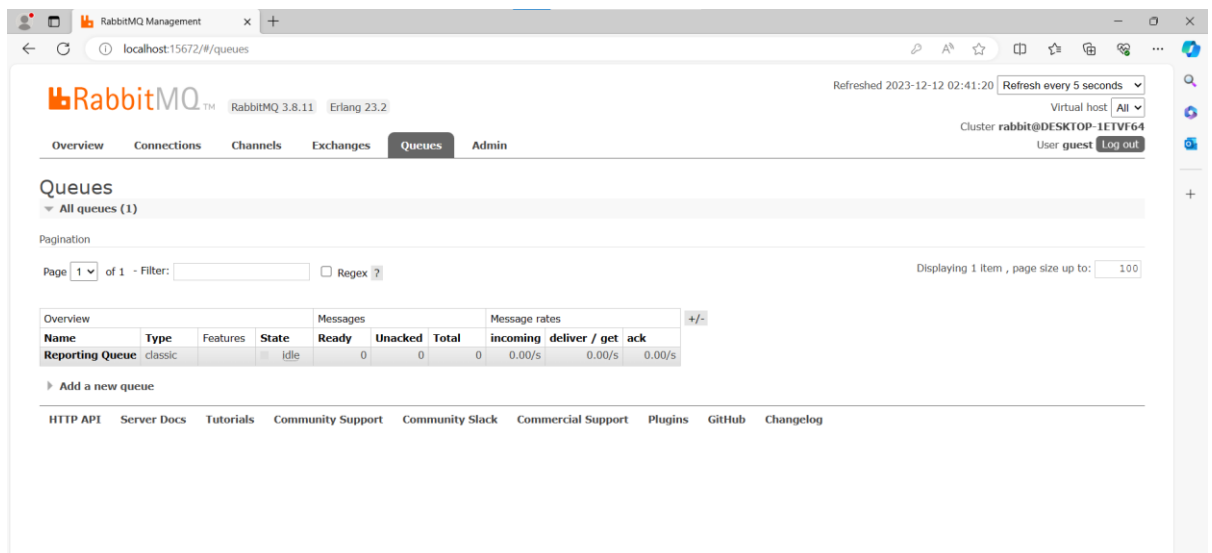


Рис. 3.9. Робота RabbitMQ на локальному комп'ютері

Одразу після запису повідомлення до RabbitMQ запускається FileSystemWatcher об'єкт, що шукає створений звіт у конкретній директорії, і



як тільки звіт буде створений, відкриває його. Для відображення звіту використовується програма, що встановлена на комп'ютері як програма за замовчуванням для конкретного розширення файлів.

Лістинг 3.3.

### Використання FileSystemWatcher

```
_fileSystemWatcher.Filter = _fileName;
_fileSystemWatcher.Created += OnFileCreated;
_fileSystemWatcher.EnableRaisingEvents = true;
private void OnFileCreated(object sender, FileSystemEventArgs e)
{
    if (e.Name.Equals(_fileName,
StringComparison.OrdinalIgnoreCase))
    {
        var filePath =
System.IO.Path.Combine(ConfigurationManager.AppSettings["Reports
Folder"], _fileName);
        Process.Start(filePath);
        Dispatcher.Invoke(() =>
        {
            loadingLabel.Visibility = Visibility.Hidden;
        });
    }
}
```

### 3.4. Підсистема створення звітів

Разом із запуском Windows сервісу «Generator» виконується підключення до брокера повідомлень, що дозволяє відстежувати нові запити на створення звіту, що були розміщені у черзі, та одразу починати їх обробку. Діаграма класів сервісу «Generator» зображена на рис. 3.10.

