

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ І ПРИКЛАДНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

АНДРІЄНКО СОФІЯ ОЛЕГІВНА

Допускається до захисту:

в.о. _____ завідувача _____ кафедри
інформаційних технологій,
д-р. техн. наук, професор

_____ Наталія ВЕСЕЛОВСЬКА

« ____ » _____ 2026 р.

**РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АГРЕГАЦІЇ ТА
АНАЛІЗУ ВІДВІДУВАНOSTІ ОНЛАЙН-ЗУСТРІЧЕЙ У MS TEAMS НА
ОСНОВІ CSV-ДАНИХ**

Спеціальність 122 Комп'ютерні науки

Кваліфікаційна (бакалаврська) робота

Керівник:

Юрій АНТОНОВ, доцент кафедри
інформаційних технологій,
к. фіз.-мат. наук, доцент

Оцінка: _____ / _____ / _____
(бали/за шкалою ЄКТС/за національного шкалою)

Голова ЕК: _____
(підпис)

Вінниця - 2026

АНОТАЦІЯ

Андрієнко С.О. Розробка програмного забезпечення для агрегації та аналізу відвідуваності онлайн-зустрічей у MS TEAMS на основі CSV-даних. Спеціальність 122 «Комп'ютерні науки», Освітня програма «Комп'ютерні науки». Донецький національний університет імені Василя Стуса, Вінниця, 2026.

У кваліфікаційній роботі розроблено програмне забезпечення для автоматизованої агрегації та аналізу відвідуваності онлайн-зустрічей у MS Teams на основі CSV-даних. Досліджено методи обробки та візуалізації статистичних показників участі користувачів. Реалізовано алгоритми агрегації для формування зведеної аналітики відвідуваності.

Ключові слова: агрегація даних, аналіз даних, MS Teams, CSV, онлайн-зустрічі, програмне забезпечення.

63 с., 16 рис., 11 табл., 52 джерела.

ABSTRACT

Andriienko S.O. Software Development for Aggregation and Analysis of Online Meeting Attendance in MS Teams Based on CSV Data. Specialty 122 “Computer Science”, Educational Program “Computer Science”. Vasyl Stus Donetsk National University, Vinnytsia, 2026.

In the qualification work, software for automated aggregation and analysis of online meeting attendance in MS Teams based on CSV data was developed. Methods for processing and visualization of statistical indicators of user participation were investigated. Aggregation algorithms for generating consolidated attendance analytics were implemented.

Keywords: data aggregation, attendance analysis, MS Teams, CSV, online meetings, software development.

63 p., 16 fig., 11 table, 52 sources.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	5
ВСТУП	6
РОЗДІЛ 1 АНАЛІЗ ТА КЛАСИФІКАЦІЯ ІСНУЮЧИХ СИСТЕМ.....	8
1.1 Огляд платформи MS Teams та її можливостей для відстеження відвідуваності.	8
1.2 Огляд проєктів автоматизації освітньої діяльності на кафедрі.....	11
1.3 Огляд існуючих інструментів для обробки та аналізу даних відвідуваності	11
1.4 Основи проєктування реляційних баз даних.....	16
1.5 Архітектурні підходи до побудови програмного забезпечення.....	18
Висновки до першого розділу.....	19
РОЗДІЛ 2 ВИЗНАЧЕННЯ ВИМОГ ДО ПРОГРАМИ ТА ПОБУДОВА МОДЕЛІ ПРОГРАМНОГО ПРОДУКТУ	21
2.1 Визначення вимог до програмного забезпечення.....	21
2.1.1 Функціональні вимоги до програмного забезпечення	21
2.1.2 Нефункціональні вимоги.....	22
2.1.3 Вимоги до інформаційного забезпечення.....	23
2.1.4 Вимоги до графічного інтерфейсу користувача	25
2.2 Аналіз структури CSV-звітів MS Teams.....	25
2.3 Алгоритм парсингу CSV-файлів MS Teams	28
2.4 Алгоритм агрегації даних відвідуваності	29
2.5 Модель бази даних	33
2.6 Вибір технологій та інструментів розробки.....	34
2.7 Проєктування архітектури системи	35
2.7.1 Загальна архітектура системи	35
2.7.2 Опис основних модулів	36
2.7.3 Принципи проєктування архітектури	37
2.7.4 Точка входу програми	37
2.7.5 Переваги обраної архітектури	39

	4
Висновки до другого розділу	39
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ДОДАТКУ	42
3.1 Реалізація модуля парсингу CSV-файлів.....	42
3.1.1 Обґрунтування вибору основних технологій.....	44
3.1.2 Переваги обраного технологічного стеку.....	45
3.2 Реалізація інтерфейсу користувача	46
3.2.1 Загальна структура інтерфейсу.....	46
3.2.2 Сторінка завантаження CSV-файлів (upload.py).....	46
3.2.3 Сторінка дашборду (аналітики) (dashboard.py).....	47
3.2.4 Сторінка історії зустрічей (history.py)	48
3.2.5 Загальні рішення щодо інтерфейсу	48
3.3 Реалізація модуля аналітики	49
3.3.1 Загальна архітектура модуля	50
3.3.2 Система фільтрації даних.....	50
3.3.3 Формування зведених показників	51
3.3.4 Таблиця «Учасники».....	51
3.3.6 Функціонал експорту	51
3.4 Тестування програмного забезпечення.....	52
3.4.1 Методика тестування.....	52
3.4.2 Результати тестування	53
3.4.3 Загальні результати тестування.....	54
3.5 Системні вимоги до програмного забезпечення.....	54
3.6 Перспективи розвитку	55
Висновки до третього розділу.....	56
ВИСНОВОК.....	57
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	59

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

- BOM – Byte Order Mark (маркер послідовності байтів)
- CSV – Comma-Separated Values (значення, розділені комами)
- DAX – Data Analysis Expressions (мова виразів аналізу даних)
- LE – Little-Endian (порядок байтів від молодшого до старшого)
- RFC – Request for Comments (документ з описом стандартів Інтернету)
- UPN – User Principal Name (ім'я користувача в каталозі)

ВСТУП

Збільшення кількості онлайн-зустрічей у корпоративному та освітньому середовищі актуалізувало задачу контролю та аналізу залученості учасників. Організації та навчальні заклади, що активно використовують MS Teams, щодня стикаються з необхідністю збирати, обробляти та аналізувати записи про присутність на зустрічах. Однак стандартні звіти платформи мають суттєві обмеження, а ручна обробка CSV-файлів потребує значних часових витрат та підвищує ризик помилок при опрацюванні даних.

Проблема полягає не лише в технічній складності обробки даних, а й у відсутності зручних інструментів, які б дозволяли оперативно отримувати узагальнену інформацію про відвідуваність. Перевірка присутності студентів на заняттях [1] або аналіз участі працівників у корпоративних нарадах часто потребують ручного зведення та обробки великої кількості табличних даних. Ця проблема є актуальною як для великих організацій, так і для невеликих команд чи окремих викладачів, які щотижня проводять заняття в онлайн-форматі.

Окремої уваги заслуговує той факт, що навіть за наявності технічних знань обробка великої кількості CSV-файлів вручну залишається трудомістким процесом. Коли йдеться про десятки зустрічей на місяць з різною кількістю учасників, зведення даних у єдину структуру потребує значних часових ресурсів та ускладнює оперативний аналіз інформації.

Об'єктом дослідження є процес агрегації та аналізу даних відвідуваності онлайн-зустрічей.

Предметом дослідження – методи, алгоритми та програмні засоби для обробки CSV-звітів, що генеруються платформою MS Teams.

Метою роботи стала розробка програмного забезпечення, яке забезпечує автоматизацію основних процесів обробки та аналізу даних відвідуваності. Розроблене програмне забезпечення призначене для спрощення процесу аналізу даних відвідуваності та не потребує спеціальних технічних навичок для використання.

Для досягнення поставленої мети необхідно вирішити такі **завдання**:

- проаналізувати структуру CSV-звітів MS Teams та виявити особливості, що ускладнюють їх автоматизовану обробку;
- дослідити існуючі інструменти для аналізу відвідуваності та визначити їхні переваги та недоліки;
- розробити алгоритми парсингу та агрегації даних з урахуванням варіативності вхідних файлів;
- спроектувати модель бази даних для локального зберігання обробленої інформації;
- реалізувати програмне забезпечення з графічним інтерфейсом, що забезпечує завантаження, обробку та візуалізацію даних відвідуваності;
- провести тестування розробленого продукту на реальних даних.

Практичне значення отриманих результатів полягає у можливості використання розробленого програмного забезпечення в навчальних закладах та організаціях для автоматизованого контролю відвідуваності онлайн-заходів без потреби в ручній обробці даних.

Зв'язок роботи з науковими програмами, планами, темами. Наведені у бакалаврській роботі дослідження пов'язані з фундаментальною науково-дослідною роботою «Розвиток методів комп'ютерно-математичного моделювання складних систем та процесів у сфері освіти та діяльності IT-підприємств» (№ держреєстрації 0126U003221, 2026-2031 рр.).

Структура роботи. Кваліфікаційна робота складається зі вступу, трьох розділів, висновків, списку використаних джерел (52 найменування) та додатків. Перший розділ присвячено аналізу предметної області та існуючих інструментів. У другому розділі визначено вимоги до програмного забезпечення, розроблено алгоритми та спроектовано архітектуру системи. Третій розділ містить опис реалізації додатку та результати тестування.

