

Київський столичний університет імені Бориса Грінченка  
Факультет інформаційних технологій та математики  
Кафедра комп'ютерних наук

**Допущено до захисту**

Завідувач кафедри комп'ютерних наук

доктор технічних наук, професор

Андрій БОНДАРЧУК

« 01 » червня 2026 р.

## **КВАЛІФІКАЦІЙНА РОБОТА**

**на здобуття освітнього ступеня «бакалавр»**

**Спеціальність 122 Комп'ютерні науки**

**Освітня програма 122.00.01 Інформатика**

**Тема: ІНТЕЛЕКТУАЛЬНА СИСТЕМА АНАЛІЗУ ПОКАЗНИКІВ  
ЕФЕКТИВНОСТІ ПРАЦІВНИКІВ МЕРЕЖІ КАВ'ЯРЕНЬ ІЗ  
ВИКОРИСТАННЯМ ДАНИХ ROS-СИСТЕМИ POSTER**

Виконав  
студент групи ІНБ-1-22-4.0д  
Ігнатенко Дем'ян Русланович,

---

Науковий керівник  
доктор філософії,  
Турукало А.В.

---

Київ – 2026

Київський столичний університет імені Бориса Грінченка  
Факультет інформаційних технологій та математики  
Кафедра комп'ютерних наук

«Затверджую»

Завідувач кафедри комп'ютерних наук  
доктор технічних наук, професор  
Андрій БОНДАРЧУК  
«01 » жовтня 2025 р.

**ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА**  
**«Інтелектуальна система аналізу показників ефективності працівників мережі**  
**кав'ярень із використанням даних POS-системи Poster»**

Виконавець – студент групи ІНБ-1-22-4.0д,  
спеціальності 122 Комп'ютерні науки,  
освітньої програми 122.00.01 Інформатика  
Ігнатенко Дем'ян Русланович

1. Вихідні дані: методичні рекомендації до виконання кваліфікаційної роботи, документація Poster API, програмна документація Python, результати власної розробки програмного засобу, приклади сформованих Excel-звітів.

2. Основні завдання: проаналізувати предметну область і підходи до формування управлінської звітності в мережі кав'ярень; дослідити можливості Poster API як джерела даних; визначити функціональні вимоги до системи; спроектувати архітектуру програмного засобу; розробити алгоритми формування рейтингу денних кас, бонусної статистики, weekly-аналітики та monthly-аналітики; реалізувати desktop-застосунок і модулі роботи з API, Excel та перевірки періодів; провести тестування роботи системи й обґрунтувати практичну цінність отриманих результатів.

3. Пояснювальна записка: вступ, три розділи, висновки, список використаних джерел, додатки. Орієнтовний обсяг – 35-45 сторінок формату А4 відповідно до вимог кафедри.

4. Графічні матеріали: презентація результатів роботи, приклади згенерованих Excel-звітів.

5. Додатки: скріншоти Excel-звітів.

6. Строк подання роботи на кафедру: «01» червня 2026 р.

Науковий керівник  
доктор філософії,  
\_\_\_\_\_ Турукало А.В.  
\_\_\_\_\_ Дата

Виконавець  
\_\_\_\_\_ Ігнатенко Д.Р.  
\_\_\_\_\_ Дата

## Календарний план

| №  | Етап виконання роботи                                                                               | Термін виконання    | Примітка |
|----|-----------------------------------------------------------------------------------------------------|---------------------|----------|
| 1  | Формування теми дипломної роботи бакалавра та її затвердження                                       | 01.10.25 - 31.10.25 | виконано |
| 2  | Аналіз предметної області, огляд існуючих рішень та написання першого розділу пояснювальної записки | 01.11.25 - 20.11.25 | виконано |
| 3  | Проектування архітектури застосунку на базі клієнт-серверної взаємодії та API-інтеграції            | 21.11.25 - 20.12.25 | виконано |
| 4  | Розробка функціональних модулів застосунку та програмної логіки обробки даних                       | 21.12.25 - 20.01.26 | виконано |
| 5  | Інтеграція клієнтської та серверної частин системи, налаштування взаємодії компонентів              | 21.01.26 - 15.02.26 | виконано |
| 6  | Реалізація основного функціоналу програмного застосунку та тестування модулів                       | 16.02.26 - 28.02.26 | виконано |
| 7  | Впровадження розробленого програмного рішення та його налаштування для використання                 | 01.03.26 - 10.03.26 | виконано |
| 8  | Розробка механізму обробки та збереження даних у системі                                            | 11.03.26 - 15.03.26 | виконано |
| 9  | Написання другого та третього розділів пояснювальної записки                                        | 16.03.26 - 15.04.26 | виконано |
| 10 | Подання роботи на перевірку, проходження антиплагіату, внесення правок керівника                    | 29.04.26 - 15.05.26 | виконано |
| 11 | Підготовка презентаційних матеріалів, тексту доповіді та захист кваліфікаційної роботи              | 16.05.26 - 15.06.26 | виконано |

**«Затверджую»**

**Науковий керівник**

доктор філософії, Турукало А.В.

**Індивідуальний план склав**

Ігнатенко Д.Р.

## РЕФЕРАТ

**Кваліфікаційна робота:** 48 с., 8 табл., 8 рис., 1 додаток.

**Актуальність.** Для мережі кав'ярень критично важливими є швидке отримання аналітичних показників, оцінювання ефективності працівників, порівняння локацій і формування управлінських висновків. Ручна підготовка звітності займає значний час, створює ризик помилок і ускладнює оперативне прийняття рішень. Використання даних POS-системи Poster дає можливість автоматизувати ці процеси та забезпечити більш системний підхід до аналізу продажів та KPI.

**Об'єкт дослідження:** процес аналізу ефективності працівників і формування операційної звітності в мережі кав'ярень.

**Предмет дослідження:** методи, моделі та програмні засоби автоматизованого збору, обробки, порівняння та інтерпретації даних POS-системи Poster для формування управлінської звітності.

**Мета роботи:** розробити інтелектуальну систему аналізу ефективності працівників та формування звітності мережі кав'ярень на основі даних POS-системи Poster у вигляді локального desktop-застосунку з генерацією Excel-файлів.

### **Завдання:**

1. проаналізувати предметну область операційної діяльності мережі кав'ярень як розподіленої системи обслуговування клієнтів;
2. дослідити можливості Poster API для автоматизованого збору інформації з хмарної POS-системи, нормалізації її та використанні у подальшій інтелектуальній обробці й агрегації аналітичних звітів;
3. визначити функціональні вимоги до інтелектуальної системи аналізу ефективності працівників та формування звітності мережі кав'ярень;

4. спроектувати архітектуру програмного засобу для мінімізації зв'язності коду, спрощення тестування, забезпечення надійної взаємодії компонентів та полегшення подальшого супроводу й розширення системи;

5. реалізувати модулі роботи з даними, побудови звітів і валідації періодів; провести тестування та оцінити практичну доцільність використання розробки.

**Основні результати.** У роботі реалізовано програмний засіб на Python, який взаємодіє з Poster API, формує Excel-звіти за декількома сценаріями, виконує побудову рейтингу денних кас, бонусної статистики, тижневих і місячних аналітичних звітів, порівнює показники між періодами, а також генерує rule-based рекомендації для керівника. Реалізовано desktop-інтерфейс на tkinter із перевіркою коректності вибраного періоду та повідомленнями про помилки.

**Практичне значення дослідження** полягає у створенні повнофункціонального прикладного рішення, яке трансформує розрізнені операційні дані POS-системи у структурований інструмент стратегічного та оперативного управління.

**Ключові слова:** POSTER API, POS-система, KPI, аналітика продажів, мережа кав'ярень, PYTHON, OPENPYXL, TKIN

## ЗМІСТ

|                                                                             |    |
|-----------------------------------------------------------------------------|----|
| РЕФЕРАТ                                                                     | 4  |
| ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ                                      | 7  |
| ВСТУП                                                                       | 8  |
| РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА                           |    |
| ЗАДАЧІ                                                                      | 12 |
| 1.1. Особливості операційної діяльності мережі кав'ярень як об'єкта аналізу | 12 |
| 1.2. Аналіз підходів до формування аналітичної звітності                    | 14 |
| 1.3. Аналіз можливостей POS-системи Poster як джерела даних                 | 17 |
| 1.4. Постановка задачі розробки інтелектуальної системи                     | 20 |
| Висновки до розділу 1                                                       | 21 |
| РОЗДІЛ 2. ПРОЄКТУВАННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ                              |    |
| АНАЛІЗУ ТА ФОРМУВАННЯ ЗВІТНОСТІ                                             | 22 |
| 2.1. Функціональні вимоги до системи                                        | 22 |
| 2.2. Архітектура програмної системи                                         | 25 |
| 2.3. Модель обробки даних та джерела інформації                             | 29 |
| Висновки до розділу 2                                                       | 32 |
| РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАСОБУ ТА АНАЛІЗ                           |    |
| РЕЗУЛЬТАТІВ                                                                 | 34 |
| 3.1. Засоби та технології реалізації                                        | 34 |

|                                                      |    |
|------------------------------------------------------|----|
| 3.2. Реалізація модуля взаємодії з Poster API        | 36 |
| 3.3. Реалізація модуля формування Excel-звітів       | 38 |
| 3.4. Тестування та аналіз результатів роботи системи | 41 |
| Висновки до розділу 3                                | 42 |
| ВИСНОВКИ                                             | 43 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ                           | 45 |
| ДОДАТКИ                                              | 48 |

## ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

| Скорочення | Розшифрування                                                             |
|------------|---------------------------------------------------------------------------|
| API        | Application Programming Interface, програмний інтерфейс прикладного рівня |
| GUI        | Graphical User Interface, графічний інтерфейс користувача                 |
| JSON       | JavaScript Object Notation, текстовий формат обміну даними                |
| KPI        | Key Performance Indicators, ключові показники ефективності                |
| POS        | Point of Sale, система обліку та супроводу продажів                       |
| XLSX       | формат електронних таблиць Microsoft Excel                                |

## ВСТУП

У сучасних умовах діяльність підприємств сфери громадського харчування дедалі більше залежить від якості використання операційних даних. Для мережі кав'ярень особливо важливими є своєчасний контроль показників продажів, оцінювання результативності працівників, аналіз роботи окремих локацій і швидке формування зрозумілої управлінської звітності. На практиці значна частина такої аналітики виконується вручну або напівавтоматично, що збільшує витрати часу, створює ризик помилок і ускладнює прийняття обґрунтованих управлінських рішень. Потреба в системному підході до аналітичної діяльності в закладах харчування узгоджується з вимогами до бакалаврських кваліфікаційних робіт і професійними результатами навчання за спеціальністю 122 [1–3].