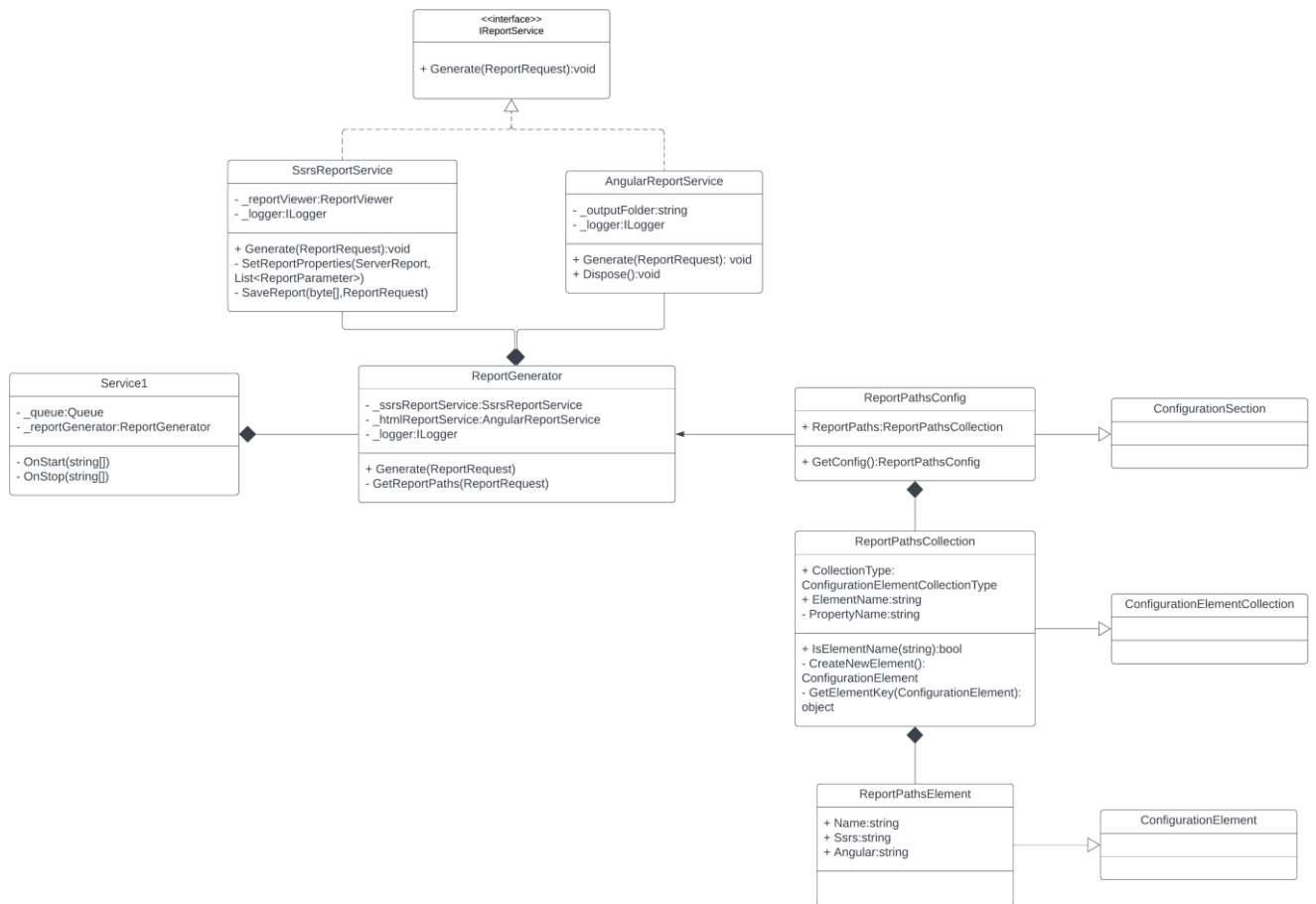


Рис. 3.10. Діаграма класів сервісу «Generator»

При отриманні нового запиту спочатку визначаються SSRSReportPath та AngularReportPath параметри, що будуть необхідні для подальшого виконання програми. Значення цих параметрів для кожного звіту визначаються у App.config файлі та зчитуються за допомогою класів спеціальної конфігурації. Подальше виконання програми визначається за значенням параметру IsHtml отриманого запиту.

Лістинг 3.4.

### Спеціальна конфігурація «reportPaths»

```
<configuration>
  <reportPaths>
    <reports>
```

```

        <add name="Company Overview" ssrs="/SSRSReports/Company
Overview" angular="" />
    </reports>
</reportPaths>
</configuration>

```

SSRS звіти створюються за допомогою ReportViewer, що є елементом керування, який дозволяє вбудовувати звіти в програми, розроблені за допомогою .NET Framework. Він може обробляти та відтворювати звіти незалежно за допомогою вбудованого механізму (локальний режим) або може відображати звіти, які обробляються та відображаються на сервері звітів (віддалений режим).

В сервісі використаний віддалений режим створення звітів.

В обох шаблонах присутні параметри, що визначаються користувачем під час створення запиту на генерацію звіту, та підставляються у шаблони під час виконання програми. В SSRS звітах ці параметри передаються у екземпляр класу ServerReport через метод SetParameters.

Лістинг 3.5.

#### Налаштування ReportViewer та створення звіту

```

private ReportViewer _reportViewer;
_reportViewer = new ReportViewer();
_reportViewer.ProcessingMode = ProcessingMode.Remote;
_reportViewer.ServerReport.ReportServerUrl = new
Uri("http://localhost/ReportServer");
_reportViewer.ServerReport.ReportPath =
reportRequest.SSRSReportPath;
SetReportParameters(_reportViewer.ServerReport,
reportRequest.ReportParameters);

```

```

byte[] renderedBytes =
_reportViewer.ServerReport.Render(reportRequest.Format.ToString(
));

private void SetReportParameters(ServerReport serverReport,
List<ReportParameter> parameters)
{
    foreach (var param in parameters)
    {
        serverReport.SetParameters(param);
    }
}

```

Робота Angular звітів побудована на використанні бібліотеки CefSharp, що дозволяє вбудувати повнофункціональний веб-браузер, сумісний зі стандартами, у програму C# або VB.NET. CefSharp має елементи керування браузером для програм WinForms і WPF, а також закадрову (від англ. «offscreen») версію. CefSharp базується на Chromium Embedded Framework, версії Google Chrome з відкритим кодом.

Передача параметрів відбувається за допомогою JavaScript коду, що виконується відносно основного вікна браузеру.

Цей код повинен виконуватись до повного завантаження сторінки, адже значення параметрів на цьому етапі вже мають бути присутні.

Лістинг 3.6.

#### Передача параметрів до вбудованого браузеру

```

var browser = new
ChromiumWebBrowser(reportRequest.AngularReportPath);
StringBuilder paramsString = new
StringBuilder("console.log('ScriptExecutionCompletedSignal'");
");
for (int i = 0; i < reportRequest.ReportParameters.Count; i++)

```

```

{
    var param = reportRequest.ReportParameters[i];
    paramsString.Append($"window.{param.Name} =
'{param.Values[0]}'");
}

string script = paramsString.ToString();
string scriptToInject = @"var script =
document.createElement('script');
    script.textContent = `{script}`;
    document.head.appendChild(script);";
browser.ExecuteScriptAsync(scriptToInject);

```

Збереження в файл для Angular звітів відбувається за допомогою вбудованого методу браузера тільки після повного завантаження сторінки, в той час як SSRS звіти можна зберігати за допомогою класу File.

Лістинг 3.7.

### Збереження Angular звіту

```

browser.FrameLoadEnd += async (sender, args) =>
{
    if (args.Frame.IsMain)
    {
        await scriptExecutionCompletion.Task;
        await Task.Delay(1000);
        result = await browser.PrintToPdfAsync(outputPath,
new PdfPrintSettings()
    {

```

```

        PrintBackground = true
    });
}
};

```

Лістинг 3.8.