РОЗДІЛ 1

АНАЛІЗ ТА КЛАСИФІКАЦІЯ ІСНУЮЧИХ СИСТЕМ

1.1 Огляд платформи MS Teams та її можливостей для відстеження відвідуваності.

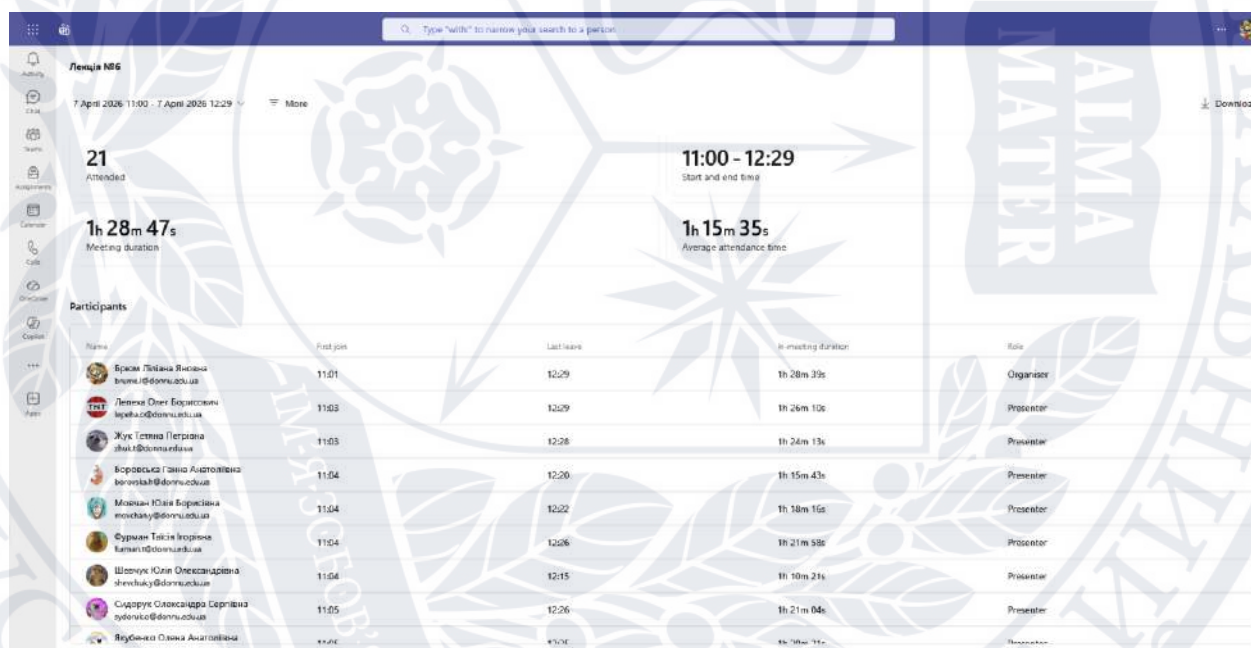
Microsoft Teams – це сучасна корпоративна платформа для командної співпраці, розроблена компанією Microsoft як один з ключових компонентів екосистеми Microsoft 365. Вона поєднує в собі інструменти обміну повідомленнями, проведення аудіо- та відеоконференцій, спільної роботи з документами в реальному часі та планування завдань. Крім того, Teams підтримує інтеграцію з іншими сервісами Microsoft, зокрема Outlook, SharePoint і OneDrive, а також сторонніми додатками. Завдяки такому комплексному підходу Teams став однією з найбільш поширених платформ для командної роботи [2, 3].

Станом на 2026 рік кількість щомісячних активних користувачів платформи Microsoft Teams перевищує 320 мільйонів осіб, а кількість організацій, що використовують сервіс, сягає понад один мільйон по всьому світу [2, 3]. Платформа демонструє стабільне зростання популярності протягом останніх років і широко використовується як інструмент для проведення онлайн-зустрічей у різних сферах діяльності.

У сучасному середовищі гібридної роботи та навчання Microsoft Teams активно застосовується для організації зустрічей різного формату та масштабу: планових робочих нарад, лекцій, семінарів, практичних занять, тренінгів, вебінарів, захисту кваліфікаційних робіт та неформальних обговорень. Платформа підтримує як невеликі зустрічі в межах однієї команди, так і великі заходи з участю сотень і тисяч учасників. Організатор зустрічі має широкий набір інструментів керування: налаштування доступу учасників, увімкнення або вимкнення чату, запис зустрічі, використання інтерактивних дощок, розбиття на кімнати для групової роботи (breakout rooms), призначення ролей доповідачів та інші засоби організації онлайн-взаємодії [4]. Окрім того, Teams підтримує

інтеграцію з навчальними платформами (Moodle, Google Classroom) та системами управління навчанням (LMS [5]), що робить його особливо привабливим для закладів освіти. Викладачі можуть створювати команди для навчальних груп, розміщувати матеріали в каналах та відстежувати активність студентів.

Однією з важливих можливостей платформи є функція відстеження відвідуваності учасників онлайн-зустрічей. Організатор або співорганізатор може отримати детальний звіт про присутність безпосередньо під час зустрічі або після її завершення. Такий документ містить детальну інформацію про кожного учасника: повне ім'я, електронну адресу (UPN), час приєднання до зустрічі, час виходу, загальну тривалість перебування, роль у зустрічі (Organizer, Presenter, Attendee) та інші метадані. Завантаження звіту здійснюється у форматі CSV-файлу через панель учасників або вкладку Attendance у чаті зустрічі [6].



Лекція MS6
7 April 2026 11:00 - 7 April 2026 12:29

21 Attended

1h 28m 47s Meeting duration

11:00 - 12:29 Start and end time

1h 15m 35s Average attendance time

Participants

Name	First join	Last leave	In meeting duration	Role
Брош Лідія Янівна lidia.bros@ukma.edu.ua	11:01	12:29	1h 28m 33s	Organizer
Ліска Олег Борисович oleg.liska@ukma.edu.ua	11:03	12:29	1h 26m 10s	Presenter
Жук Тетяна Петрівна tetyana.zhuk@ukma.edu.ua	11:03	12:28	1h 24m 13s	Presenter
Воробей Тетяна Анатоліївна tetyana.vorobey@ukma.edu.ua	11:04	12:20	1h 15m 43s	Presenter
Мозик Юлія Борисівна yuliya.mozik@ukma.edu.ua	11:04	12:22	1h 18m 16s	Presenter
Фурман Тетяна Ігорівна tetyana.furman@ukma.edu.ua	11:04	12:26	1h 21m 58s	Presenter
Шенчук Юлія Олександрівна yuliya.shenchuk@ukma.edu.ua	11:04	12:15	1h 10m 21s	Presenter
Сидорук Олександр Сергійович oleksandr.sidoruk@ukma.edu.ua	11:05	12:26	1h 21m 04s	Presenter
Васильченко Олена Анатоліївна	11:06	12:26	1h 20m 11s	Attendee

Рисунок 1.1 – Скріншот інтерфейсу MS Teams із функцією завантаження звіту відвідуваності

Функція генерації звітів відвідуваності була впроваджена у 2020 році і з того часу неодноразово вдосконалювалася. Сучасні версії звіту фіксують не лише загальний час присутності, але й кожен окрему сесію підключення учасника. Це дозволяє детально аналізувати поведінку учасників, зокрема

кількість входів і виходів протягом однієї зустрічі, та оцінювати їхню активність [6, 7].

Однак функціонал має низку обмежень. По-перше, доступ до генерації та завантаження звіту відвідуваності мають лише організатор зустрічі та користувачі з роллю співорганізатора (Presenter). По-друге, існує обмежений термін зберігання даних: дані про відвідуваність доступні для завантаження на серверах Microsoft лише протягом 30 днів після проведення зустрічі, після чого вони автоматично видаляються без можливості відновлення [8, 9].

Це робить своєчасне завантаження та архівування CSV-файлів важливим завданням для всіх організацій та навчальних закладів, які ведуть систематичний облік відвідуваності. Без створення власної бази даних та регулярного збереження звітів інформація про відвідуваність може бути безповоротно втрачена вже через місяць після зустрічі.

По-третє, вбудовані інструменти аналітики MS Teams не підтримують автоматичну агрегацію даних з великої кількості зустрічей. Користувач не може за допомогою стандартних засобів платформи оперативно отримувати узагальнену статистику за певний період (наприклад, за місяць або тиждень), побудувати динаміку відвідуваності конкретного учасника чи групи, або здійснити фільтрацію за типом заняття, групою чи потоком. Стандартні звіти орієнтовані переважно на разовий аналіз окремої зустрічі і обмежено адаптовані під потреби вищої освіти (систематичний облік студентів) або регулярного корпоративного контролю [6].

Таким чином, попри значний потенціал і широкі можливості Microsoft Teams, відсутність гнучких вбудованих інструментів для довгострокової агрегації та аналізу даних відвідуваності створює суттєве ускладнення для кінцевих користувачів. Викладачі, куратори навчальних закладів, HR-менеджери та керівники команд потребують значних часових витрат на ручну обробку CSV-файлів, що ускладнює процес аналізу та контролю відвідуваності їхньої роботи та підтверджує потребу у розробці спеціалізованого програмного забезпечення.

1.2 Огляд проєктів автоматизації освітньої діяльності на кафедрі

Перш ніж перейти до аналізу систем для відстеження відвідуваності, варто розглянути, які завдання освітнього процесу вже автоматизувалися на кафедрі інформаційних технологій.