Сучасні POS-системи акумулюють значні обсяги структурованих даних про продажі, чеки, працівників, категорії товарів і результати роботи локацій. Однією з таких систем є Poster, що надає програмний інтерфейс доступу до даних і дозволяє інтегрувати зовнішні програмні засоби аналітики [6]. Проте наявність доступу до даних сама по собі не вирішує проблеми керівника: інформацію потрібно не лише отримати, а й обробити, зіставити між періодами, інтерпретувати та подати в зручному вигляді.

Актуальність теми полягає в необхідності створення програмного засобу, який автоматизує формування аналітичної звітності мережі кав'ярень на основі даних POS-системи Poster, зменшує обсяг ручної роботи, підвищує точність розрахунків і забезпечує підтримку прийняття рішень. Особливого значення набуває автоматизоване оцінювання ефективності працівників і локацій, побудова порівняльних звітів за тиждень і місяць та формування рекомендацій на основі обчислених показників.



Метою кваліфікаційної роботи є розробка інтелектуальної системи аналізу ефективності працівників та формування звітності мережі кав'ярень на основі даних POS-системи Poster у вигляді локального desktop-застосунку для автоматизації збору, обробки, аналізу та візуалізації операційних показників діяльності.

Для досягнення поставленої мети необхідно розв'язати такі завдання: проаналізувати предметну область та існуючі підходи до формування звітності; дослідити можливості Poster API як джерела даних; визначити функціональні вимоги до системи; спроектувати архітектуру програмного засобу; розробити алгоритми формування рейтингів, бонусної статистики, weekly-аналітики та monthly-аналітики; реалізувати desktop-застосунок і модулі системи; провести тестування та оцінити практичну доцільність використання розробки.

Об'єктом дослідження є процес аналізу ефективності працівників і формування операційної звітності в мережі кав'ярень. Предметом дослідження є методи, моделі та програмні засоби автоматизованого збору, обробки, порівняння й інтерпретації даних POS-системи Poster для оцінювання результативності працівників і локацій.

У роботі використано методи системного аналізу, аналізу даних, порівняльного аналізу, узагальнення, алгоритмізації, проектування програмних систем, модульної декомпозиції та тестування програмного забезпечення [7–10].

Практичне значення одержаних результатів полягає у створенні програмного засобу, який забезпечує автоматизоване формування Excel-звітів за декількома сценаріями, підтримує перевірку коректності обраного періоду, дозволяє порівнювати локації між собою та формує рекомендації для керівника. Елементи інтелектуальної складової системи реалізовано через

автоматичне обчислення похідних показників, rule-based класифікацію змін і генерацію підсумкових рекомендацій.

Структурно кваліфікаційна робота складається зі вступу, трьох розділів, висновків, списку використаних джерел і додатків. У першому розділі розглянуто предметну область, виконано аналіз підходів до формування аналітичної звітності та здійснено постановку задачі. У другому розділі подано проєктування системи, її архітектуру, модель обробки даних і алгоритми формування звітів. У третьому розділі висвітлено програмну реалізацію системи, описано desktop-застосунок, наведено результати тестування та приклади сформованих звітів.

Підготовка та оформлення кваліфікаційної роботи здійснювалися з урахуванням методичних рекомендацій кафедри, стандарту вищої освіти за спеціальністю 122 «Комп'ютерні науки», освітньо-професійної програми та вимог до структури науково-технічних звітів [1–5]. Крім того, під час оформлення бібліографічних посилань враховано вимоги ДСТУ 8302:2015 [6]. Такий підхід дозволив узгодити практичну програмну реалізацію з формальними вимогами до бакалаврської кваліфікаційної роботи.

Наукова новизна дослідження полягає у поєднанні в одному локальному desktop-застосунку декількох сценаріїв аналітики, що раніше виконувалися вручну: рейтингу працівників за денним виторгом, бонусної статистики, тижневої та місячної аналітики з порівнянням періодів. На відміну від типових звітів облікової системи, запропоноване рішення реалізує модульну архітектуру, власні правила інтерпретації динаміки та пояснювані управлінські сигнали, що відповідає сучасним підходам програмної інженерії й бізнес-аналітики [14; 20; 23].

Практична цінність результатів проявляється в тому, що розроблений засіб орієнтований на реальний сценарій використання керівником мережі

кав'ярень. Застосунок реалізовано як локальний desktop-інструмент на Python, який не потребує витрат на серверну інфраструктуру та працює безпосередньо на Windows, та дозволяє значно скоротити час на підготовку аналітики, який зазвичай витрачається на ручний експорт та зведення даних у таблиці формучи готові XLSX-файли й забезпечує зрозумілий інтерфейс для непідготовленого користувача, що особливо важливо для малих і середніх бізнес-процесів [17; 24]. Система не лише відображає статистику, а й автоматично інтерпретує динаміку показників (виторгу, трафіку, середнього чека) за допомогою rule-based підходу.

Це дозволяє керівнику миттєво отримувати пояснювані управлінські сигнали, такі як «падіння трафіку» чи «системне просідання», та відповідні рекомендації щодо маркетингових чи кадрових дій, тому система є готовим інструментом для оперативного управління, що дозволяє масштабувати бізнес без пропорційного зростання витрат на аналітичну роботу.

## РОЗДІЛ 1

### АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

#### 1.1 Особливості операційної діяльності мережі кав'ярень як об'єкта аналізу

Мережа кав'ярень є розподіленою системою обслуговування клієнтів, у якій кожна локація має власний трафік, структуру попиту, команду працівників і специфіку продажів. З управлінської точки зору важливими є не тільки абсолютні значення виторгу, а й супутні характеристики: кількість чеків, середній чек, розподіл продажів за товарними категоріями, результативність конкретних працівників та динаміка показників у часі.

Для мережі кав'ярень операційна діяльність є безперервним циклом коротких управлінських рішень. Менеджмент щоденно оцінює навантаження на локації, результативність зміни, стабільність середнього чека, структуру продажів та динаміку трафіку. Якщо така інформація надходить із затримкою або у фрагментованому вигляді, керівник не може своєчасно коригувати графіки працівників, маркетингові активності чи товарну матрицю. Тому в цій предметній області швидкість підготовки звіту майже так само важлива, як і його точність.

З практичної точки зору аналіз роботи мережі кав'ярень складається з декількох рівнів. На рівні працівника необхідно оцінювати денний виторг, кількість чеків, середній чек, кількість змін і внесок у продаж окремих категорій. На рівні локації важливими є сумарний виторг, трафік, середній чек, середній денний виторг і частка ключових категорій у структурі продажів. На рівні мережі керівника цікавить уже порівняння точок між собою та порівняння показників між періодами. Саме така багаторівнева структура показників обумовлює потребу у формалізованій автоматизації.

Окремою особливістю предметної області є повторюваність аналітичних процедур. Щотижня і щомісяця керівник повертається до тих самих операцій: перевіряє виконання планових показників, аналізує відхилення від базового періоду, оцінює сильні й слабкі локації, а також вирішує, чи потребує окрема точка маркетингового або кадрового втручання. Якщо такі процедури щоразу виконуються вручну, зростають витрати часу та знижується відтворюваність висновків.

Предметна область також характеризується високою залежністю від якості первинних даних. Дані про продажі працівників, категорії товарів, чеки й структуру меню мають бути зведені до єдиного вигляду. Лише після нормалізації й агрегації ці дані стають придатними для побудови КРІ та порівняльної аналітики. Отже, розробка прикладної системи в цій області передбачає не лише відображення чисел, а й організацію повного циклу обробки операційних даних.

Таким чином, мережа кав'ярень як об'єкт дослідження поєднує формалізовані показники, повторювані аналітичні задачі, наявність структурованих даних і потребу у швидкому управлінському реагуванні. Це робить її придатною для застосування методів системного аналізу, модульного проєктування та автоматизованої звітності [7; 13; 14].

Для керівника мережі особливого значення набуває оперативний доступ до достовірної статистики. Умовно всі показники можна поділити на три групи: індивідуальні показники працівників, показники окремої локації та порівняльні показники між локаціями й періодами. До першої групи належать денний виторг, кількість чеків, середній чек, кількість змін і частка продажів певних категорій. Друга група охоплює загальний виторг точки, загальну кількість чеків, середні денні показники та структуру виторгу. Третя група дає змогу оцінити динаміку тиждень до тижня, місяць до місяця і рік до року.

У реальних умовах керівник мережі кав'ярень часто формує такі звіти вручну на основі даних з POS-системи або проміжних Excel-файлів. Це означає необхідність щоразу повторювати однаковий цикл дій: обрати період, отримати дані, очистити їх, звести в таблиці, виконати порівняння, побудувати рейтинг і сформулювати висновки. Такий підхід є трудомістким і погано масштабується зі збільшенням кількості локацій.

Отже, предметна область характеризується високою повторюваністю аналітичних процедур, наявністю формалізованих показників і потребою у швидкому отриманні результатів. Саме тому вона є придатною для автоматизації засобами програмного забезпечення.

## **1.2 Аналіз існуючих підходів до формування аналітичної звітності**

Найпростішим підходом до формування аналітичної звітності є ручне опрацювання даних. Користувач експортує статистику з облікової системи, після чого переносить її до електронних таблиць, виконує сортування, фільтрування, обчислення та візуалізацію. Перевагою такого підходу є низький поріг входу, однак він має суттєві недоліки: високі часові витрати, залежність від людського фактора, низьку відтворюваність і обмежену масштабованість.

Другий підхід полягає у використанні типових звітів, що вже вбудовані в POS-систему. Такі звіти дозволяють швидко отримати окремі показники, але часто не враховують специфіку конкретного бізнесу, не підтримують складні правила порівняння між періодами й не формують індивідуалізовані висновки. Для керівника мережі цього зазвичай недостатньо, оскільки потрібне поєднання декількох типів аналітики в одному файлі.

Третій підхід передбачає розробку спеціалізованих програмних засобів, які інтегруються з джерелом даних через API. У цьому випадку можна гнучко задавати періоди, розраховувати власні KPI, формувати специфічні звіти для

компанії та застосовувати формальні правила оцінювання. Саме цей підхід є найбільш доцільним для задачі, розглянутої у кваліфікаційній роботі.

Порівняльну характеристику основних підходів до формування звітності наведено в таблиці 1.1.