### Збереження SSRS звіту

```

byte[] renderedBytes =
_reportViewer.ServerReport.Render(reportRequest.Format.ToString(
));
string folder =
ConfigurationManager.AppSettings["ReportsFolder"];
File.WriteAllBytes(Path.Combine(folder, reportRequest.FileName),
renderedBytes);

```

Для логування подій у сервісі використовується Serilog, діагностична бібліотека запису подій для програм .NET, що налаштована на запис подій до інструменту Event Viewer (рис. 3.11). Event Viewer – це компонент, що входить до складу операційних систем сімейства Windows, який дозволяє адміністраторам переглядати список подій на локальному комп'ютері або на віддаленій машині.

Лістинг 3.9.

### Налаштування логеру Serilog

```

public class LoggingService{
    public ILogger Logger { get; }
    public LoggingService(){
        Logger = new LoggerConfiguration()
        .WriteTo.EventLog("Generator").CreateLogger();
    }
}

```

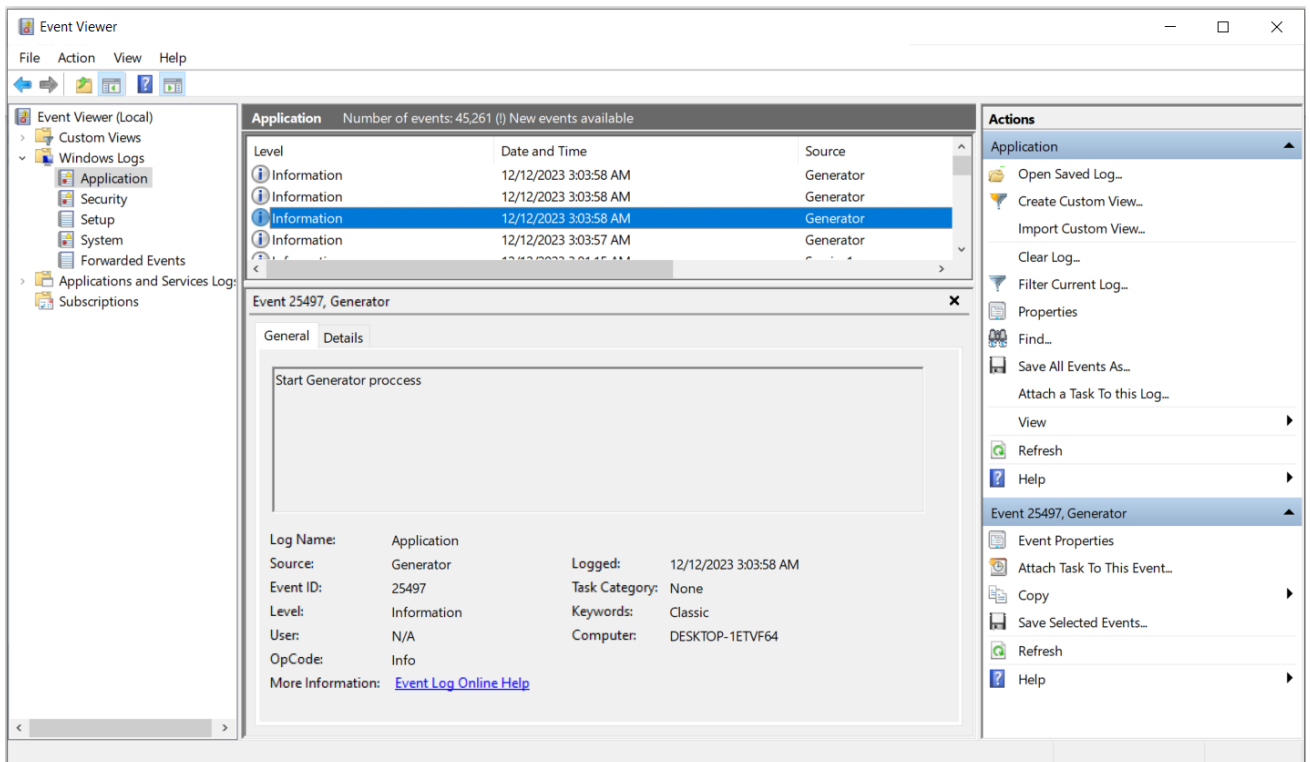


Рис. 3.11. Перегляд подій сервісу «Generator»

Для розгортання Windows Service на локальному комп'ютері необхідно виконати наступні команди у командному рядку.

Лістинг 3.10.

Команди для встановлення сервісу через командний рядок

```
sc create Generator binPath= "path to .exe file"
sc start Generator
```

## Висновки

У третьому розділі була описана архітектура системи створення звітів, виконана реалізація цієї системи та описані основні технології, використані у розробці. Також були розглянуті найважливіші аспекти реалізації застосунку.

## РОЗДІЛ 4

### РЕЗУЛЬТАТИ РОБОТИ СИСТЕМИ

#### 4.1. Результати роботи системи

Після встановлення сервісу «Generator» на локальний комп'ютер сервіс можна знайти у вбудованому інструменті операційної системи під назвою «Services» (рис. 4.1). Від надає легкий доступ до деяких налаштувань сервісів, а також дозволяє користувачу запускати чи зупиняти обраний сервіс.