Автоматизація тестування знань розглядається в [10]. Студентські роботи [11, 12] присвячені розробці систем тестування, а [13] – дистанційному контролю з урахуванням самооцінки студентів.

Питання формування розкладу висвітлено в [14, 16], а [15] пропонує використання телеграм-бота для доступу до розкладу. Автоматизація роботи деканату [17] та розподілу студентів між викладачами [18] також є важливими напрямками.

Значний внесок у автоматизацію наукової експертизи зробив професор кафедри С.Д. Штовба. Його роботи [19, 20] присвячені автоматичному підбору опонентів дисертацій та рецензентів.

Підходи до інтелектуального аналізу даних, застосовні в освіті, запропоновано в [21].

Таким чином, на кафедрі накопичено досвід автоматизації різних освітніх процесів. Однак, як показано в підрозділі 1.3, спеціалізованого інструменту для агрегації та аналізу відвідуваності з MS Teams серед них немає, що підтверджує актуальність роботи.

1.3 Огляд існуючих інструментів для обробки та аналізу даних відвідуваності

Одним із найбільш поширених інструментів для обробки даних відвідуваності онлайн-зустрічей у MS Teams серед викладачів і керівників залишається Microsoft Excel. Після завантаження CSV-файлів користувачі зазвичай вручну імпортують дані, об'єднують таблиці з різних зустрічей за допомогою функцій VLOOKUP, Power Query [22] або ручного об'єднання даних. Далі застосовуються зведені таблиці (PivotTables), умовне форматування, формули для розрахунку відсотків відвідуваності та побудови графіків.

Перевагами такого підходу є повна доступність інструменту, відсутність

додаткових витрат та добре знання Excel більшістю користувачів. Однак при роботі з великим обсягом даних (наприклад, 50–120 зустрічей за семестр) ефективність такого підходу суттєво знижується. Ручне об'єднання файлів, виправлення некоректних форматів дати, агрегація кількох сесій одного учасника, розрахунок середніх показників і фільтрація вимагають значних часових витрат. Крім того, зростає ймовірність помилок, пов'язаних із людським фактором, а процес оновлення даних потребує повторного виконання значної частини операцій [9].

Альтернативним підходом є використання Microsoft Access. CSV-файли попередньо конвертуються у формат Excel або імпортуються безпосередньо, після чого в Access створюються запити для об'єднання даних з різних зустрічей, фільтрації за учасниками, а також формуються звіти для друку. Access дозволяє працювати з більшими обсягами даних, ніж Excel (до 2 ГБ), що є перевагою при обробці десятків CSV-файлів за семестр. Однак він потребує ручного налаштування запитів та звітів для кожної нової зустрічі, а також знання SQL для створення складних агрегацій.

Teams Admin Center [23] та вбудовані засоби аналітики Microsoft 365 призначені переважно для IT-адміністраторів організацій. Вони надають агреговану статистику на рівні організації: загальну кількість зустрічей, кількість активних користувачів, середню тривалість зустрічей, географію підключень та інші технічні метрики. Ці інструменти зручні для отримання загальної картини використання платформи, але недостатньо придатні для вирішення практичних задач викладачів. Вони не дозволяють формувати індивідуальну статистику відвідуваності конкретного студента за весь семестр, проводити аналіз за групами, потоками чи типами занять (лекція, практика, семінар) [7, 9].

Power BI [24] вважається одним із найбільш поширених інструментів для бізнес-аналітики в екосистемі Microsoft. Він дозволяє імпортувати значні обсяги CSV-файлів, створювати багаторівневі моделі даних, встановлювати зв'язки між таблицями, писати розрахункові формули на мові DAX та будувати інтерактивні дашборди з фільтрами, діаграмами та візуалізаціями. Теоретично Power BI може

використовуватися для агрегації та аналізу даних відвідуваності.

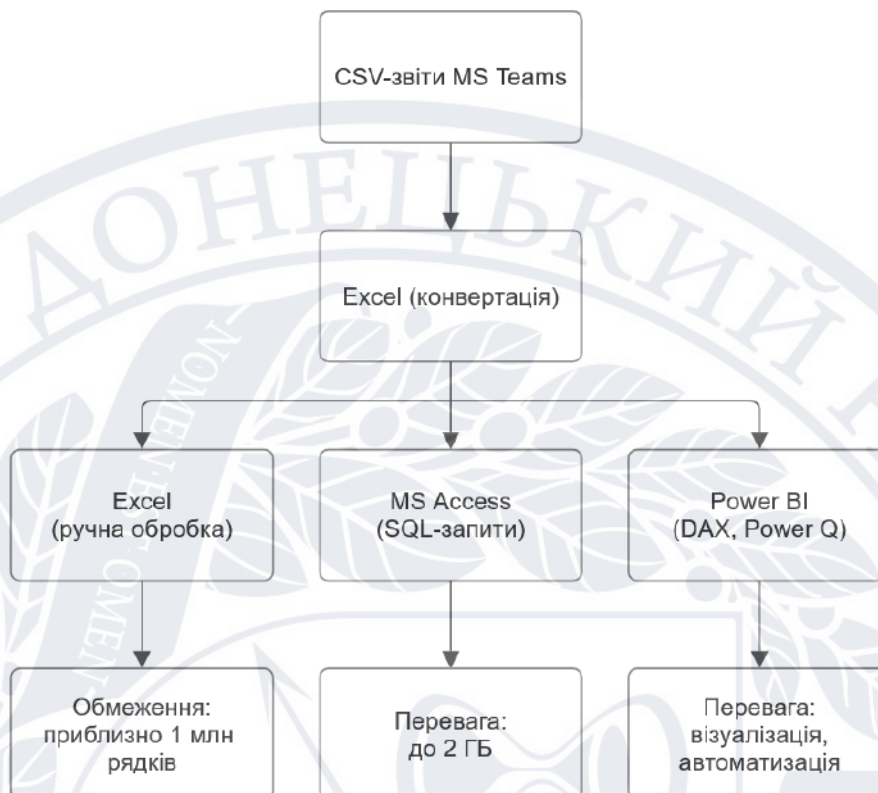


Рисунок 1.2 – Альтернативні шляхи обробки CSV-звітів MS Teams після конвертації в Excel

Однак на практиці для ефективної роботи з цим інструментом потрібні технічні знання у сфері обробки та моделювання даних: розуміння принципів моделювання даних, вміння працювати з Power Query, знання DAX та регулярне оновлення наборів даних. Для звичайного викладача або HR-менеджера, який не є спеціалістом у сфері даних, використання інструменту може бути складним через відсутність необхідних технічних знань. Крім того, повноцінне використання Power BI часто вимагає платної ліцензії Pro або Premium, а також стабільного підключення до інтернету для оновлення даних.

Microsoft Viva Insights [25] є комерційне рішення для розширеної аналітики продуктивності та залученості співробітників. Інструмент опрацьовує великі масиви даних, аналізує особливості комунікації, показує, скільки часу працівники витрачають на зустрічі, наскільки вони залучені, і навіть формує персональні рекомендації.

Однак Viva Insights має низку обмежень для використання в освітньому середовищі. По-перше, він орієнтований насамперед на корпоративний сектор і погано адаптований під специфіку університетів. По-друге, інструмент вимагає дорогих ліцензій (зазвичай рівня Microsoft 365 E5 або спеціальних доповнень), що є недоступним для більшості навчальних закладів. По-третє, він не підтримує важливі для освіти параметри, такі як розподіл за академічними групами, потоками, типами занять чи семестрами.

Окрім офіційних інструментів Microsoft, у відкритому доступі на платформі GitHub та в PowerShell Gallery існують численні open-source скрипти та допоміжні програмні інструменти, створені спільнотою розробників. Деякі з них дозволяють автоматизувати парсинг CSV-файлів, агрегацію за учасниками, розрахунок відсотків відвідуваності та базову візуалізацію.

Перевагами таких рішень є їхня безкоштовність, можливість вільної модифікації коду та відсутність залежності від платних ліцензій. Однак більшість цих скриптів мають ряд обмежень: відсутність зручного графічного інтерфейсу, необхідність встановлення середовища розробки (Python, PowerShell тощо), потребу в знаннях програмування, відсутність централізованого зберігання даних в базі та відсутність регулярного оновлення аналітики при додаванні нових файлів. Через це такі інструменти є складними у використанні для переважної більшості нетехнічних користувачів – викладачів та керівників [26].

Крім того, у роботі [27] досліджуються підходи до автоматизації оцінювання відповідей за допомогою експертних систем, що може бути корисно для подальшого вдосконалення систем тестування.

Для наочності порівняння основних інструментів наведено таблицю 1.2.

Як свідчать дані таблиці 1.1, жоден із розглянутих інструментів не забезпечує виконання всіх необхідних критеріїв одночасно.

Excel, хоча й є доступним у складі підписки Microsoft 365 та працює офлайн, вимагає ручної обробки кожного CSV-файлу, що призводить до значних часових витрат і підвищує ризик помилок.