Для коректного порівняння підходів до формування звітності доцільно використовувати декілька критеріїв: трудомісткість, повторюваність, адаптивність до змін бізнес-процесу, вимоги до кваліфікації користувача та придатність до масштабування на декілька локацій. Ручний підхід у середовищі Excel демонструє прийнятну гнучкість лише доти, доки йдеться про обмежену кількість точок і прості одноразові вибірки. Зі збільшенням кількості локацій та періодів виникає ефект накопичення рутинних операцій: одна й та сама логіка сортування, фільтрації та порівняння повторюється багаторазово.

Вбудовані звіти POS-систем дають змогу швидко отримати окремі зрізи даних, але їх типовою проблемою є слабка адаптивність до специфіки управлінського запиту. Наприклад, керівнику мережі потрібне не лише відображення виторгу або кількості чеків, а й одночасне поєднання показників працівників, локацій та динаміки між періодами в одному файлі. Крім того, бізнесу часто потрібні власні правила інтерпретації відхилень, які не реалізовано у стандартних шаблонах системи.

Спеціалізований застосунок на базі API, навпаки, дозволяє будувати процес аналізу від задачі керівника, а не від обмежень готового інтерфейсу. Такий підхід відповідає сучасному баченню програмної інженерії, де окрема прикладна система створюється навколо конкретного сценарію використання, має чітко визначені модулі та може поступово розширюватися новими функціями без повної перебудови архітектури [13; 14; 16].

У контексті бізнес-аналітики перевага спеціалізованого інструмента полягає ще й у тому, що він дає можливість інтегрувати аналітичну логіку безпосередньо у процес формування звіту. Це означає, що користувач отримує не лише таблиці з даними, а й інтерпретовані сигнали: зростання, просідання, падіння трафіку, нестабільну базу порівняння тощо. Саме тому для мережі кав'ярень найбільш доцільним є застосування окремого програмного засобу, який поєднає отримання даних, обчислення KPI та rule-based аналіз [20; 21].

Таблиця 1.1

## Порівняльна характеристика підходів до формування звітності

| Підхід                            | Переваги                                         | Недоліки                                                 | Придатність для мережі |
|-----------------------------------|--------------------------------------------------|----------------------------------------------------------|------------------------|
| Ручне опрацювання Excel           | Гнучкість; доступність                           | Високі витрати часу; помилки; відсутність стандартизації | Низька                 |
| Типові вбудовані звіти POS        | Швидкий запуск; мінімум налаштувань              | Обмежена кастомізація; слабка порівняльна аналітика      | Середня                |
| Спеціалізований застосунок на API | Автоматизація; власні KPI; єдина структура звіту | Потребує розробки й супроводу                            | Висока                 |

Наведене порівняння показує, що найбільш доцільним є створення спеціалізованого програмного засобу, який поєднає гнучкість, повторюваність і автоматичне формування висновків.



### 1.3 Аналіз можливостей POS-системи Poster як джерела даних

Poster є хмарною POS-системою, орієнтованою на заклади громадського харчування. Система надає API, за допомогою якого можна отримувати інформацію про продажі по працівниках, категоріях товарів, локаціях, меню та інші дані, потрібні для аналітики [6]. У розробленому програмному засобі використовуються чотири ключові джерела даних: продажі працівників, продажі по категоріях, зведена статистика локації та дерево категорій меню.

З технічної точки зору API Poster є зручним джерелом даних для автоматизації звітності з трьох причин. По-перше, доступ до даних виконується за чітко визначеними HTTP-запитами. По-друге, відповіді повертаються в структурованому форматі, придатному до машинної обробки. По-третє, дані можна отримувати окремо для кожної кав'ярні, використовуючи токени доступу.

У таблиці 1.2 наведено основні API-ендпоїнти, які були використані в роботі.

Практична робота з Poster API у межах цього дослідження показала, що для побудови коректної аналітики недостатньо лише отримати JSON-відповідь. Важливим етапом є нормалізація формату дат, числових полів, назв категорій та ідентифікаторів користувачів. Наприклад, суми можуть повертатися в мінорних одиницях, а окремі поля потребують переведення у формат, придатний для подальших обчислень у Python. Саме на цьому етапі формується основа для достовірних аналітичних висновків [7; 9–11].

Особливої уваги потребує використання токенів доступу для різних локацій. У роботі було обрано підхід, за якого для кожної кав'ярні зберігається окремий токен, а самі запити інкапсульовані в окремому модулі. Це дозволяє відокремити логіку мережевої взаємодії від бізнес-логіки застосунку та спрощує подальший супровід системи. Подібна ізоляція зовнішніх

залежностей відповідає практиці побудови підтримуваних програмних систем [13; 23].

Оскільки в дослідженні використовуються декілька API-ендпоїнтів, дані мають різну семантику: продажі працівників описують мікрорівень роботи мережі, зведена статистика локації дає макрорівень, а дерево меню і продажі по категоріях забезпечують контекст для розрахунку частки кави, їжі чи інших груп товарів. Комбінування цих джерел в одному звіті є важливою перевагою розробленого рішення, оскільки керівник отримує багаторівневий аналітичний погляд на діяльність мережі.

Таким чином, Poster API у межах роботи виконує роль не просто транспортного механізму для передачі даних, а фактично джерела структурованої бази операційної аналітики. Саме поєднання чотирьох ключових ендпоїнтів, їх нормалізація та подальша агрегація роблять можливим формування повноцінних weekly- і monthly-звітів.

Уточнюючи постановку задачі, слід підкреслити, що система має реалізувати не один, а декілька взаємопов'язаних рівнів функціональності. Перший рівень стосується доступу до даних і валідації введених користувачем параметрів. Другий рівень охоплює аналітичну обробку: обчислення похідних показників, побудову порівнянь і формування rule-based сигналів. Третій рівень відповідає за представлення результатів у вигляді придатного до використання Excel-файлу та взаємодію з користувачем через локальний графічний інтерфейс.

Крім функціональних вимог, для системи є важливими й нефункціональні вимоги: зрозумілість інтерфейсу, відтворюваність результатів, модульність коду, придатність до локального запуску на типовій Windows-конфігурації, а також можливість подальшого розширення без повної

переробки програми. Саме ці вимоги визначили вибір мови Python, модульного принципу організації проєкту та окремих бібліотек для API, Excel і GUI.

Постановка задачі також передбачає наявність перевірюваного результату. Тобто система має бути апробована не лише теоретично, а й на реальних даних мережі кав'ярень. Очікуваним результатом є сформовані файли CashBonusReport, WeeklyAnalytics і MonthlyAnalytics, а також підтвердження того, що система коректно обробляє негативні сценарії: незавершений період, помилковий формат дати, майбутні дати чи відсутність даних.

Отже, постановка задачі в цій роботі охоплює повний цикл: від аналізу предметної області та джерел даних до проєктування, реалізації, тестування й практичної оцінки застосовності системи. Саме така постановка відповідає логіці прикладної бакалаврської роботи за спеціальністю «Комп'ютерні науки» [1–3].

Таблиця 1.2

## Ендпоїнти Poster API та їх призначення

| Ендпоїнт                | Призначення                                 | Результат обробки                          |
|-------------------------|---------------------------------------------|--------------------------------------------|
| dash.getWaitersSales    | Отримання продажів по працівниках за період | Рейтинг денних кас, бонусна статистика     |
| dash.getCategoriesSales | Отримання виторгу по категоріях             | Частка кави та іншої продукції, частка їжі |
| dash.getSpotsSales      | Зведена статистика локації                  | Виторг, чеки, середній чек                 |
| menu.getCategories      | Отримання дерева категорій меню             | Класифікація категорій для аналітики       |

Отримані через API дані після нормалізації використовуються як вхід для алгоритмів агрегації, порівняння та rule-based інтерпретації. Таким чином

Poster виступає не лише обліковою системою, а й базою даних для побудови зовнішнього аналітичного рішення.

#### **1.4 Постановка задачі розробки інтелектуальної системи**

На основі проведеного аналізу можна сформулювати постановку задачі. Необхідно створити програмний засіб, який для мережі кав'ярень автоматично отримує дані з Poster API, перевіряє коректність вибраного користувачем періоду, виконує обчислення показників ефективності, будує Excel-звіти за декількома сценаріями та формує короткі аналітичні рекомендації.

У функціональному аспекті система має забезпечувати три основні сценарії використання. Перший сценарій – ручне формування звіту за довільний період із побудовою рейтингу денних кас та бонусної системи. Другий сценарій – формування weekly-аналітики для завершеного тижня з порівнянням до попереднього тижня. Третій сценарій – формування monthly-аналітики для завершеного календарного місяця з помісячним і річним порівнянням.

Інтелектуальна складова системи полягає не у використанні машинного навчання, а в автоматизованому застосуванні правил інтерпретації показників. Система класифікує динаміку виторгу, трафіку та середнього чека, визначає характер зміни ситуації на локації та формує рекомендацію, орієнтовану на управлінське рішення.

Отже, задача кваліфікаційної роботи полягає у проєктуванні та реалізації локального desktop-застосунку, який автоматизує типові аналітичні процедури мережі кав'ярень і забезпечує зручне отримання структурованих результатів у форматі Excel.

## **Висновки до розділу 1**

У першому розділі було проведено комплексне дослідження предметної області, що дозволило визначити мережу кав'ярень як розподілену систему, де швидкість та точність управлінської аналітики безпосередньо впливають на ефективність бізнесу. Аналіз існуючих підходів до формування звітності виявив недоліки ручної обробки даних та обмеженість стандартних звітів POS-систем, що підтвердило необхідність розробки автоматизованого інструменту на базі API. Дослідження можливостей системи Poster показало, що її програмний інтерфейс є надійним джерелом структурованої інформації, яка після нормалізації та агрегації стає основою для розрахунку ключових показників ефективності. На основі отриманих результатів було сформульовано постановку задачі, яка передбачає створення інтелектуальної системи у вигляді локального застосунку на Python для автоматизації збору даних, проведення порівняльного аналізу періодів та генерації управлінських рекомендацій на основі правил. Таким чином, результати першого розділу створили необхідне аналітичне підґрунтя для подальшого проєктування архітектури та алгоритмів роботи системи.

## РОЗДІЛ 2

### ПРОЄКТУВАННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ АНАЛІЗУ ТА ФОРМУВАННЯ ЗВІТНОСТІ

#### 2.1. Функціональні вимоги до системи

Для ручного сценарію важливою є не лише перевірка правильності введення періоду, а й прозорість самого результату. Користувач має чітко розуміти, які саме аркуші буде сформовано, які показники потрапляють до файлу і в чому полягає відмінність між рейтингом денних кас та бонусною статистикою. Тому у вимогах до інтерфейсу окремо враховано наявність підказок і опису складу вихідного документа.