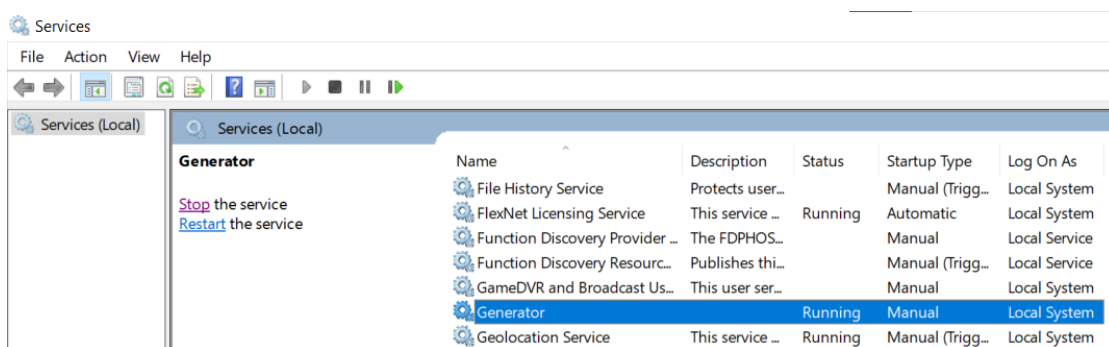
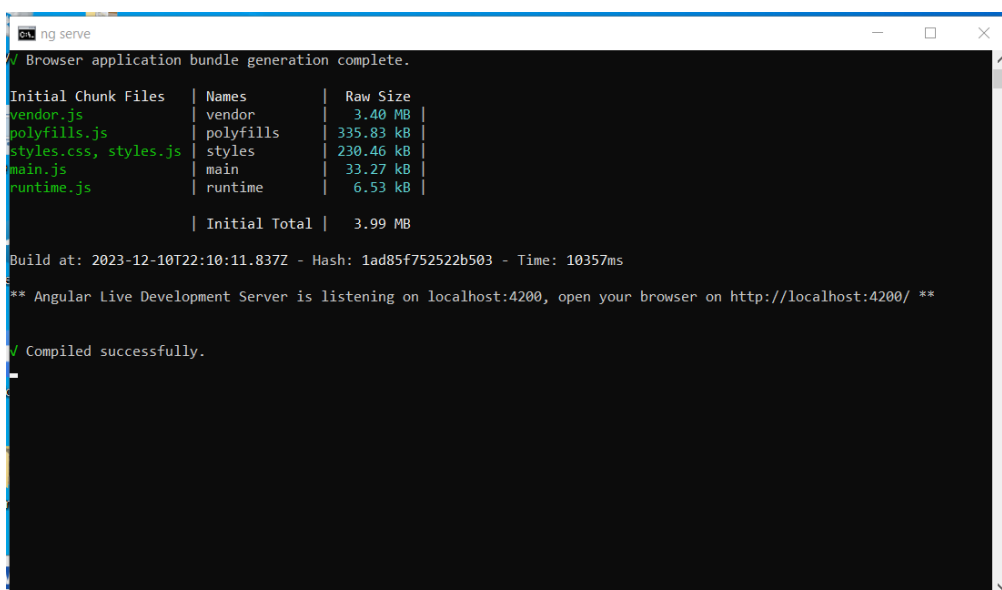


Рис. 4.1. Сервіс Generator встановлений на локальний комп'ютер

Під час запуску WPF додатку відбувається також запуск Angular додатку, що є необхідним для коректної роботи системи. Для відстеження процесу відкривається вікно командного рядка (рис. 4.2).



```
ng serve
✓ Browser application bundle generation complete.

Initial Chunk Files | Names | Raw Size
vendor.js | vendor | 3.40 MB |
polyfills.js | polyfills | 335.83 kB |
styles.css, styles.js | styles | 230.46 kB |
main.js | main | 33.27 kB |
runtime.js | runtime | 6.53 kB |

| Initial Total | 3.99 MB

Build at: 2023-12-10T22:10:11.837Z - Hash: 1ad85f752522b503 - Time: 10357ms

** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **

✓ Compiled successfully.
```



## Рис. 4.2. Запуск Angular додатку

Основні сторінки користувацького інтерфейсу приведені на рисунках 4.3 та 4.4.

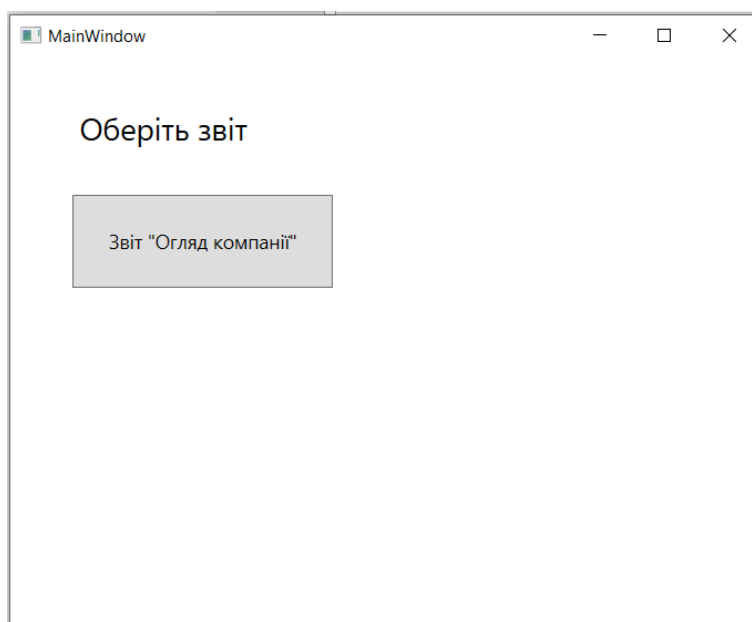


Рис. 4.3. Головна сторінка

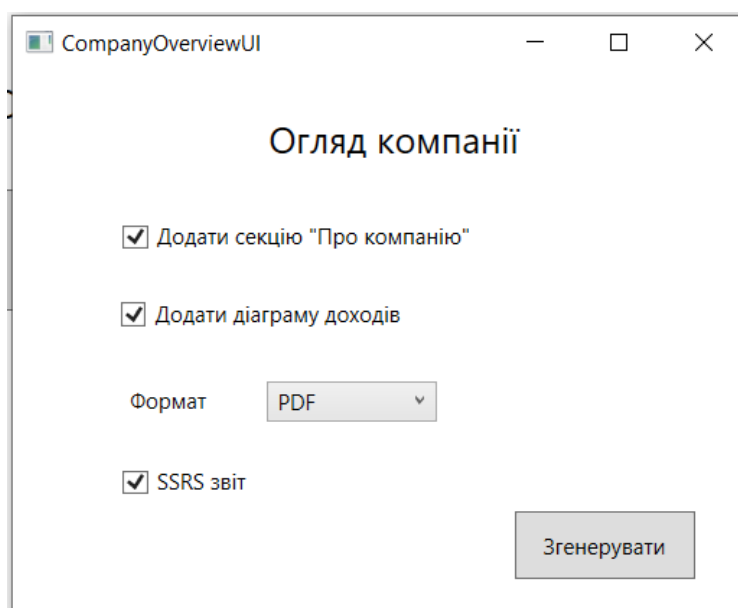


Рис. 4.4. Сторінка звіту «Огляд компанії»

Після натиску кнопки «Згенерувати» запит на створення звіту відправляється до RabbitMQ черги (рис. 4.5), що можна побачити у його кабінеті користувача.