Таблиця 1.1 – Порівняльний аналіз існуючих інструментів

Критерій	Excel	MS Access	Teams Admin Center	Power BI	Viva Insights
Простота використання	Середня	Низька	Висока	Низька	Низька
Агрегація кількох файлів	Вручну	Запити (SQL)	Ні	Так	Так
Індивідуальна статистика	Вручну	Запити (SQL)	Ні	Так	Так
Автоматизація обробки	Ні	Ні (ручні запити)	Ні	Частково	Так
Підтримка освітньої специфіки	Ні	Ні	Ні	Потребує налаштування	Ні
Робота без Інтернету	Так	Так	Ні	Можливо	Ні
Вартість	У складі підписки M365	У складі M365	Безкоштовно	Безкоштовно або Pro ~10\$/міс	~57\$/міс (в складі E5)

Teams Admin Center простий у використанні, але не підтримує агрегацію даних з багатьох зустрічей та не дозволяє отримувати індивідуальну статистику учасників.

Power BI надає широкі можливості для автоматизації та візуалізації, однак потребує спеціальних знань (DAX, Power Query), а його повноцінне використання передбачає платну ліцензію (Pro ~10\$/користувача/міс).

Viva Insights пропонує розширену аналітику залученості, але вимагає найдорожчої ліцензії Microsoft 365 E5 (~57\$ за користувача на місяць) та не враховує освітню специфіку (академічні групи, потоки, типи занять).

1.4 Основи проєктування реляційних баз даних

Для забезпечення ефективного, несуперечливого та цілісного зберігання даних у розроблюваній системі було застосовано теорію проєктування реляційних баз даних. Метою цього процесу є створення структури даних, яка мінімізує надлишковість (дублювання) та усуває потенційні аномалії при внесенні, оновленні чи видаленні інформації [28, с. 327-329]. Проєктування базується на принципах нормалізації.

Нормалізація бази даних

Нормалізація – це формальний процес перевірки та оптимізації структури бази даних. У цій роботі проєктування здійснювалося до третьої нормальної форми (3НФ), що є загальноприйнятим компромісом між уникненням аномалій та збереженням продуктивності [29, с. 85-87].

- Перша нормальна форма (1НФ): Забезпечує атомарність (неподільність) кожного значення в таблицях. У розробленій моделі в жодному полі не зберігаються списки чи масиви значень, кожен стовпець містить лише одне значення, а всі записи в таблиці однорідні [24, с. 348-350].
- Друга нормальна форма (2НФ): Застосовується до таблиць, які вже знаходяться в 1НФ, та вимагає, щоб всі неключові поля залежали від повного первинного ключа. У моделі attendance первинний ключ id є унікальним ідентифікатором, а всі інші поля (duration_minutes, attendance_percentage) описують саме цей запис про відвідування, а не лише частину ключа [29, с. 90-92].
- Третя нормальна форма (3НФ): Вимагає відсутності транзитивних залежностей, тобто неключові поля не повинні залежати від інших неключових полів [28, с. 366-368]. Саме тому такі дані, як назва групи чи потоку, винесено в окремі таблиці (groups, streams), а в таблиці meetings зберігаються лише їхні ідентифікатори (зовнішні ключі).

Уникнення аномалій проектування

Дотримання вищезазначених нормальних форм дозволяє мінімізувати або повністю уникнути шести типових аномалій [29, с. 112-118]:

- Аномалія вставлення (Insertion Anomaly): Виникає, коли не можна додати об'єкт до БД без існування іншого, пов'язаного об'єкта. У спроектованій схемі можна вільно додати нового учасника в таблицю `participants`, навіть якщо він ще не відвідав жодної зустрічі.
- Аномалія оновлення (Update Anomaly): Виникає через надлишковість даних, коли зміна одного значення вимагає оновлення багатьох рядків. Це унеможлиблюється шляхом винесення повторюваних даних (наприклад, назв груп) в окремі таблиці.
- Аномалія видалення (Deletion Anomaly): Виникає, коли під час видалення одного об'єкта ми ненавмисно втрачаємо дані про інший. Наприклад, видалення останньої зустрічі певного типу не призведе до зникнення інформації про сам тип заняття з таблиці `meeting_types`.
- Аномалія надлишковості (Redundancy Anomaly): Призводить до невиправданого дублювання даних, що збільшує розмір БД. Винесення інформації про групи та потоки в окремі таблиці повністю вирішує цю проблему для спроектованої моделі.
- Аномалія часткової залежності (Partial Dependency Anomaly): Характерна для таблиць зі складеними ключами, коли поле залежить лише від частини ключа. Уникається шляхом переходу до 2НФ.
- Аномалія транзитивної залежності (Transitive Dependency Anomaly): Виникає, коли одне неключове поле залежить від іншого неключового поля. Уникається шляхом переходу до 3НФ та використанням зовнішніх ключів.

Таким чином, на етапі проєктування було закладено підґрунтя для створення логічної, несуперечливої та продуктивної моделі даних, що відповідає сучасним вимогам до розробки реляційних баз даних [30].

1.5 Архітектурні підходи до побудови програмного забезпечення

Архітектура програмного забезпечення визначає загальну структуру системи, розподіл відповідальностей між компонентами та принципи їх взаємодії. Вибір архітектури безпосередньо впливає на підтримуваність, тестованість, масштабованість та надійність системи [31, с. 45-48].

Одношарова (монолітна) архітектура

У одношарових додатках весь код – інтерфейс користувача, бізнес-логіка та робота з даними – зосереджено в одному модулі. Такий підхід є простим у реалізації та підходить для невеликих програм з мінімальною логікою (наприклад, прості калькулятори, одноразові скрипти). Однак при ускладненні системи одношарова архітектура призводить до таких проблем:

- Складна підтримка – зміна в одній частині може вплинути на всю систему;
- Низька тестованість – неможливість тестувати компоненти ізольовано;
- Проблеми масштабування – додавання нової функціональності потребує змін у всьому коді [32, с. 87-90].

Багатошарова (n-tier) архітектура

Багатошарова архітектура передбачає розділення системи на логічні рівні, кожен з яких відповідає за окрему функціональну область. Класична тришарова архітектура включає:

1. Рівень представлення (Presentation Layer) – відповідає за взаємодію з користувачем;
2. Рівень бізнес-логіки (Business Logic Layer) – реалізує основні алгоритми та правила;

3. Рівень доступу до даних (Data Access Layer) – забезпечує збереження та отримання даних.

Такий підхід забезпечує низку переваг [31, с. 120-125]:

- Чітке розділення відповідальностей – кожен рівень виконує лише свої функції;
- Підтримуваність – зміни в одному рівні не впливають на інші;
- Тестованість – можливість тестувати кожен рівень окремо;
- Можливість заміни – наприклад, заміна SQLite на PostgreSQL не потребує змін у рівні бізнес-логіки.

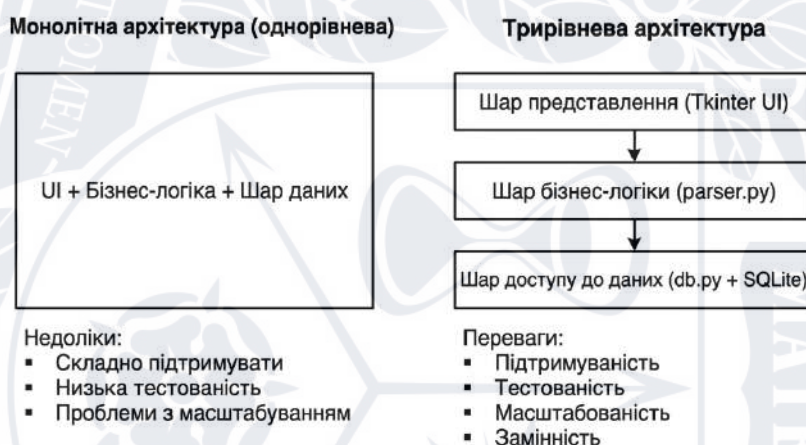


Рисунок 1.3 – Порівняння одношарової та тришарової архітектури

Для розроблюваної системи обрано тришарову архітектуру, оскільки вона забезпечує оптимальний баланс між простотою реалізації та гнучкістю для подальшого розширення. Детальний опис реалізованої архітектури наведено в підрозділі 2.7.

Висновки до першого розділу

У першому розділі проведено аналіз предметної області кваліфікаційної роботи. Розглянуто платформу Microsoft Teams як основний інструмент проведення онлайн-зустрічей, проаналізовано можливості вбудованого функціоналу відстеження відвідуваності, досліджено структуру CSV-звітів, а також виконано порівняльний аналіз існуючих інструментів їх обробки та аналізу.

Встановлено, що попри значну популярність MS Teams та наявність базової функції формування звітів відвідуваності, платформа має ряд обмежень. Звіти доступні лише протягом 30 днів, доступ до них має обмежене коло осіб. Крім того, відсутня можливість автоматичної агрегації даних з багатьох зустрічей, побудови динаміки відвідуваності та формування індивідуальної статистики учасників. Ці обмеження унеможливають ефективне використання стандартних засобів платформи для систематичного контролю відвідуваності в освітніх закладах.

Проведене дослідження підтверджує актуальність обраної теми. Виявлені особливості CSV-звітів безпосередньо вплинули на розробку алгоритму парсингу, а порівняльний аналіз існуючих систем дозволив сформулювати чіткі вимоги до нового програмного забезпечення. Отримані результати стали основою для формування технічних вимог до системи, проєктування її архітектури та подальшої практичної реалізації, які детально описані в наступних розділах роботи.