Weekly- та monthly-сценарії мають додаткові вимоги, пов'язані з логікою завершеності періодів. Для weekly-аналітики система повинна блокувати вибір неповного тижня або інтервалу, що не відповідає межам з понеділка по неділю. Для monthly-аналітики необхідно контролювати, щоб період починався з першого і завершувався останнім днем календарного місяця. Реалізація таких перевірок усуває неоднозначність інтерпретації результатів і підвищує довіру до звіту.

Важливою вимогою є також стабільність роботи з зовнішнім API. Система має коректно повідомляти користувача про помилки мережевої взаємодії, таймаути, відсутність даних чи інші збої, не допускаючи створення некоректного файлу. У прикладних інформаційних системах така поведінка є ознакою надійності та інженерної зрілості рішення [7; 13; 14].

Таким чином, функціональні вимоги в межах цієї роботи описують не лише набір операцій, які виконує програма, а й очікувану поведінку системи за нормальних та аномальних умов. Саме це забезпечує практичну придатність розробки до повсякденного використання.

Функціональні вимоги до системи визначалися виходячи з реальних сценаріїв використання керівником мережі кав'ярень. Система повинна працювати локально на персональному комп'ютері, мати зрозумілий графічний інтерфейс, забезпечувати введення або вибір періоду, перевіряти коректність дат, взаємодіяти з Poster API і створювати підсумкові Excel-файли.

Для структурування вимог доцільно виділити три групи: вимоги до ручного звіту, вимоги до weekly-аналітики та вимоги до monthly-аналітики. Окремо слід врахувати вимоги до надійності обробки помилок, локальної роботи на Windows і зрозумілості інтерфейсу.

З архітектурної точки зору центральним елементом системи є модуль `main.py`, який координує виконання сценаріїв та об'єднує результати роботи допоміжних модулів. Поруч із ним функціонують спеціалізовані компоненти: модуль конфігурації, модуль роботи з API, модуль календарної логіки, модуль побудови звітів та графічні компоненти desktop-застосунку. Такий розподіл обов'язків мінімізує зв'язність між частинами програми та полегшує тестування.

Взаємодія модулів у процесі побудови звіту має направлений характер. Графічний інтерфейс формує запит користувача, `backend` запускає відповідний сценарій, після чого центральний модуль отримує від `scheduler.py` підтвердження коректності періоду, звертається до `poster_api.py` для завантаження даних, виконує агрегацію та передає підготовлені структури до `report_builder.py`. На виході формується готовий XLSX-файл у каталозі `reports`. Охайне відображення цієї взаємодії подано на рисунку 2.1.

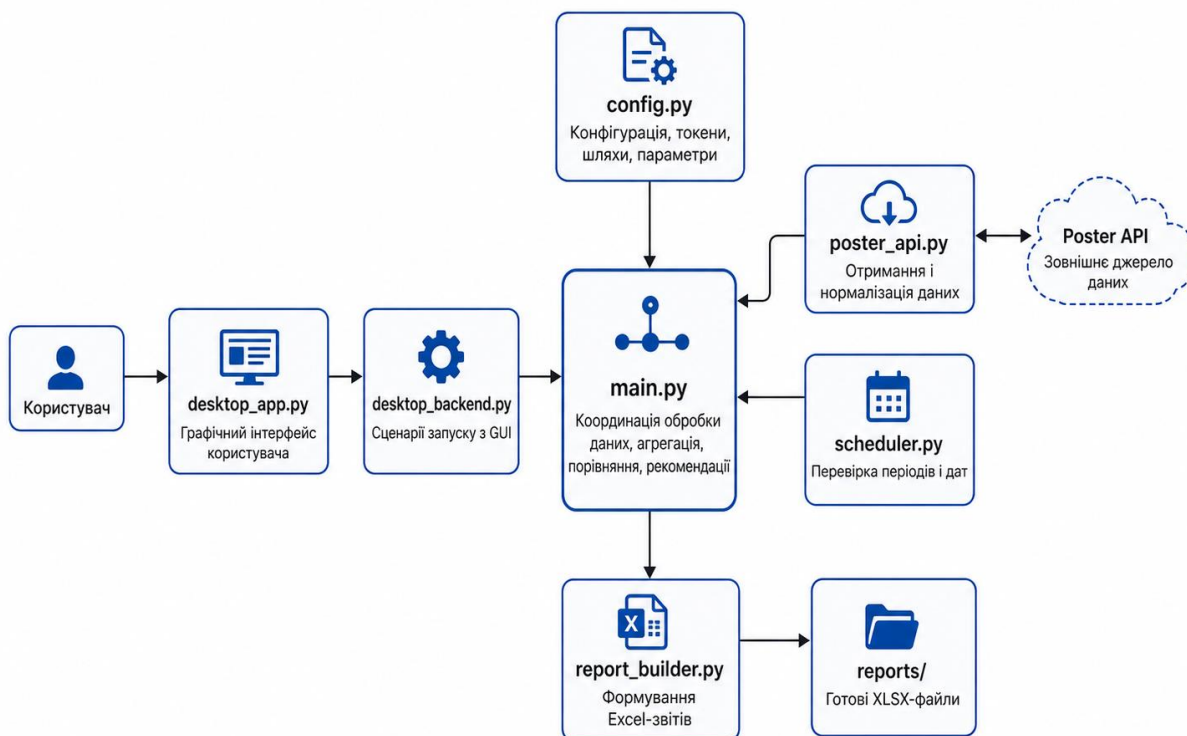


Рисунок 2.1. Архітектурна схема програмного засобу

На відміну від архітектурної схеми, модель обробки даних описує не структуру файлів, а послідовність логічних етапів. У межах цього дослідження важливо було відокремити етапи введення й перевірки періоду від етапів мережевої взаємодії, нормалізації, агрегації та інтерпретації показників. Таке відокремлення дає змогу локалізувати помилки та робить процес побудови звіту відтворюваним.

Після надходження сирих даних із Poster API система виконує нормалізацію і лише потім переходить до побудови похідних метрик. Саме на цій стадії формуються показники, що безпосередньо використовуються в бізнес-аналітиці: виторг, кількість чеків, середній чек, частка категорій, середня каса та інші KPI. Далі ці показники порівнюються з базовим періодом, а їх динаміка інтерпретується через набір правил. Узагальнений конвеєр обробки даних наведено на рисунку 2.2.





Рисунок 2.2 .Узагальнена схема обробки даних у системі

Таблиця 2.1

## Основні функціональні вимоги до системи

| Група вимог       | Зміст вимоги                                                                 | Результат                         |
|-------------------|------------------------------------------------------------------------------|-----------------------------------|
| Ручний звіт       | Формування файлу за довільний період;<br>перевірка наявності даних           | Файл CashBonusReport              |
| Weekly-аналітика  | Робота лише з завершеним тижнем;<br>порівняння з попереднім тижнем           | Файл WeeklyAnalytics              |
| Monthly-аналітика | Робота лише з завершеним місяцем;<br>порівняння з попереднім місяцем і роком | Файл MonthlyAnalytics             |
| Інтерфейс         | Окремі вкладки, повідомлення про помилки,<br>відкриття папки зі звітами      | Зручна локальна робота            |
| Надійність        | Валідація дат, перевірка повноти періоду,<br>обробка помилок API             | Стабільність<br>формування звітів |

Сформульовані вимоги визначають логіку архітектури системи та набір модулів, що розглядаються далі.

## 2.2 Архітектура програмної системи та модель обробки даних

Програмна система реалізована за модульним принципом. Такий підхід дозволяє розділити відповідальність між окремими файлами, спростити супровід і зробити архітектуру зрозумілою для захисту кваліфікаційної роботи. Ядро системи складається з модулів `config.py`, `poster_api.py`, `main.py`, `scheduler.py`, `report_builder.py`, `desktop_backend.py` та `desktop_app.py`.

Для кращого формалізованого опису алгоритмів у цій роботі можна виділити спільний шаблон побудови звіту: перевірка вхідних параметрів, завантаження сирих даних, нормалізація, агрегація, порівняння, формування рекомендацій, побудова XLSX-файлу. Відмінності між режимами полягають переважно у визначенні базового періоду та наборі підсумкових аркушів.

Ручний сценарій орієнтований насамперед на оперативний рейтинг працівників і бонусну статистику. Weekly-сценарій доповнює підсумкові дані порівнянням із попереднім тижнем. Monthly-сценарій є найповнішим, оскільки окрім порівняння з попереднім місяцем підтримує аналіз «рік до року», що є корисним для виявлення сезонних змін. Подібне нарощення аналітичної глибини від простого до складнішого сценарію робить систему зручною для користувача та пояснюваною під час захисту роботи.

Модуль `config.py` містить константи системи: адреси API-ендпоінтів, перелік локацій і токенів доступу, шляхи до папки зі звітами та параметри HTTP-запитів. Модуль `poster_api.py` ізолює всю взаємодію з Poster API та повертає дані у нормалізованій формі. Модуль `scheduler.py` відповідає за обчислення календарних періодів і перевірку того, чи є вибраний інтервал завершеним тижнем або завершеним місяцем.

Модуль `main.py` містить основні функції побудови категорій, агрегації даних, порівняння показників та генерації rule-based рекомендацій. Модуль

report\_builder.py відповідає за створення та форматування Excel-файлів. Desktop\_backend.py інкапсулює backend-функції для графічного інтерфейсу, а desktop\_app.py реалізує сам інтерфейс користувача на tkinter.