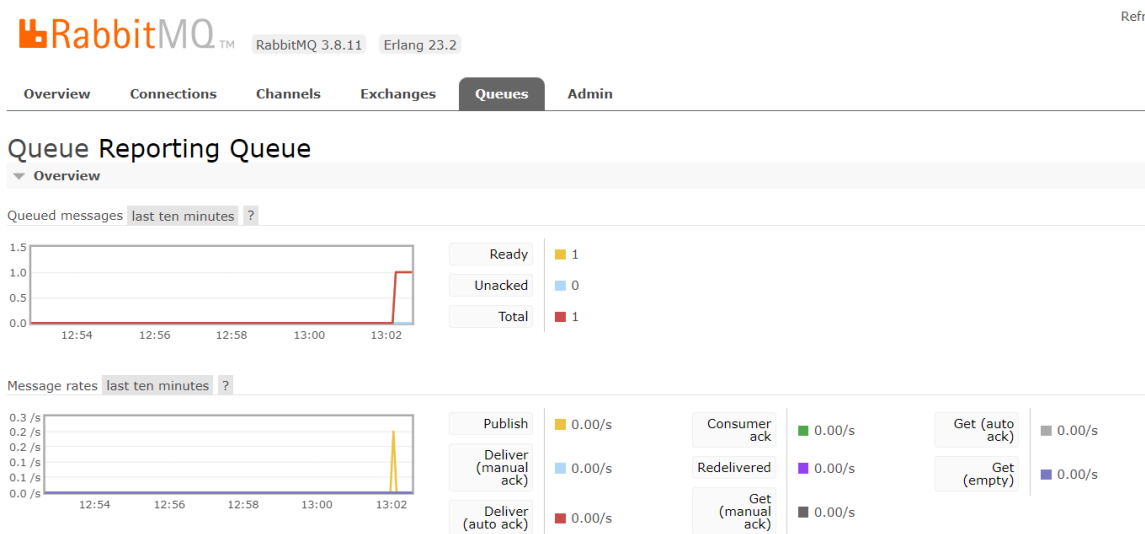


Рис. 4.5. Запит записано до черги

Сервіс Generator отримує повідомлення з черги, створює звіт відповідно до вибраних користувачем параметрів та зберігає файл до визначеної директорії (рис. 4.6).

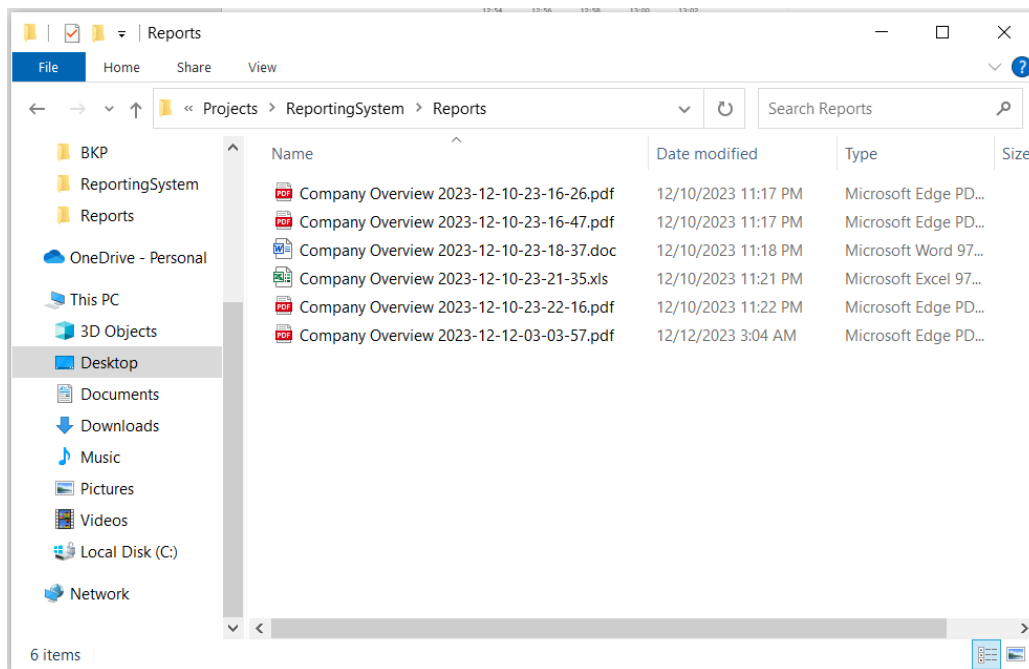


Рис. 4.6. Створені звіти зберігаються у директорії

Як тільки файл збережено, запускається процес, що відкриває файл у програмі за замовчуванням. На рисунках 4.7 – 4.10 наведені звіти, згенеровані системою, у всіх доступних форматах.