РОЗДІЛ 2

ВИЗНАЧЕННЯ ВИМОГ ДО ПРОГРАМИ ТА ПОБУДОВА МОДЕЛІ ПРОГРАМНОГО ПРОДУКТУ

2.1 Визначення вимог до програмного забезпечення

Визначення вимог є ключовим етапом проєктування програмного забезпечення. Саме від чіткості та повноти вимог залежить відповідність майбутнього продукту реальним потребам користувачів [33]. У рамках цієї кваліфікаційної роботи вимоги до системи агрегації та аналізу відвідуваності онлайн-зустрічей у MS Teams були сформульовані на основі аналізу проблематики, описаної в першому розділі, вивчення структури CSV-звітів платформи та опрацювання потреб основної цільової аудиторії – викладачів, кураторів академічних груп, методистів та керівників підрозділів.

Для зручності всі вимоги поділено на три групи: функціональні, нефункціональні та системні (вимоги до середовища експлуатації).

2.1.1 Функціональні вимоги до програмного забезпечення

Функціональні вимоги визначають, що саме повинне робити програмне забезпечення. Головною метою є повна автоматизація процесу обробки даних відвідуваності, який зараз виконується вручну та потребує значних часових витрат.

Система повинна забезпечувати завантаження та автоматичний парсинг CSV-файлів звітів [34] MS Teams з підтримкою різних кодувань (UTF-8, UTF-16), роздільників і регіональних форматів дати. Після парсингу програма має автоматично виявляти метадані зустрічі (назва, дата проведення, загальна тривалість, організатор) та агрегувати множинні сесії підключення одного учасника в єдиний запис з розрахунком сумарного часу присутності.

Важливою функціональною можливістю є збереження оброблених даних у локальну базу даних з можливістю доповнення кожної зустрічі метаінформацією: типом заняття (лекція, семінар, практичне заняття, нарада тощо), групою або потоком. Користувач повинен мати змогу переглядати, редагувати та видаляти збережені зустрічі, а також приховувати певних

учасників (наприклад, організаторів або технічні акаунти) з подальшої аналітики.

Додаток має формувати зведену аналітику відвідуваності, розраховувати відсоток присутності кожного учасника, проводити категоризацію (висока, середня, низька відвідуваність) та надавати інтерактивний дашборд з можливістю фільтрації даних за потоком, групою, типом заняття та періодом. Також передбачено експорт отриманих таблиць у CSV-формат для подальшого аналізу в інших програмах [35].



Рисунок 2.1 – Use Case діаграма основних функціональних можливостей системи

2.1.2 Нефункціональні вимоги

Нефункціональні вимоги описують якісні характеристики рішення та умови її ефективної роботи.

Програмне забезпечення повинно забезпечувати високу продуктивність – стабільно обробляти до 300 CSV-файлів (понад 50 тисяч записів) без помітного сповільнення [26]. Час завантаження дашборду та перерахунку аналітики не повинен перевищувати 2–3 секунди навіть при значному обсязі даних.

Особлива увага приділяється надійності: система має коректно обробляти файли з різними кодуваннями, нестандартними форматами дати та пошкоджені

файли, видаючи зрозумілі повідомлення про помилки. Усі дані зберігаються виключно локально, що гарантує конфіденційність персональної інформації відповідно до вимог законодавства [36, 37].

Інтерфейс користувача має бути максимально інтуїтивним і орієнтованим на нетехнічних користувачів – викладачів та адміністраторів. Підтримка української мови, зручна навігація, кольорове кодування показників відвідуваності та контекстні підказки є обов’язковими.

Програмне забезпечення розробляється з використанням лише стандартної бібліотеки Python [32], що забезпечує простоту встановлення та портативність. Основним цільовим середовищем є операційна система Windows 10/11.

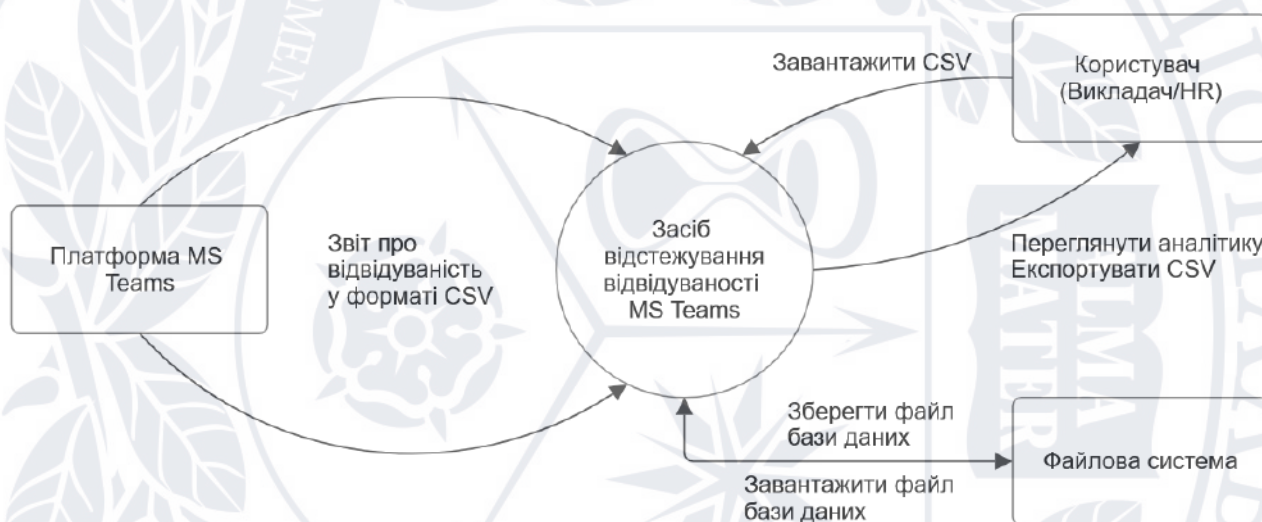


Рисунок 2.2 – Контекстна діаграма системи

2.1.3 Вимоги до інформаційного забезпечення

Перед проєктуванням бази даних необхідно визначити склад та структуру даних, які підлягають зберіганню та обробці, а також архітектурні особливості системи.

Архітектурні вимоги:

- Система є автономним персональним додатком – кожен користувач працює з власною локальною базою даних, одночасний доступ кількох користувачів до однієї БД не передбачений.
- Орієнтовний обсяг даних: до 1000 унікальних учасників, до 400 зустрічей

Вхідні дані (CSV-звіти MS Teams):

- метадані зустрічі: назва, дата проведення, загальна тривалість, організатор
- дані учасників: ПІБ, електронна пошта (UPN), роль у зустрічі, час приєднання/виходу, тривалість перебування

1. Summary						
Meeting title	Лекція №6					
Attended participants	21					
Start time	4/07/26, 11:00:57 AM					
End time	4/07/26, 12:29:44 AM					
Meeting duration	1h 28m 47s					
Average attendance time	1h 15m 35s					
2. Participants						
Name	First Join	Last Leave	In-Meeting Duration	Email	Participant ID (UPN)	Role
Брюм Ліліана Яновна	4/07/26, 11:01:05 AM	4/07/26, 12:29:44 AM	1h 28m 39s	brume.l@donnu.edu.ua	brume.l@donnu.edu.ua	Organiser
Лепеха Олег Борисович	4/07/26, 11:03:25 AM	4/07/26, 12:29:35 AM	1h 26m 10s	lepeha.o@donnu.edu.ua	lepeha.o@donnu.edu.ua	Presenter
Жук Тетяна Петрівна	4/07/26, 11:03:56 AM	4/07/26, 12:28:09 AM	1h 24m 13s	zhuk.t@donnu.edu.ua	zhuk.t@donnu.edu.ua	Presenter
Боровська Ганна Анатоліївна	4/07/26, 11:04:26 AM	4/07/26, 12:20:09 AM	1h 15m 43s	borovska.h@donnu.edu.ua	borovska.h@donnu.edu.ua	Presenter
Мовчан Юлія Борисівна	4/07/26, 11:04:34 AM	4/07/26, 12:22:50 AM	1h 18m 16s	movchan.y@donnu.edu.ua	movchan.y@donnu.edu.ua	Presenter
Фурман Татія Ігорівна	4/07/26, 11:04:40 AM	4/07/26, 12:26:38 AM	1h 21m 58s	furman.t@donnu.edu.ua	furman.t@donnu.edu.ua	Presenter
Шевчук Юлія Олександрівна	4/07/26, 11:04:51 AM	4/07/26, 12:15:12 AM	1h 10m 21s	shevchuk.y@donnu.edu.ua	shevchuk.y@donnu.edu.ua	Presenter
Сидорук Олександра Сергіївна	4/07/26, 11:05:00 AM	4/07/26, 12:26:04 AM	1h 21m 04s	sydoruk.o@donnu.edu.ua	sydoruk.o@donnu.edu.ua	Presenter
Якубенко Олена Анатоліївна	4/07/26, 11:05:10 AM	4/07/26, 12:25:31 AM	1h 20m 21s	yakubenko.o@donnu.edu.ua	yakubenko.o@donnu.edu.ua	Presenter
Борсук Анна Валеріївна	4/07/26, 11:05:15 AM	4/07/26, 11:59:20 AM	0h 54m 05s	borsuk.a@donnu.edu.ua	borsuk.a@donnu.edu.ua	Presenter
Михайліста Ірина Василівна	4/07/26, 11:05:22 AM	4/07/26, 12:03:47 AM	0h 58m 25s	mykhailista.i@donnu.edu.ua	mykhailista.i@donnu.edu.ua	Presenter
Іванова Дар'я Володимирівна	4/07/26, 11:05:26 AM	4/07/26, 12:25:27 AM	1h 20m 01s	ivanova.d@donnu.edu.ua	ivanova.d@donnu.edu.ua	Presenter
Швець Вероніка Степанівна	4/07/26, 11:05:54 AM	4/07/26, 12:11:40 AM	1h 05m 46s	shvets.v@donnu.edu.ua	shvets.v@donnu.edu.ua	Presenter
Криворучко Анна Олександрівна	4/07/26, 11:06:06 AM	4/07/26, 12:05:15 AM	0h 59m 09s	kryvoruchko.a@donnu.edu.ua	kryvoruchko.a@donnu.edu.ua	Presenter
Грантовський Володимир Олександрович	4/07/26, 11:06:23 AM	4/07/26, 12:25:59 AM	1h 19m 36s	hrantovskiy.v@donnu.edu.ua	hrantovskiy.v@donnu.edu.ua	Presenter
Сіденко Наталя Юріївна	4/07/26, 11:06:56 AM	4/07/26, 12:29:29 AM	1h 22m 33s	sidenko.n@donnu.edu.ua	sidenko.n@donnu.edu.ua	Presenter
Абдулла Жанна Енаутуллаївна	4/07/26, 11:08:01 AM	4/07/26, 12:29:43 AM	1h 21m 42s	abdulla.z@donnu.edu.ua	abdulla.z@donnu.edu.ua	Presenter
Ходанівська Людмила Анатоліївна	4/07/26, 11:08:12 AM	4/07/26, 12:24:32 AM	1h 16m 20s	khodakivska.l@donnu.edu.ua	khodakivska.l@donnu.edu.ua	Presenter
Калашнікова Олег Віталійович	4/07/26, 11:08:29 AM	4/07/26, 12:23:16 AM	1h 14m 47s	kalashnikova.o@donnu.edu.ua	kalashnikova.o@donnu.edu.ua	Presenter
Воробець Марія Петрівна	4/07/26, 11:09:47 AM	4/07/26, 12:19:55 AM	1h 10m 08s	vorobets.m@donnu.edu.ua	vorobets.m@donnu.edu.ua	Presenter
Коваль Дмитро Володимирович	4/07/26, 11:10:08 AM	4/07/26, 12:28:10 AM	1h 18m 02s	koval.d@donnu.edu.ua	koval.d@donnu.edu.ua	Presenter