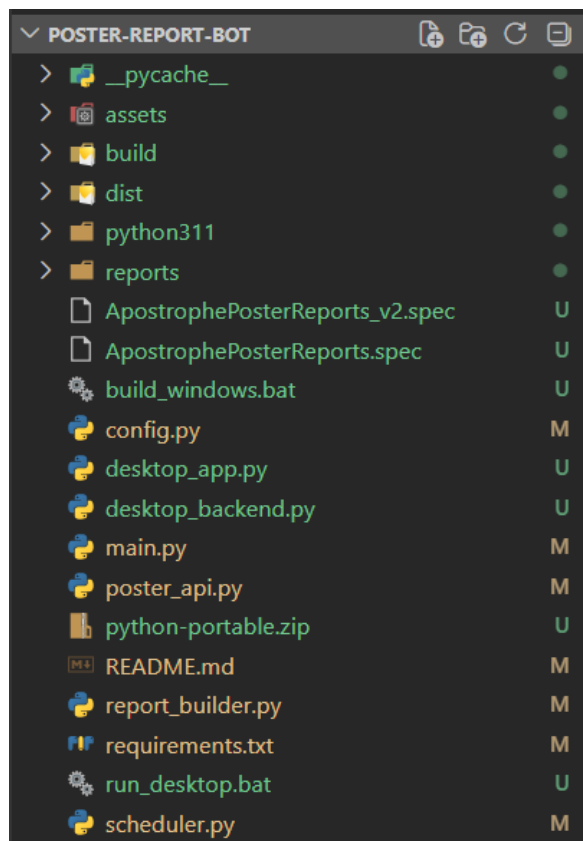
Rule-based підхід обрано з кількох причин. По-перше, він не потребує значних обсягів навчальних вибірок і тому є доцільним для малого прикладного рішення. По-друге, така логіка легко пояснюється керівнику та членам екзаменаційної комісії, оскільки кожен сигнал породжується конкретним поєднанням змін показників. По-третє, правила можуть поступово змінюватися або розширюватися без перебудови всієї системи.

У межах дослідження правила охоплюють як типові ситуації росту та спаду, так і спеціальні випадки: малу базу порівняння, аномальні відхилення, нестабільні відсотки. Така комбінація кількісних та якісних перевірок дозволяє уникнути хибних висновків у тих випадках, коли різке відсоткове зростання зумовлене занадто малими початковими значеннями. Отже, інтелектуальна складова тут полягає саме у пояснюваному автоматизованому аналізі, а не у використанні «чорних скриньок» машинного навчання [20; 21]. Загальна архітектура інтерфейсу дозволяє користувачеві не блокувати роботу вікна під час генерації звітів, оскільки обробка даних відбувається в окремому потоці. Така структура дозволяє відокремити логіку мережевої взаємодії від бізнес-логіки застосунку та представлення даних, що відповідає сучасним практикам побудови підтримуваних програмних систем

Загальний вигляд структури програми відображений на рис. 2.3. Розподіл відповідальності між модулями наведено в таблиці 2.2.

Рисунок 2.3.  
програмних файлів

Таблиця 2.2  
Розподіл  
відповідальності між  
системи



Структура  
проєкту

модулями

| Модуль             | Основне призначення                                             |
|--------------------|-----------------------------------------------------------------|
| config.py          | Конфігурація системи, токени, шляхи, таймаути                   |
| poster_api.py      | HTTP-запити до Poster API та нормалізація відповідей            |
| scheduler.py       | Обчислення тижневих і місячних періодів, перевірка завершеності |
| main.py            | Агрегація даних, порівняння, формування рекомендацій            |
| report_builder.py  | Побудова та оформлення Excel-звітів                             |
| desktop_backend.py | Backend-сценарії для GUI та валідація введення                  |
| desktop_app.py     | Графічний інтерфейс користувача                                 |

Обрана архітектура відповідає принципу розподілу відповідальності та спрощує як подальшу підтримку програми, так і опис її структури в дипломній роботі. Модель обробки даних у системі включає декілька послідовних етапів. На першому етапі користувач через інтерфейс задає тип звіту та період. На другому етапі backend виконує валідацію введених дат і визначає, чи

відповідає період правилам конкретного сценарію. На третьому етапі модуль `poster_api.py` надсилає HTTP-запити до Poster API та отримує сирі JSON-відповіді. На четвертому етапі дані нормалізуються і перетворюються на уніфіковані Python-структури.

Далі система виконує побудову похідних показників. Для бариста визначається виторг, кількість змін, середня каса та частка реалізації їжі. Для локацій обчислюються загальний виторг, кількість чеків, середні денні значення, середній чек, сума й частка продажів кави та інших категорій. На етапі порівняння формуються дельти між поточним і попереднім періодом.

Окреме значення має класифікація категорій товарів. Для цього програма завантажує дерево категорій меню і формує множини `category_id` для кави, напоїв та нехарчових позицій. Такий підхід дозволяє гнучко визначати частку кави й іншої продукції та використовувати її в подальших аналітичних висновках.

Узагальнена модель обробки даних має лінійно-модульний характер: інтерфейс → валідація → API → нормалізація → агрегація → порівняння → генерація звіту.

## **2.3 Алгоритми формування аналітичних звітів та оцінювання ефективності**

Алгоритм формування ручного звіту для рейтингу кас та бонусної системи починається з перевірки введеного користувачем періоду. Після цього система отримує щоденні продажі по працівниках і відбирає для кожної локації десять найкращих денних результатів за виторгом. Паралельно збирається

бонусна статистика: виторг працівника, кількість змін, середня каса та частка реалізації їжі в сукупному виторгу.

Алгоритм weekly-аналітики використовує завершений тиждень з понеділка по неділю. Для кожної локації обчислюються зведені показники за поточний тиждень, після чого аналогічні показники отримуються для попереднього тижня. На основі двох наборів даних формується таблиця порівняння та набір rule-based рекомендацій.

Алгоритм monthly-аналітики працює аналогічно, але використовує завершений календарний місяць. Поточний місяць порівнюється з попереднім місяцем і, за наявності коректної бази, з аналогічним місяцем попереднього року. Це дозволяє оцінити як короткострокову, так і річну динаміку.

Формування рекомендацій виконується на основі матриці змін двох ключових показників: кількості чеків і середнього чека. Напрями змін визначаються як up, flat або down залежно від порогу  $\pm 2$  %. Поєднання напрямів у матриці  $3 \times 3$  породжує сигнали типу «Сильне зростання», «Падіння трафіку», «Трафік росте, але чек слабшає» тощо. Завдяки такій схемі (див. таблицю 2.3) аналітичні висновки формуються автоматично, але залишаються прозорими та пояснюваними для користувача.

Інтелектуальна складова системи базується на наборі формальних правил, реалізованих у кодї. По-перше, система обчислює абсолютні та відносні зміни показників між поточним і попереднім періодами. По-друге, використовується правило малої бази: якщо попередній виторг менший за 5000 грн, відсоткові висновки вважаються нестабільними. По-третє, для відхилень понад 300 % система формує попередження про аномальну динаміку.

Матриця rule-based сигналів для інтерпретації динаміки

| Напря́м зміни чеків | Напря́м зміни середнього че́ка | Сигнал                        |
|---------------------|--------------------------------|-------------------------------|
| up                  | up                             | Сильне зростання              |
| up                  | flat                           | Зростання за рахунок трафіку  |
| up                  | down                           | Трафік росте, але чек слабшає |
| down                | flat                           | Падіння трафіку               |
| down                | down                           | Системне просідання           |

Для weekly-звіту рекомендація будується на основі поточної та попередньої тижневої статистики. Для monthly-звіту в пояснення додатково інтегрується інформація про зміну виторгу рік до року, якщо така база порівняння є коректною. Це дозволяє враховувати не лише короткострокову динаміку, а й ширший контекст розвитку локації.

Важливо, що система не просто відображає цифри, а інтерпретує їх мовою управлінських сигналів. Наприклад, зростання чеків за стабільного середнього чека класифікується як зростання за рахунок трафіку, а одночасне зростання чеків і середнього чека – як сильне зростання. Такий підхід відповідає ознакам інтелектуального аналізу на основі правил і пояснюваної логіки.

Отже, інтелектуальність розробленого засобу проявляється в поєднанні автоматичного розрахунку КРІ, класифікації динаміки, перевірки якості бази порівняння та генерації рекомендацій, а не лише в побудові табличних звітів.

## Висновки до розділу 2

У другому розділі було спроектовано архітектуру та логіку функціонування інтелектуальної системи, що дозволило перетворити сформульовану постановку задачі у чітку технічну модель.

На основі аналізу потреб менеджменту мережі було визначено детальні функціональні вимоги, які охоплюють три ключові сценарії використання: формування оперативних звітів (рейтинги та бонуси), а також проведення поглибленої тижневої та місячної аналітики з обов'язковим автоматизованим контролем завершеності та коректності часових періодів.

Запропонована модульна архітектура системи, що базується на розподілі відповідальності між окремими компонентами для взаємодії з API, календарною логікою, агрегацією даних та графічним інтерфейсом, забезпечує високу надійність, мінімальну зв'язність коду та простоту подальшого супроводу програмного засобу.

Особливу увагу приділено моделі обробки даних, яка передбачає послідовну нормалізацію «сирих» відповідей Poster API, розрахунок складних KPI (зокрема частки їжі у виторгу чи середньої каси на зміну) та їх подальшу агрегацію для багатofакторного порівняльного аналізу.

Ключовим результатом розділу є розробка інтелектуальної складової на основі rule-based підходу, що використовує матрицю змін трафіку та середнього чека для автоматичного формування прозорих управлінських рекомендацій.

Такий підхід дозволяє системі не лише обчислювати суху статистику, а й самостійно інтерпретувати характер динаміки бізнес-показників, враховуючи при цьому обмеження малої бази порівняння та виявлення аномальних відхилень.



Отже, розроблені проектні рішення, алгоритми та моделі створюють цілісне та обґрунтоване підґрунтя для успішної програмної реалізації системи та її практичної апробації на реальних операційних показниках мережі кав'ярень.

## РОЗДІЛ 3

### РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАСОБУ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

#### 3.1 Засоби та технології реалізації

Розроблений програмний засіб реалізовано мовою Python 3, що забезпечує достатній рівень гнучкості, велику кількість готових бібліотек і зручність для побудови прототипів та прикладних desktop-застосунків [8; 17]. Для взаємодії з HTTP API використано бібліотеку requests [12], для формування Excel-файлів – openpyxl [12], а для графічного інтерфейсу – tkinter [12].

Завдяки використанню Python вдалося організувати чисту модульну структуру проєкту, у якій логіка отримання даних, агрегації показників, формування файлів і взаємодії з користувачем розділені між окремими компонентами. Така реалізація є достатньо простою для захисту бакалаврської роботи, але водночас демонструє коректні інженерні підходи до побудови прикладного програмного забезпечення.

У таблиці 3.1 наведено основні технології, використані під час реалізації системи.

Модуль poster\_api.py виконує роль обгортки над Poster API. У ньому реалізовано універсальну функцію \_request\_json, яка надсилає GET-запит, використовує таймаут з config.py та повертає масив response з JSON-відповіді.