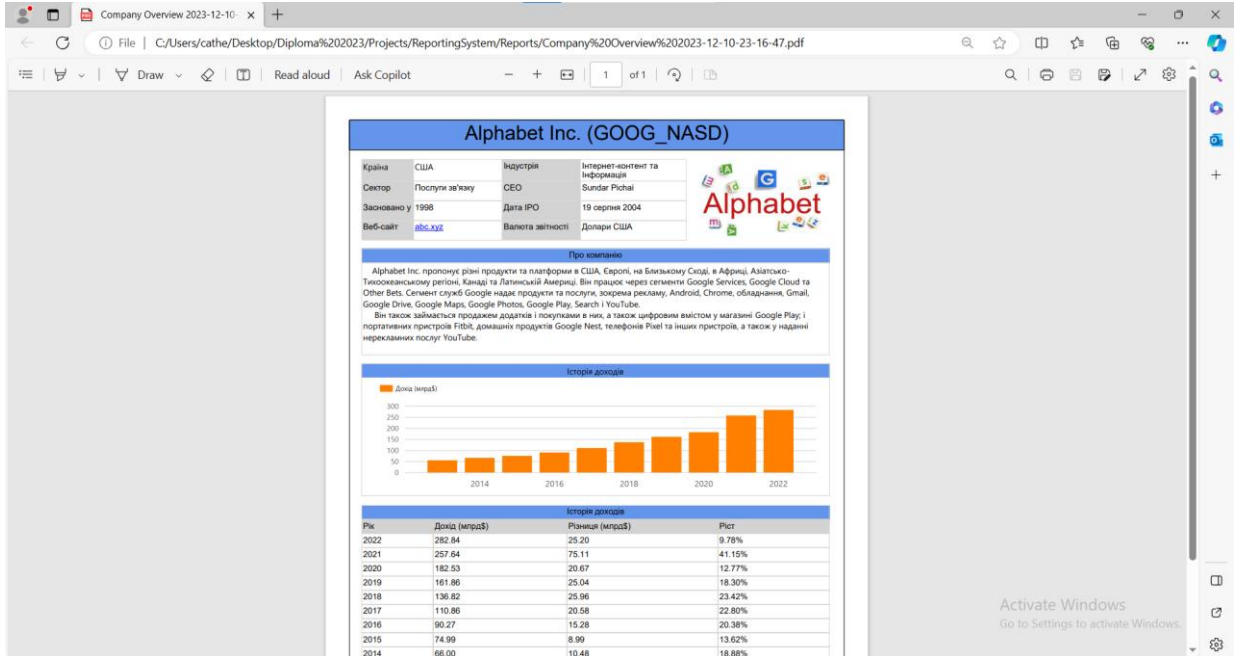


Рис. 4.7. SSRS звіт у форматі PDF

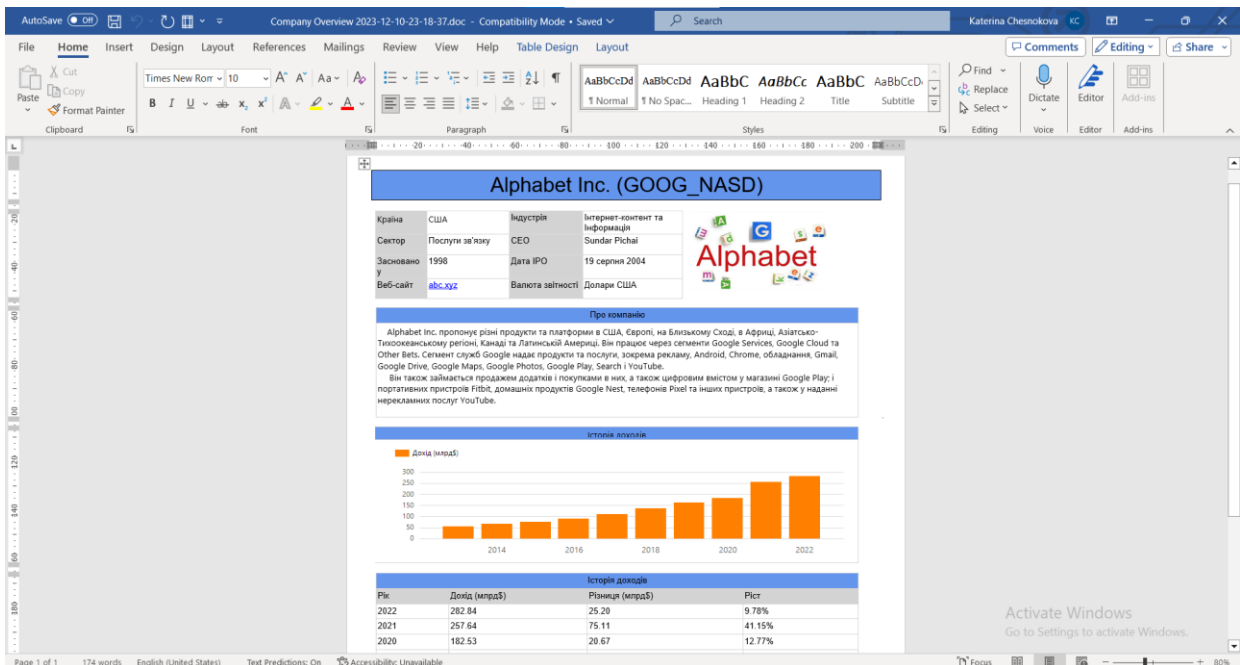


Рис. 4.8. SSRS звіт у форматі MS Word

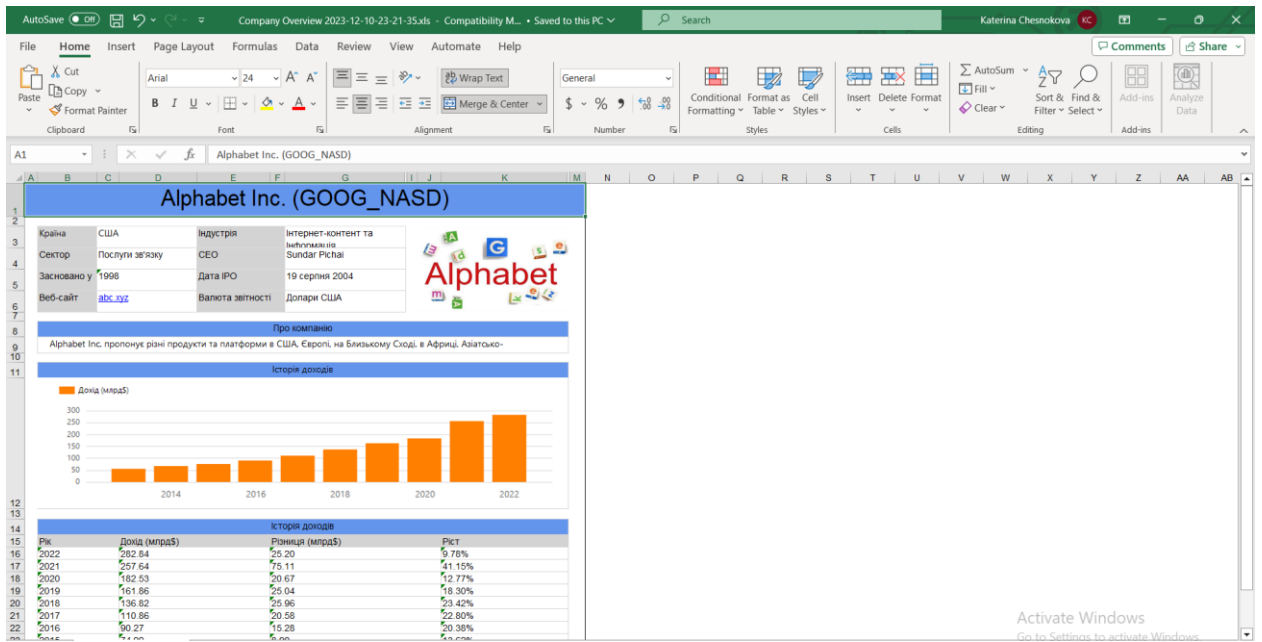


Рис. 4.9. SSRS звіт у форматі MS Excel

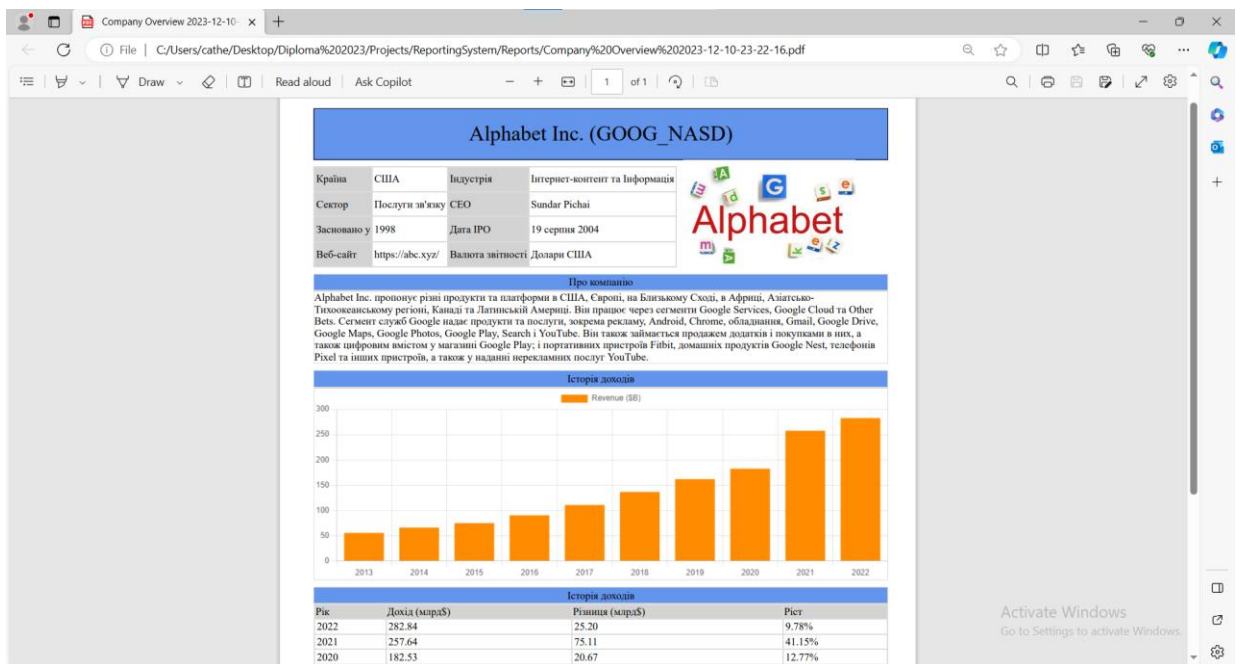


Рис. 4.10. Angular звіт у форматі PDF

## 4.2. Напрямки розширення функціоналу

Одними з можливих варіантів розширення функціоналу системи, які можуть підвищити як її універсальність, так і досвід користувача, є:

- підтримка більшої кількості форматів для Angular звітів. Для досягнення цього необхідно використовувати сторонні бібліотеки, наприклад бібліотеки Aspose;
- удосконалення інтерфейсу користувача, запровадження більш гнучкого та динамічного підходу до його побудови;
- розширення системи звітності вдосконаленими інструментами візуалізації даних;
- адаптивний дизайн для різноманітних пристроїв, що забезпечить безперебійну роботу користувачів на настільних ПК, планшетах і мобільних пристроях;
- введення можливості створення звітів за графіком, що дозволить забезпечити своєчасний доступ користувачів до важливої інформації.

## Висновки

У четвертому розділі була продемонстрована робота системи, приведено її основні сторінки, що доступні користувачам, та проаналізовані можливі напрямки розширення функціоналу програми.