Рисунок 2.3 – Приклад фрагменту CSV-звіту MS Teams

Метадані, що додаються користувачем:

- тип заняття (лекція, семінар, практичне заняття, лабораторна робота, нарада)
- група (за потреби)
- потік (необов'язково)

Похідні (обчислювані) дані:

- відсоток відвідуваності кожного учасника відносно загальної тривалості зустрічі
- категорія відвідуваності (висока: $\geq 80\%$, середня: 50-79%, низька: $< 50\%$)
- сумарний час присутності учасника (при множинних підключеннях)

Вимоги до цілісності:

- кожна зустріч має унікальний ідентифікатор

- комбінація ПІБ + email є унікальною для ідентифікації учасника
- дата проведення зустрічі є обов'язковим полем
- загальна тривалість зустрічі не може дорівнювати нулю

Обмеження:

- обсяг даних розрахований на один факультет або кафедру (до 1000 унікальних учасників, до 300 зустрічей)
- зберігання виключно локальне (без передачі на зовнішні сервери)
- персональні дані (ПІБ, email) не передаються третім особам

2.1.4 Вимоги до графічного інтерфейсу користувача

Графічний інтерфейс має відповідати таким вимогам:

- Інтуїтивність – користувач має розуміти призначення кожного елемента без додаткових інструкцій.
- Мінімалізм – на головному екрані відображаються лише найнеобхідніші елементи.
- Зворотній зв'язок – будь-яка дія (завантаження файлу, збереження, видалення) супроводжується сповіщенням про успіх або помилку.
- Кольорове кодування – показники відвідуваності візуально відрізняються (зелений/жовтий/червоний).
- Адаптивність – вікно програми можна змінювати за розміром, елементи підлаштовуються.
- Україномовний інтерфейс – усі підписи, кнопки, повідомлення виконані українською мовою.

2.2 Аналіз структури CSV-звітів MS Teams

CSV (Comma-Separated Values) є поширеним форматом для експорту табличних даних завдяки простоті, універсальності та сумісності з більшістю офісних програм. Однак цей тип даних CSV не має єдиного стандартизованого вигляду. Найближчим до стандарту є документ RFC 4180, опублікований Internet Engineering Task Force у 2005 році, який описує базові правила формування CSV-файлів [38].

Звіти відвідуваності, що генеруються платформою Microsoft Teams, мають велику варіативність структури та параметрів, яка суттєво ускладнює їх автоматизовану обробку. На відміну від типових CSV-файлів, що створюються іншими системами, файли з MS Teams містять низку нестандартних особливостей, які можуть спричинити помилки під час обробки стандартними засобами.

Основні особливості структури таких файлів такі:

- Кодування файлу. Найчастіше використовується UTF-16 (Little Endian) або UTF-8 з BOM. Багато текстових редакторів та простих скриптів за замовчуванням очікують UTF-8, що призводить до некоректного відображення символів при відкритті файлу в програмі [6, 39].
- Роздільники полів. Хоча RFC 4180 рекомендує кому, MS Teams може використовувати кому або символ табуляції залежно від локалі користувача. У випадках, коли назва зустрічі містить кому, неправильне визначення роздільника призводить до хибного розбиття даних на поля [38, 40].
- Загальна структура файлу. Файл складається з двох основних частин: блоку метаданих зустрічі та таблиці учасників. Блок метаданих містить ключову інформацію про зустріч (Meeting Title, Start Time, Meeting Duration, Organizer тощо). Після метаданих без чіткого технічного роздільника розміщується таблиця з даними учасників. Відсутність явного роздільника між блоками є однією з основних труднощів для автоматичного парсингу [6].
- Формат дати і часу. Формат може відрізнитися залежно від регіональних налаштувань: MM/DD/YYYY, DD/MM/YY, різні розділювачі тощо. Некоректне розпізнавання дати може призвести до того, що зустріч буде віднесена до неправильного дня або іншого календарного періоду.

- Інформація учасників. Один учасник може мати декілька записів через повторні підключення та відключення під час зустрічі. Також зустрічаються порожні значення для гостей учасників, які приєдналися без облікового запису Microsoft. Це вимагає окремої логіки агрегації сесій одного й того ж учасника [6].
- Формат тривалості. Тривалість може бути записана у різних варіантах: “1h 30m”, “1:30:00”, “90 min”, просто число хвилин або навіть бути відсутньою. Така неоднорідність потребує використання регулярних виразів та додаткової логіки перетворення в єдиний числовий формат.

Таблиця 2.1 – Основна структура полів CSV-звіту

Назва поля	Тип даних	Опис	Приклад
Full Name	Текст	Повне ім'я учасника	Брюм Ліліаня Яновна
UPN/Email	Текст	Електронна адреса	brume.l@donnu.edu.ua
User Action	Текст	Дія учасника	Joined/Left
Timestamp	Дата/час	Час події	04/07/26, 11:01:05
Role	Текст	Роль учасника	Organizer/Presenter
Duration	Текст/число	Тривалість перебування	1h 28m 39s

Крім технічних особливостей, структура звіту може відрізнятися залежно від версії Microsoft Teams, мови інтерфейсу користувача та типу зустрічі (звичайна зустріч, вебінар, запланована подія). Це означає, що парсер повинен бути адаптованим до різних варіантів файлів, згенерованих протягом кількох років.

Таким чином, аналіз структури CSV-звітів MS Teams показує, що використання стандартних бібліотек без додаткової обробки для читання таких файлів є недостатнім. Для створення програмного забезпечення для коректної обробки даних необхідна розробка спеціалізованого модуля парсингу, який

поєднує автоматичне визначення кодування, роздільника, пошук метаданих, обробку форматів дати та часу та агрегацію сесій учасників. Без такого модуля подальша агрегація даних та отримання достовірної аналітики суттєво ускладнюється [38, 6, 26].

2.3 Алгоритм парсингу CSV-файлів MS Teams

Однією з ключових складових розробленого програмного забезпечення є модуль парсингу CSV-файлів. Від коректності первинного зчитування напряду залежить якість подальшої агрегації та аналізу. Як було показано в підрозділі 1.2, звіти відвідуваності Microsoft Teams характеризуються високою варіативністю формату, що робить використання стандартних бібліотек для читання CSV-файлів недостатнім. Тому було розроблено спеціалізований алгоритм парсингу, здатний адаптуватися до різних версій платформи [40], регіональних налаштувань та мов інтерфейсу.

Алгоритм парсингу реалізовано у модулі `parser.py` і складається з декількох послідовних етапів, кожен з яких вирішує конкретну проблему, притаманну файлам MS Teams.

На першому етапі відбувається автоматичне визначення кодування файлу. Оскільки звіти можуть бути збережені в UTF-16 Little Endian, UTF-8 з BOM або звичайному UTF-8, функція `decode_file()` здійснює спробу декодування файлу послідовно у семи найбільш поширених кодуваннях [39]. Файл зчитується в бінарному режимі, після чого виконується спроба декодування. Успішне декодування без помилок та відсутність спотвореного тексту визначає правильне кодування.