Це дозволяє уникнути дублювання HTTP-логіки в інших частинах програми:

```
def _request_json(url: str, params: dict, timeout: int = 30) ->
list[dict]:
    response = requests.get(url, params=params, timeout=timeout)
    response.raise_for_status()
```

```
payload = response.json()
return payload.get("response", [])
```

Таблиця 3.1

## Технології, використані під час реалізації системи

| Технологія         | Призначення                | Причина вибору                             |
|--------------------|----------------------------|--------------------------------------------|
| Python 3           | Основна мова реалізації    | Простота, читабельність, велика екосистема |
| requests           | Робота з Poster API        | Зручна взаємодія з HTTP-запитами           |
| openpyxl           | Формування XLSX-файлів     | Гнучке створення аркушів і форматування    |
| tkinter            | Desktop-інтерфейс          | Входить до стандартної бібліотеки Python   |
| ThreadPoolExecutor | Паралельне отримання даних | Прискорення завантаження по локаціях       |

Для роботи з даними працівників реалізовано функцію `fetch_waiters_sales_for_period`, яка отримує дані за обраний період і повертає список нормалізованих словників із полями `location`, `weekday`, `date`, `user_id`, `waiter_name`, `revenue`, `clients` та `middle_invoice`. Аналогічно функції `fetch_categories_sales`, `fetch_menu_categories` та `fetch_spot_sales` повертають дані для категорій продажів, дерева меню і зведеної статистики локації.

```
def fetch_waiters_sales_for_period(token: str, date_from: date,
date_to: date) -> list[dict]:
    params = {
        "token": token,
        "dateFrom": date_from.isoformat(),
        "dateTo": date_to.isoformat(),
    }

    raw_rows = _request_json(POSTER_WAITERS_SALES_URL, params)
    normalized_rows = []

    for row in raw_rows:
        normalized_rows.append({
            "location": row.get("spot", "Невідомо"),
            "date": row.get("date", ""),
```

```

        "user_id": row.get("user_id"),
        "waiter_name": row.get("name", "Невідомо"),
        "revenue": float(row.get("revenue", 0)) / 100,
        "clients": int(row.get("clients", 0)),
        "middle_invoice": float(row.get("middle_invoice", 0))
/ 100,
    })
    return normalized_rows

```

Особливістю реалізації є попередня нормалізація дат і числових значень. Зокрема, суми, що повертаються API в мінорних одиницях, переводяться у гривні діленням на 100. Дати приводяться до єдиного формату ISO, а поля з можливими відсутніми значеннями мають безпечні значення за замовчуванням. Це спрощує подальшу обробку даних.

Для прискорення завантаження статистики по локаціях і днях у ряді сценаріїв використано ThreadPoolExecutor. Це дозволяє виконувати незалежні запити паралельно та зменшувати загальний час побудови звіту.

### 3.2 Реалізація модуля формування Excel-звітів та застосунку

Модуль report\_builder.py відповідає за побудову Excel-книг у форматі XLSX. Для цього використано бібліотеку openpyxl, що дозволяє створювати аркуші, об'єднувати комірки, задавати ширину стовпців, вирівнювання, рамки та стилі заливки. У програмі реалізовано окремі функції для побудови ручного звіту, weekly-аналітики та monthly-аналітики.

Ручний звіт формує два аркуші: «Топ-10 кас» і «Бонусна система». На першому аркуші для кожної локації виводяться десять найкращих денних результатів працівників із зазначенням дати, виторгу, кількості чеків і середнього чека. На другому аркуші для кожної локації виводиться бонусна статистика працівників із такими показниками, як частка їжі, середня каса, кількість змін, загальний виторг і сума реалізації їжі.

Weekly-аналітика і monthly-аналітика будуються за схожим принципом. На аркуші «Звіти по кав'ярнях» виводяться основні підсумкові показники по кожній локації. На аркушах «Порівняння тижнів», «Порівняння місяців» та «Рік до року» розміщуються таблиці порівняння поточного періоду з базовим. Окремий аркуш «Рекомендації» містить автоматично згенеровані висновки.

Таким чином, Excel-файл виступає не лише контейнером даних, а й завершеним інструментом представлення результатів аналізу користувачу, що підтверджують рисунки 3.1 і 3.2.

| Топ-10 денних кас по локаціях: 2026-04-01 - 2026-04-30 |            |              |          |                 |              |
|--------------------------------------------------------|------------|--------------|----------|-----------------|--------------|
| Apostrophe 1                                           |            |              |          |                 |              |
| Місце                                                  | Дата       | Працівник    | Виторг   | Кількість чеків | Середній чек |
| 1                                                      | 2026-04-01 | Крістіна     | 17450.75 | 126             | 138.50       |
| 2                                                      | 2026-04-07 | Анастасія П. | 17133.75 | 102             | 167.98       |
| 3                                                      | 2026-04-06 | Анастасія П. | 16873.53 | 119             | 141.79       |
| 4                                                      | 2026-04-09 | Крістіна     | 15635.50 | 109             | 143.44       |
| 5                                                      | 2026-04-16 | Крістіна     | 15596.75 | 113             | 138.02       |
| 6                                                      | 2026-04-04 | Крістіна     | 15522.75 | 92              | 168.73       |
| 7                                                      | 2026-04-02 | Анастасія П. | 15271.17 | 107             | 142.72       |
| 8                                                      | 2026-04-30 | Анастасія П. | 15244.00 | 122             | 124.95       |
| 9                                                      | 2026-04-15 | Анастасія П. | 15189.63 | 100             | 151.90       |
| 10                                                     | 2026-04-17 | Анастасія П. | 15137.00 | 104             | 145.55       |

Рисунок 3.1. Приклад рейтингу кас за квітень 2026

| Бонусна статистика: 2026-04-01 - 2026-04-30 |        |              |       |           |           |
|---------------------------------------------|--------|--------------|-------|-----------|-----------|
| Apostrophe 1                                |        |              |       |           |           |
| Бариста                                     | Їжа, % | Середня каса | Зміни | Виторг    | Їжа, сума |
| Анастасія П.                                | 25.24% | 13906.00     | 15    | 208590.05 | 52651.41  |
| Крістіна                                    | 26.50% | 13131.32     | 14    | 183838.52 | 48709.49  |

Рисунок 3.2. Приклад бонусної статистики за квітень 2026

### 3.3 Реалізація desktop-застосунку

Графічний інтерфейс реалізовано в модулі `desktop_app.py` на основі `tkinter`. Вікно програми має вкладкову структуру з трьома сценаріями: «Рейтинг кас і бонуси», «Weekly-аналітика» та «Monthly-аналітика». Для кожного сценарію передбачено власну форму введення дат і окремі пояснення щодо того, який склад звіту буде сформовано.

```
root = tk.Tk()
root.title("Poster Report Bot")
root.geometry("900x600")

notebook = ttk.Notebook(root)
notebook.pack(fill="both", expand=True)

manual_tab = ttk.Frame(notebook)
weekly_tab = ttk.Frame(notebook)
monthly_tab = ttk.Frame(notebook)

notebook.add(manual_tab, text="Рейтинг кас і бонуси")
notebook.add(weekly_tab, text="Weekly-аналітика")
notebook.add(monthly_tab, text="Monthly-аналітика")
```

Desktop-застосунок використовує backend-функції з `desktop_backend.py`, що запобігає дублюванню бізнес-логіки. Кнопка запуску для кожного сценарію спочатку виконує валідацію періоду, після чого в окремому потоці запускає побудову файлу. Такий підхід не блокує інтерфейс і покращує користувацький досвід.

У разі помилок користувач отримує повідомлення з поясненням, наприклад, про некоректний формат дати, спробу вибору незавершеного тижня або місяця чи відсутність даних за обраний період. Після успішного формування програма показує шлях до створеного файлу і дозволяє відкрити папку зі звітами.

З точки зору користувача це перетворює програму з набору Python-скриптів на повноцінний прикладний desktop-інструмент, придатний для реального щоденного використання.

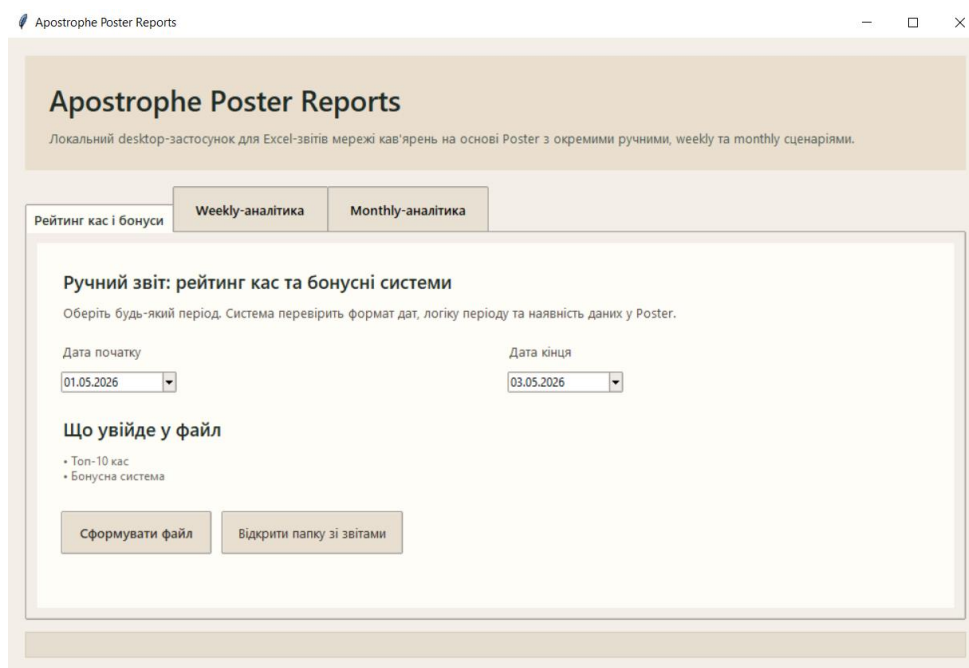


Рисунок 3.3. Головне вікно desktop-застосунку (рейтинг кас)