## ВИСНОВКИ

Сучасні компанії знаходяться в постійному стані трансформації та адаптації, що зумовлено впливом глобалізації та стрімкого розвитку технологій. Такі обставини є не тільки викликом, але й шансом для просування та росту компанії. В такий час визначення економічної ефективності стає вирішальним для підтримки конкурентоспроможності.

Інвестори, зі свого боку, також зацікавлені в оцінці економічної ефективності компаній для визначення їх інвестиційної привабливості. При проведенні аналізу вони мають зібрати та проаналізувати велику кількість інформації. Полегшити цю задачу допоможе система створення звітності, що дозволяє експортувати інформацію до документів різних форматів і систематизувати її.

У даній роботі було досліджено предметну область спостереження за ефективністю компаній з точки зору інвесторів. Були розглянуті такі ключові аспекти, як фінансові показники, стратегії управління та ризиків, що дозволяє інвесторам отримати повну картину економічної стабільності та потенціалу компаній. Також в роботі розглянуті найбільш відомі існуючі системи для перегляду фінансових показників компаній, проаналізовані їхні переваги та недоліки, а також функціонал створення звітів, який вони надають користувачам.

В роботі були розглянуті різні варіанти архітектур систем створення звітів, а також розроблено та продемонстровано роботу системи на основі платформи .NET Framework, що об'єднує в собі два підходи до створення звітів та підтримує експорт документів різних форматів.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. КРАВЧУК А. В., ПЕРЕПВА П. Г. Efficiency as an economic category - CORE. CORE – Aggregating the world’s open access research papers. [Електронний ресурс]. – Режим доступу до ресурсу : [https://core.ac.uk/display/161789675?utm\\_source=pdf&utm\\_medium=banner&utm\\_campaign=pdf-decoration-v1](https://core.ac.uk/display/161789675?utm_source=pdf&utm_medium=banner&utm_campaign=pdf-decoration-v1).
2. Хівренко Т. А. Сутність поняття ефективності діяльності підприємства. Repository Simon Kuznets Kharkiv National University of Economics: Главная страница. [Електронний ресурс]. – Режим доступу до ресурсу: <http://repository.hneu.edu.ua/bitstream/123456789/14095/1/X|вренко.pdf>.
3. Архіпов Н. М. ВИДИ ЕФЕКТИВНОСТІ ОПЕРАЦІЙНОЇ ДІЯЛЬНОСТІ ПІДПРИЄМСТВА ТОРГІВЛІ. Науковий вісник Ужгородського національного університету. 2018. № 18. [Електронний ресурс]. – Режим доступу: [http://www.visnyk-econom.uzhnu.uz.ua/archive/18\\_1\\_2018ua/6.pdf](http://www.visnyk-econom.uzhnu.uz.ua/archive/18_1_2018ua/6.pdf).
4. Лобов С. П. Сучасні концепції економічної ефективності діяльності та ефективності управління підприємством. Ефективна економіка. 2015. № 4. [Електронний ресурс]. – Режим доступу до ресурсу : <http://www.economy.nayka.com.ua/?op=1&z=4011>.
5. Кирилова Л.І. Принципи оцінки економічної ефективності інвестиційних проектів CORE – Aggregating the world’s open access research papers. [Електронний ресурс]. – Режим доступу до ресурсу : <https://core.ac.uk/download/pdf/38334072.pdf>.
6. LibreTexts. Виробничі можливості кордону та соціальний вибір. LibreTexts - Ukrayinska. [Електронний ресурс]. – Режим доступу до ресурсу : <https://ukrayinska.libretexts.org/>

7. ОПЕРАЦІЙНА ЕФЕКТИВНІСТЬ: визначення, приклади та все, що потрібно знати. About Us. [Електронний ресурс]. – Режим доступу до ресурсу: <https://businessyield.com/uk/bs-business/operational-efficiency/>.
8. Isuru S. Financial Statement Analysis: A Comprehensive Guide for Investment Purposes. LinkedIn: Log In or Sign Up. [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.linkedin.com/pulse/financial-statement-analysis-comprehensive-guide-investment-isuru>.
9. 4 Benefits of Optimized Reporting and Analytics in an Organization. Blue Link ERP. [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.bluelinkerp.com/blog/4-benefits-of-optimized-reporting-and-analytics-in-an-organization/>.
10. Introduction to Windows Service Applications - .NET Framework. Microsoft Learn: Build skills that open doors in your career. [Електронний ресурс]. – Режим доступу: <https://learn.microsoft.com/en-us/dotnet/framework/windows-services/introduction-to-windows-service-applications>.
11. What Are .NET Worker Services? - Steve Gordon - Code with Steve. Steve Gordon - Code with Steve. [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.stevejgordon.co.uk/what-are-dotnet-worker-services>.
12. Що таке Message Queue (черга повідомлень)? | ІТЕксперт. ІТ-рекрутингове агентство "ІТЕксперт". [Електронний ресурс]. – Режим доступу до ресурсу: <https://itexpert.work/uk/chergha-povidomlen-message-queue-mq/>.
13. Report Definition Language - SQL Server Reporting Services (SSRS). Microsoft Learn: Build skills that open doors in your career. [Електронний ресурс]. – Режим доступу: <https://learn.microsoft.com/en-us/sql/reporting-services/reports/report-definition-language-ssrs?view=sql-server-ver16>.
14. Editor. Pros and Cons of Angular Development Framework. AltexSoft. [Електронний ресурс]. – Режим доступу: <https://www.altexsoft.com/blog/the-good-and-the-bad-of-angular-development/>.



15. Architecture of SQL Server Reporting Services (SSRS). C# Corner - Community of Software and Data Developers. [Электронный ресурс]. – Режим доступа до ресурсу: <https://www.c-sharpcorner.com/UploadFile/35fef0/architecture-of-sql-server-reporting-servicesssrs/>.
16. Chapter 3: Reporting Services Architecture. CODE Online. [Электронный ресурс]. – Режим доступа: <https://www.codemag.com/Article/070103/Chapter-3-Reporting-Services-Architecture>.