Другим важливим етапом є визначення роздільника полів. Алгоритм аналізує перші 20 рядків файлу та визначає найбільш імовірний роздільник (кома, табуляція або крапка з комою). Це особливо важливо, оскільки назва зустрічі часто містить кому, що призводить до розбиття рядка на неправильну кількість полів при жорсткому використанні коми як роздільника.

Наступним кроком є пошук та витягування метаданих зустрічі. Алгоритм здійснює пострядковий аналіз файлу для виявлення цього блоку, який зазвичай

розміщується на початку файлу. Витягуються такі ключові параметри: назва зустрічі (Meeting Title), дата та час початку, загальна тривалість зустрічі та ім'я організатора. У разі відсутності явних метаданих назва зустрічі формується з імені файлу, а дата визначається за першою подією у логах.

Однією з найскладніших частин є визначення заголовка таблиці учасників та подальше зчитування даних. Алгоритм шукає рядок, який одночасно містить ключові стовпці, такі як “Full Name” (або “Name”), “UPN/Email”, “User Action”, “Timestamp”, “Role” та стовпець з тривалістю. Після виявлення заголовка відбувається визначення індексів усіх необхідних колонок. Зчитування даних учасників триває до першого порожнього рядка або кінця файлу.

Особлива увага приділяється обробці тривалості присутності. Формат цього поля у звітах MS Teams може бути різним: “1h 30m”, “1:30:00”, “90 min”, просто число хвилин або навіть відсутнім. Для розбору цих варіантів реалізована функція `parse_duration()` [40], яка використовує регулярні вирази та підтримує декілька шаблонів.

Також алгоритм реалізує агрегацію множинних сесій одного учасника вже на етапі парсингу. Якщо учасник кілька разів підключався та відключався протягом однієї зустрічі, система об'єднує ці сесії та обчислює сумарну тривалість.

Розроблений алгоритм є достатньо гнучким і стійким до варіацій [6, 40], що виникають у різних версіях Microsoft Teams та при різних мовних налаштуваннях користувача. Він забезпечує високу точність зчитування даних навіть у випадку нестандартних або частково пошкоджених файлів, що є критичним для надійної роботи всієї системи.

2.4 Алгоритм агрегації даних відвідуваності

Після успішного парсингу CSV-файлів (підрозділ 2.3) наступним критичним етапом є агрегація даних. Парсинг забезпечує коректне зчитування сирих даних, тоді як агрегація – їх осмислення, узагальнення та підготовку до аналізу. Саме на цьому етапі формується цінна аналітична інформація [1]:

індивідуальна статистика учасників, відсотки відвідуваності, зведені показники за групами та потоками.

Таблиця 2.2 – Підтримувані формати даних при парсингу CSV-файлів

Кодування	UTF-8, UTF-8 з BOM, UTF-16 LE, UTF-16 BE, Windows-1251, ISO-8859-1, CP1252
Роздільники	кома (,), табуляція (t), крапка з комою (;)
Формати дати	7 основних регіональних варіантів (MM/DD/YYYY, DD/MM/YYYY, DD.MM.YYYY тощо)
Формати тривалості	“1h 30m”, “1:30:00”, “90 min”, числове значення в хвилинах

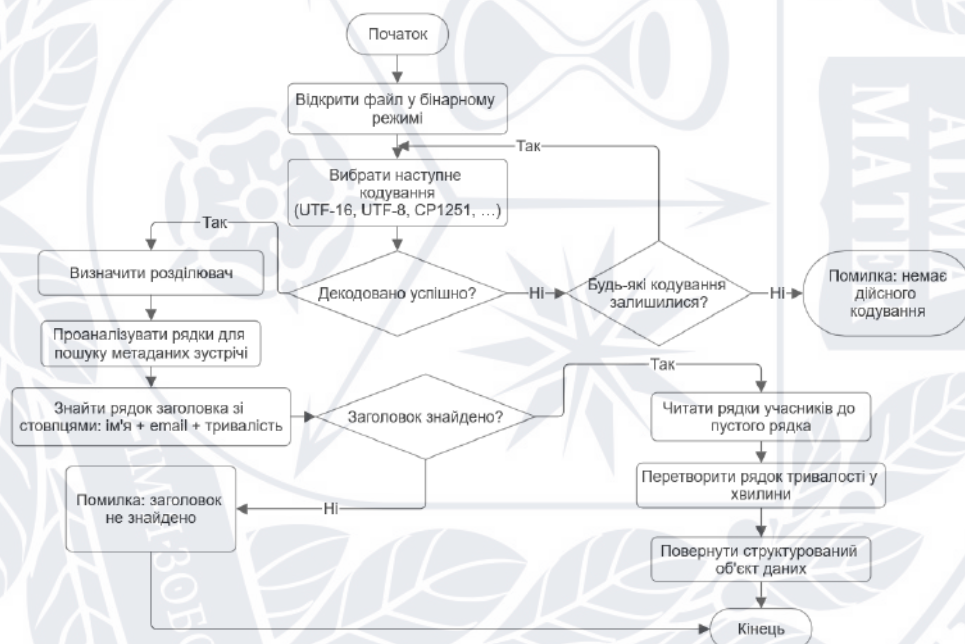


Рисунок 2.4 – Блок-схема алгоритму парсингу CSV-файлу

Алгоритм агрегації реалізовано з урахуванням специфіки освітнього процесу. Він дозволяє не просто накопичувати інформацію, а перетворювати її на зручну для прийняття рішень аналітику. Основні задачі алгоритму: об'єднання даних з різних зустрічей, розрахунок ключових метрик, категоризація рівня відвідуваності та врахування прихованих учасників.

Основні етапи алгоритму агрегації:

1. Нормалізація та збагачення даних

Після парсингу кожного файлу система додає до записів метайнформацію: тип заняття, академічну групу, потік, дату зустрічі. Це дозволяє в подальшому виконувати фільтрацію та групування за цими параметрами.

2. Агрегація по учасниках у межах однієї зустрічі

Для кожної зустрічі дані учасників групуються за комбінацією «ПІБ + Email». Якщо учасник має кілька сесій підключення, їх тривалість сумуються, а відсоток відвідуваності розраховується відносно загальної тривалості зустрічі.

3. Розрахунок ключових метрик

- Для кожного учасника на рівні окремої зустрічі обчислюється:
- Загальна тривалість присутності (у хвилинах);
- Відсоток відвідуваності

$$P = \left(\frac{D}{T}\right) \times 100 \quad (2.1)$$

Де P – відсоток відвідуваності, D – тривалість присутності учасника у хвилину, T – загальна тривалість зустрічі у хвилину [1].

- Категорія відвідуваності.

4. Зведена агрегація по всіх зустрічах

При переході на дашборд система виконує глобальну агрегацію: для кожного унікального учасника (за комбінацією ПІБ + Email) підраховується:

- Кількість зустрічей, на яких був присутній;
- Сумарний час присутності;
- Середній відсоток відвідуваності;
- Кількість зустрічей за кожною категорією.

5. Обробка прихованих учасників

Система підтримує механізм приховування учасників (наприклад, викладачів, організаторів, технічних акаунтів). Приховані учасники виключаються з загальної аналітики, але залишаються доступними для перегляду в деталях конкретної зустрічі.

Алгоритм агрегації спроектовано з урахуванням продуктивності. Для пришвидшення роботи деякі показники (наприклад, `attendance_percentage`) зберігаються в базі даних у вже обчисленому вигляді, а не розраховуються динамічно при кожному відкритті дашборду. Це рішення особливо важливе при роботі з великим обсягом даних.

Також передбачено можливість перерахунку агрегації після редагування метаінформації зустрічі або зміни статусу приховування учасника. Такий підхід забезпечує актуальність аналітики в реальному часі.

Відповідно до джерела [1], категорії рівня відвідуваності поділяються на три групи (табл. 2.3).

Таблиця 2.3 – Категорії рівня відвідуваності

Висока відвідуваність	80% і вище
Середня відвідуваність	Від 50% до 79%
Низька відвідуваність	Нижче 50%

Розроблений алгоритм агрегації даних відвідуваності дозволяє трансформувати розрізнені CSV-звіти в єдину аналітичну систему. Він забезпечує точний розрахунок ключових показників, гнучку фільтрацію, підтримку освітньої специфіки (групи, потоки, типи занять) та механізм виключення нерелевантних записів. Якісна реалізація цього алгоритму є основою для створення інформативного та зручного дашборду [41], який дає користувачеві можливість швидко оцінювати рівень залученості учасників онлайн-зустрічей.

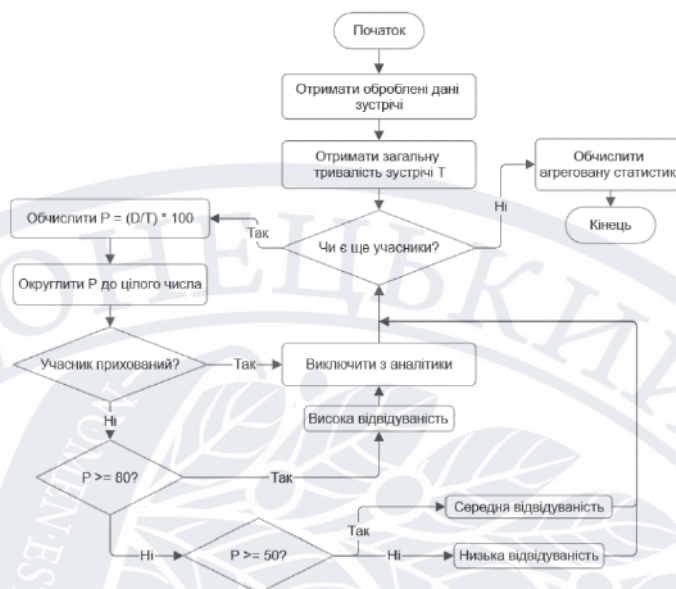


Рисунок 2.5 – Блок-схема алгоритму агрегації даних відвідуваності

2.5 Модель бази даних

Структура бази даних наведена на ER-діаграмі (рис. 2.6). База складається з восьми таблиць: `meeting_types`, `groups`, `streams`, `meetings`, `participants`, `attendance`, `tab_groups`, `tab_group_members`.