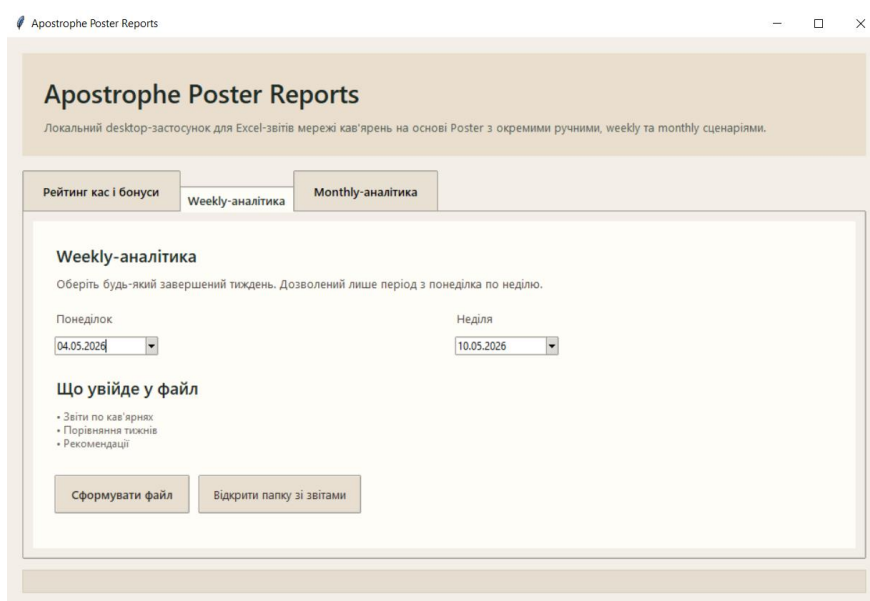


Рисунок 3.4. Головне вікно desktop-застосунку (weekly-аналітика)

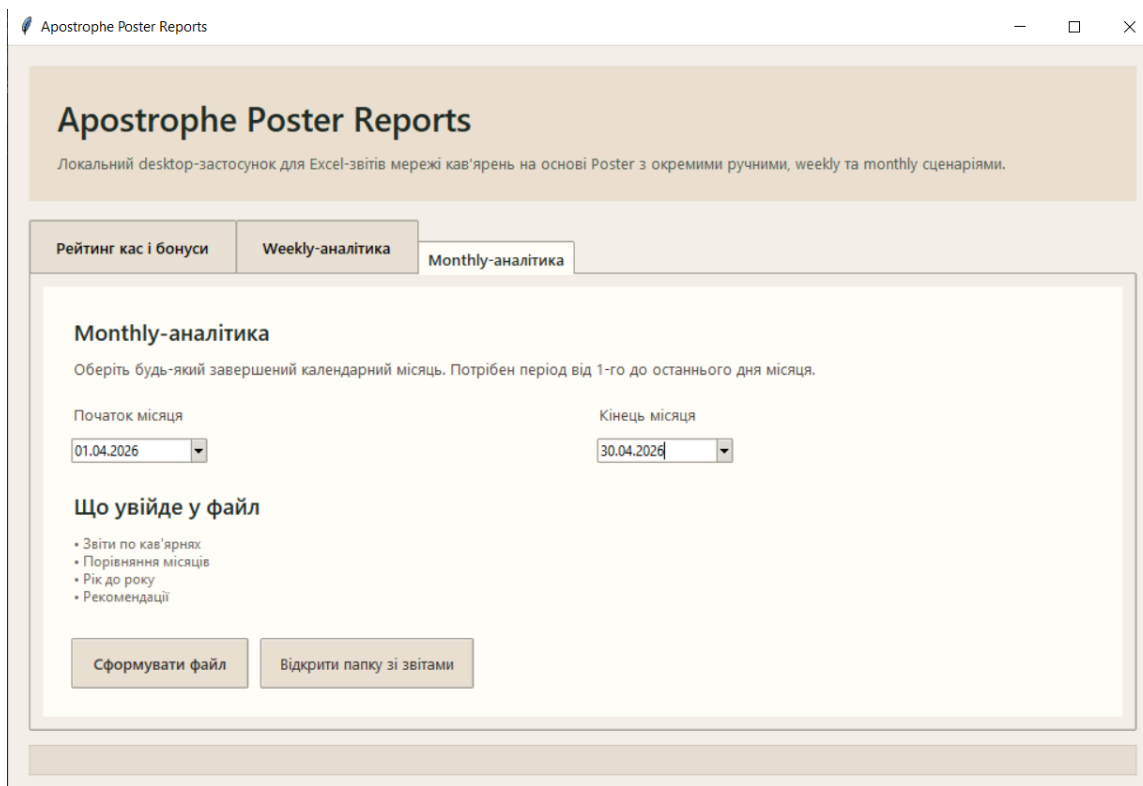


Рисунок 3.5. Головне вікно desktop-застосунку (monthly-аналітика)

### 3.4 Тестування та аналіз результатів роботи системи

Тестування програмного засобу виконувалося на реальних даних мережі кав'ярень із використанням сформованих прикладів файлів. Було перевірено три ключові сценарії: ручний звіт за довільний період, weekly-аналітика за завершений тиждень і monthly-аналітика за завершений місяць. Крім того, окремо перевірялися негативні сценарії: введення некоректних дат, спроба побудови звіту за майбутній або незавершений період, а також відсутність даних.

Результати тестування підтвердили коректність формування файлів. Зокрема, за період 01.03.2026–31.03.2026 у monthly-звіті зафіксовано, що найбільший виторг мала локація Apostrophe 2 – 495296,39 грн, тоді як для Apostrophe 1 цей показник становив 446666,30 грн. Для weekly-звіту за



13.04.2026–19.04.2026 система сформувала порівняльний лист, де для Apostrophe 2 було виявлено сигнал «Падіння трафіку», а для локацій Apostrophe 3, Apostrophe 4 та Apostrophe 5 – сигнал «Сильне зростання».

Важливою перевагою розробки є те, що користувач отримує не лише набір чисел, а завершений пакет аналітики. Наприклад, на аркуші рекомендацій для тижневого звіту система автоматично сформувала пояснення, що для Apostrophe 2 виторг змінився на –8,08 %, кількість чеків – на –6,59 %, а середній чек – на –1,60 %, і на цій підставі було запропоновано перевірити трафік та локальний маркетинг.

Результати перевірки базових сценаріїв наведено в таблиці 3.2.

Таблиця 3.2

## Результати перевірки базових сценаріїв

| Сценарій               | Вхідні дані                              | Очікуваний результат                                 | Фактичний результат |
|------------------------|------------------------------------------|------------------------------------------------------|---------------------|
| Ручний звіт            | 01.04.2026–<br>14.04.2026                | Формування<br>CashBonusReport з двома<br>аркушами    | Виконано<br>успішно |
| Weekly-<br>аналітика   | 13.04.2026–<br>19.04.2026                | Формування<br>WeeklyAnalytics з трьома<br>аркушами   | Виконано<br>успішно |
| Monthly-<br>аналітика  | 01.03.2026–<br>31.03.2026                | Формування<br>MonthlyAnalytics з чотирма<br>аркушами | Виконано<br>успішно |
| Некоректний<br>тиждень | Неповний або не з<br>понеділка по неділю | Повідомлення про помилку                             | Виконано<br>успішно |
| Майбутній<br>період    | Дата кінця ><br>поточної дати            | Блокування формування<br>звіту                       | Виконано<br>успішно |

Узагальнюючи результати, можна стверджувати, що розроблена система забезпечує автоматизацію типових аналітичних задач мережі кав'ярень, скорочує ручну роботу та підвищує відтворюваність управлінської звітності.

Практична апробація на реальних даних підтверджує доцільність її використання.

### **Висновки до розділу 3**

У третьому розділі було виконано програмну реалізацію інтелектуальної системи на мові Python 3 з використанням спеціалізованих бібліотек requests, openpyxl та tkinter, що дозволило забезпечити повний цикл збору, обробки та візуалізації аналітичних даних. В межах розробки було організовано ефективну модульну взаємодію з Poster API, реалізовано компоненти для побудови багатосторінкових XLSX-файлів та створено графічний інтерфейс користувача з інтуїтивною вкладковою структурою. Наведені практичні приклади сформованих звітів підтвердили здатність системи коректно функціонувати у ручному, тижневому та місячному сценаріях аналітики, а також її здатність адекватно обробляти негативні сценарії та некоректні вхідні дані. Проведені тестування на реальних операційних показниках мережі кав'ярень продемонстрували високу точність розрахунків KPI та релевантність автоматично згенерованих управлінських рекомендацій, що базуються на rule-based підході. Таким чином, результати апробації повністю підтвердили працездатність програмного засобу та його практичну доцільність як інструменту оперативної аналітики для підтримки прийняття рішень у реальних бізнес-процесах.

## ВИСНОВКИ

У кваліфікаційній роботі розв'язано актуальну задачу автоматизації аналітичної звітності для мережі кав'ярень на основі даних POS-системи Poster.

У процесі виконання роботи проаналізовано предметну область, визначено ключові проблеми ручного формування звітів і обґрунтовано доцільність використання спеціалізованого програмного засобу для побудови управлінської аналітики.

Досліджено можливості Poster API як джерела структурованих даних. Показано, що на основі ендпоінтів продажів по працівниках, категоріях, локаціях і меню можна сформувати повний набір показників, необхідних для оцінювання ефективності працівників і роботи локацій.

Спроектовано модульну архітектуру програмної системи, яка містить конфігураційний модуль, модуль взаємодії з API, модуль календарної логіки, модуль побудови Excel-звітів, backend-функції та desktop-інтерфейс.

Розроблено алгоритми побудови рейтингу денних кас, бонусної статистики, weekly-аналітики та monthly-аналітики. Запропоновано rule-based підхід до інтерпретації змін, який базується на порівнянні трафіку і середнього чека та дозволяє автоматично формувати управлінські рекомендації.

Реалізовано програмний засіб у вигляді локального desktop-застосунку на Python. Система забезпечує перевірку коректності вибраного періоду, отримання даних із Poster API, обчислення показників, формування структурованих Excel-файлів і подання результатів у зручному для користувача вигляді.

Виконане тестування на реальних даних мережі кав'ярень підтвердило працездатність системи. Сформовані звіти дозволяють порівнювати локації

між собою, відстежувати динаміку між періодами і швидко виявляти проблемні або успішні точки.

Практичне значення роботи полягає у скороченні часу на підготовку звітності, стандартизації аналітичних процедур, зменшенні впливу людського фактора та підвищенні якості управлінських рішень. Отже, поставлену мету досягнуто, а завдання кваліфікаційної роботи виконано.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Машкіна, І. В., Абрамов, В. О., Носенко, Т. І., Іваніченко, Є. В., & Яскевич, В. О. (2024). Навчально-методичний посібник до виконання і захисту кваліфікаційної роботи бакалавра спеціальностей 122 Комп'ютерні науки для здобувачів вищої освіти денної форми навчання.
2. Стандарт вищої освіти України: перший (бакалаврський) рівень, галузь знань 12 «Інформаційні технології», спеціальність 122 «Комп'ютерні науки».
3. Освітньо-професійна програма 122.00.01 «Інформатика» Київського столичного університету імені Бориса Грінченка.
4. Положення про академічну доброчесність науково-педагогічних, наукових, педагогічних працівників та здобувачів вищої освіти Київського університету імені Бориса Грінченка.
5. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання.
6. ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання. Загальні вимоги та правила складання.
7. Poster API Documentation. Режим доступу: <https://dev.joinposter.com/docs/v3/start/index>.
8. Python Software Foundation. Python 3 Documentation. Режим доступу: <https://docs.python.org/3/>.
9. Requests: HTTP for Humans. Official Documentation. Режим доступу: <https://requests.readthedocs.io/>.
10. openpyxl Documentation. Режим доступу: <https://openpyxl.readthedocs.io/>.
11. Python tkinter Documentation. Режим доступу: <https://docs.python.org/3/library/tkinter.html>.

12. PyInstaller Manual. Режим доступа: <https://pyinstaller.org/>.
13. Pressman R. S., Maxim B. R. Software Engineering: A Practitioner's Approach. 9th ed. New York: McGraw-Hill, 2019.
14. Sommerville I. Software Engineering. 10th ed. Boston: Pearson, 2016.
15. Martin R. C. Clean Code: A Handbook of Agile Software Craftsmanship. Upper Saddle River: Prentice Hall, 2008.
16. Fowler M. Refactoring: Improving the Design of Existing Code. 2nd ed. Boston: Addison-Wesley, 2018.
17. McKinney W. Python for Data Analysis. 3rd ed. Sebastopol: O'Reilly Media, 2022.
18. Downey A. B. Think Python: How to Think Like a Computer Scientist. 2nd ed. Needham: Green Tea Press, 2015.
19. Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software. Boston: Addison-Wesley, 1994.
20. Sharda R., Delen D., Turban E. Business Intelligence, Analytics, and Data Science: A Managerial Perspective. 5th ed. Pearson, 2020.
21. Han J., Kamber M., Pei J. Data Mining: Concepts and Techniques. 4th ed. Morgan Kaufmann, 2022.
22. Fielding R. T. Architectural Styles and the Design of Network-based Software Architectures. Doctoral Dissertation. University of California, Irvine, 2000.
23. Kleppmann M. Designing Data-Intensive Applications. Sebastopol: O'Reilly Media, 2017.
24. Hunt A., Thomas D. The Pragmatic Programmer. 20th Anniversary ed. Addison-Wesley, 2019.
25. Martin R. C. Agile Principles, Patterns, and Practices in C#. Prentice Hall, 2006.

26. PEP 8 – Style Guide for Python Code. Режим доступу: <https://peps.python.org/pep-0008/>.
27. PEP 257 – Docstring Conventions. Режим доступу: <https://peps.python.org/pep-0257/>.
28. JSON: JavaScript Object Notation, ECMA-404. 2nd ed. Geneva: Ecma International, 2017.
29. TkDocs. Modern Tkinter for Busy Python Developers. Режим доступу: <https://tkdocs.com/>.
30. DСТУ ISO 5807:2016. Обробляння інформації. Символи та угоди щодо документації стосовно даних, програм та системних блоксхем, схем мережевих програм та схем системних ресурсів.
31. Крискун, І. М. (2026). Model for evaluating the effectiveness of it project management using artificial intelligence. Економіка. Менеджмент. Бізнес, (1), 4-11.
32. Тушич, А. М., Сторчак, К. П., & Бондарчук, А. П. (2019). Вимоги до інтелектуальних систем аналізу даних та їх класифікацій. Телекомунікаційні та інформаційні технології, (1), 31-36.
33. Глушак О., Семеняка С., Зінченко Н., Соломко В (2025) Оцінювання факторів стійкості систем інформаційної безпеки методами регресійного аналізу. Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка», 2025, 3(31), 333–345
34. Abramov, V., Astafieva, M., Boiko, M., Bodnenko, D., Bushma, A., Vember, V., ... & Yaskevych, V. (2021). Theoretical and practical aspects of the use of mathematical methods and information technology in education and science.

## ДОДАТКИ А

## Приклади сформованих Excel-звітів у MonthlyAnalytics

| Звіти по кав'ярнях: 2026-04-01 - 2026-04-30 |           |                 |                         |                                 |              |            |         |            |         |
|---------------------------------------------|-----------|-----------------|-------------------------|---------------------------------|--------------|------------|---------|------------|---------|
| Кав'ярня                                    | Виторг    | Кількість чеків | Середній виторг за день | Середня кількість чеків за день | Середній чек | Кава: сума | Кава: % | Інше: сума | Інше: % |
| Apostrophe 1                                | 398514.07 | 2821            | 13283.80                | 94.03                           | 141.27       | 295558.67  | 74.17   | 102955.40  | 25.83   |
| Apostrophe 2                                | 437236.60 | 3472            | 14574.55                | 115.73                          | 125.93       | 328561.01  | 75.14   | 108675.59  | 24.86   |
| Apostrophe 3                                | 289550.18 | 2744            | 9651.67                 | 91.47                           | 105.52       | 229116.36  | 79.13   | 60433.82   | 20.87   |
| Apostrophe 4                                | 269230.84 | 2415            | 8974.36                 | 80.50                           | 111.48       | 215305.36  | 79.97   | 53925.48   | 20.03   |
| Apostrophe 5                                | 201613.25 | 1679            | 6720.44                 | 55.97                           | 120.08       | 158983.37  | 78.86   | 42629.88   | 21.14   |

| Порівняння місяців: 2026-04-01 - 2026-04-30 |                 |                   |             |              |                |           |                       |                         |               |
|---------------------------------------------|-----------------|-------------------|-------------|--------------|----------------|-----------|-----------------------|-------------------------|---------------|
| Порівняння кав'ярень                        |                 |                   |             |              |                |           |                       |                         |               |
| Кав'ярня                                    | Поточний виторг | Попередній виторг | Δ виторг, % | Поточні чеки | Попередні чеки | Δ чеки, % | Поточний середній чек | Попередній середній чек | Δ сер. чек, % |
| Apostrophe 1                                | 398514.07       | 446666.30         | -10.78      | 2821         | 3162           | -10.78    | 141.27                | 141.26                  | 0.00          |
| Apostrophe 2                                | 437236.60       | 495296.39         | -11.72      | 3472         | 3828           | -9.30     | 125.93                | 129.39                  | -2.67         |
| Apostrophe 3                                | 289550.18       | 335070.39         | -13.59      | 2744         | 3095           | -11.34    | 105.52                | 108.26                  | -2.53         |
| Apostrophe 4                                | 269230.84       | 260888.26         | 3.20        | 2415         | 2348           | 2.85      | 111.48                | 111.11                  | 0.33          |
| Apostrophe 5                                | 201613.25       | 222168.25         | -9.25       | 1679         | 1970           | -14.77    | 120.08                | 112.78                  | 6.48          |

| Рік до року: 2026-04-01 - 2026-04-30 |                 |                   |             |              |                |           |                       |                         |               |
|--------------------------------------|-----------------|-------------------|-------------|--------------|----------------|-----------|-----------------------|-------------------------|---------------|
| Порівняння кав'ярень                 |                 |                   |             |              |                |           |                       |                         |               |
| Кав'ярня                             | Поточний виторг | Попередній виторг | Δ виторг, % | Поточні чеки | Попередні чеки | Δ чеки, % | Поточний середній чек | Попередній середній чек | Δ сер. чек, % |
| Apostrophe 1                         | 398514.07       | 385251.95         | 3.44        | 2821         | 2665           | 5.85      | 141.27                | 144.56                  | -2.28         |
| Apostrophe 2                         | 437236.60       | 476354.40         | -8.21       | 3472         | 4015           | -13.52    | 125.93                | 118.64                  | 6.14          |
| Apostrophe 3                         | 289550.18       | 265067.90         | 9.24        | 2744         | 2783           | -1.40     | 105.52                | 95.25                   | 10.79         |
| Apostrophe 4                         | 269230.84       | 0.00              |             | 2415         | 0              |           | 111.48                | 0.00                    |               |
| Apostrophe 5                         | 201613.25       | 0.00              |             | 1679         | 0              |           | 120.08                | 0.00                    |               |

| Автоматичні рекомендації |              |                   |                                 |                                                                                                                                   |                                                                                                                                            |
|--------------------------|--------------|-------------------|---------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| Рівень                   | Об'єкт       | Період            | Сигнал                          | Пояснення                                                                                                                         | Рекомендація                                                                                                                               |
| Кав'ярня                 | Apostrophe 2 | Місячний висновок | Системне просідання             | Місяць до місяця: виторг -11.72%, чеки -9.30%, середній чек -2.67%. Рік до року виторг змінився на -8.21%.                        | Локація просідає і за кількістю гостей, і за якістю продажу. Потрібен детальніший аудит роботи точки, команди та структури попиту.         |
| Кав'ярня                 | Apostrophe 3 | Місячний висновок | Системне просідання             | Місяць до місяця: виторг -13.59%, чеки -11.34%, середній чек -2.53%. Рік до року виторг змінився на +9.24%.                       | Локація просідає і за кількістю гостей, і за якістю продажу. Потрібен детальніший аудит роботи точки, команди та структури попиту.         |
| Кав'ярня                 | Apostrophe 1 | Місячний висновок | Падіння трафіку                 | Місяць до місяця: виторг -10.78%, чеки -10.78%, середній чек +0.00%. Рік до року виторг змінився на +3.44%.                       | Середній рівень продажу не змінився, але гостей стало менше. Варто перевірити трафік, сезонність, зовнішні фактори та локальний маркетинг. |
| Кав'ярня                 | Apostrophe 4 | Місячний висновок | Зростання за рахунок трафіку    | Місяць до місяця: виторг +3.20%, чеки +2.85%, середній чек +0.33%. Річне порівняння недоступне через нульову базу минулого року.  | Гостей стало більше, але якість продажу майже не змінилась. Потрібно зберегти джерело трафіку та посилити апсейл усередині чека.           |
| Кав'ярня                 | Apostrophe 5 | Місячний висновок | Менше гостей, але сильніший чек | Місяць до місяця: виторг -9.25%, чеки -14.77%, середній чек +6.48%. Річне порівняння недоступне через нульову базу минулого року. | Команда добре працює всередині чека, однак гостей стало менше. Основний фокус варто змістити на залучення трафіку.                         |