Таблиця `meeting_types` є довідником типів занять (лекція, семінар, практичне заняття тощо). Таблиці `groups` та `streams` зберігають назви академічних груп та потоків відповідно. Таблиця `meetings` містить інформацію про кожну завантажену зустріч та посиляється на довідники через зовнішні ключі. Таблиця `participants` зберігає дані про учасників, а прапор `is_hidden` дозволяє виключати окремих осіб з аналітики. Таблиця `attendance` є зв'язуючою між зустрічами та учасниками, фіксуючи тривалість присутності та відсоток відвідуваності. Таблиці `tab_groups` та `tab_group_members` забезпечують зберігання користувацьких налаштувань дашборду, дозволяючи створювати групи вкладок для швидкої фільтрації за типами занять, групами та потоками.

Зв'язки між таблицями та індексація

Між таблицями `meetings` та `attendance` встановлено зв'язок один-до-багатьох через зовнішній ключ `meeting_id`. Аналогічно, `attendance` зв'язана з `participants` через `participant_id`. Таблиці-довідники (`meeting_types`, `groups`, `streams`) зв'язані з `meetings` відповідними зовнішніми ключами. Для підвищення

продуктивності створено індекси за полями, що найчастіше використовуються у фільтрах.

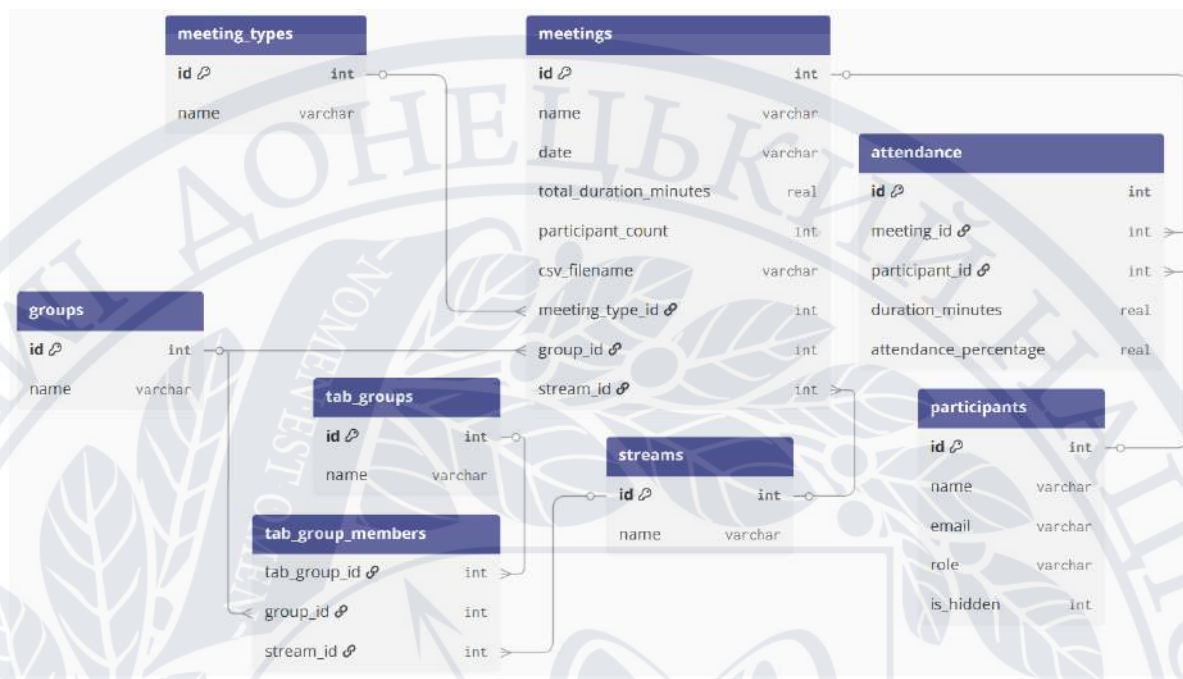


Рисунок 2.6 – ER-діаграма бази даних

Обґрунтування проєктних рішень

- Нормалізація – винесення типів занять, груп та потоків в окремі таблиці-довідники виключає дублювання даних.
- Зберігання обчислених значень – attendance_percentage зберігається в таблиці attendance, що дозволяє уникнути повторних розрахунків.
- Розширюваність – механізм міграцій (_migrate() у db.py) дозволяє оновлювати структуру БД при зміні версії програми.

2.6 Вибір технологій та інструментів розробки

Вибір технологічного стеку [32] є важливим етапом проєктування, оскільки він безпосередньо впливає на відповідність програмного забезпечення сформульованим функціональним і нефункціональним вимогам. При виборі технологій враховувалися ключові пріоритети: можливість повноцінної роботи в локальному (офлайн) режимі, мінімальний поріг входу для кінцевого користувача, простота встановлення та розгортання, а також зручність підтримки та розширення коду.

Для порівняння обраних технологій з альтернативними варіантами наведено таблицю 2.4.

Таблиця 2.4 – Порівняння обраних технологій з альтернативами

Критерій	Обране рішення (Python + Tkinter + SQLite)	Альтернатива (PyQt / Electron)	Альтернатива (PostgreSQL)
Необхідність встановлення	Ні (стандартна бібліотека)	Так	Так
Робота в офлайн-режимі	Повністю	Повністю	Потребує сервера
Простота розгортання	Дуже висока	Середня	Низька
Швидкодія для даного обсягу	Висока	Висока	Висока
Підтримка нетехнічними користувачами	Відмінна	Середня	Низька
Розмір дистрибутиву	Мінімальний	Великий	Великий

Як видно з таблиці, обраний стек технологій є оптимальним для поставлених завдань.

2.7 Проектування архітектури системи

Проектування архітектури програмного забезпечення є важливим етапом, на якому визначається загальна структура системи, взаємодія між її компонентами та принципи організації коду. Архітектура розробленого ПЗ [ID_36] побудована за модульним принципом відповідно до принципу єдиної відповідальності (Single Responsibility Principle – SRP) [30]. Це дозволяє легко підтримувати, тестувати та розширювати систему в майбутньому.

2.7.1 Загальна архітектура системи

Система має тришарову архітектуру:

1. Рівень інтерфейсу користувача (Presentation Layer)
2. Рівень бізнес-логіки (Business Logic Layer)
3. Рівень даних (Data Access Layer)

Таке розділення забезпечує чітке розмежування відповідальності, що є найкращою практикою сучасної розробки програмного забезпечення.

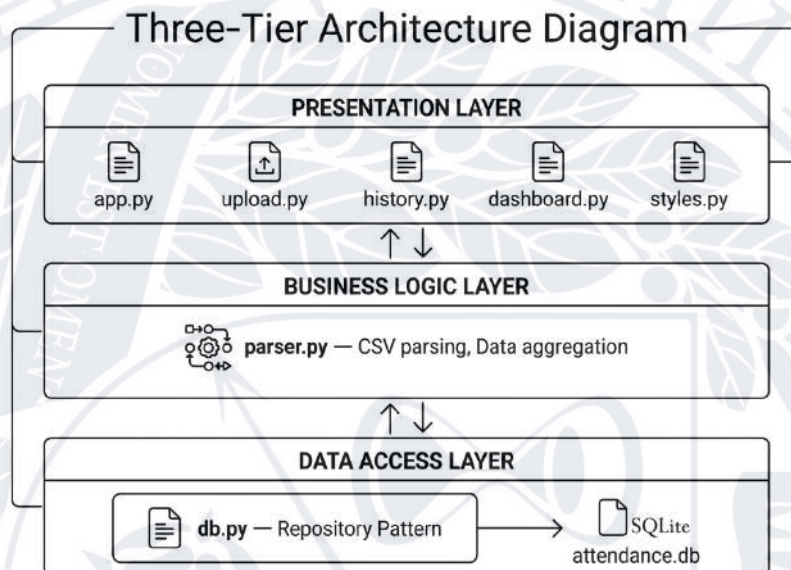


Рисунок 2.7 – Тришарова архітектура програмного забезпечення [ID_37]

2.7.2 Опис основних модулів

Рівень інтерфейсу користувача реалізований у папці `ui/` і включає такі модулі:

- `app.py` – головний клас `App`. Він керує вікном програми, бічною панеллю навігації та перемиканням між сторінками.
- `upload.py` – сторінка завантаження та попереднього перегляду CSV-файлів.
- `history.py` – сторінка історії завантажених зустрічей з можливістю редагування та видалення.
- `dashboard.py` – головна аналітична сторінка (дашборд) з фільтрами, таблицями та картками показників.
- `styles.py` – централізоване зберігання стилів, кольорів, шрифтів та допоміжних функцій для оформлення інтерфейсу.

Взаємодія між сторінками відбувається через центральний клас App, який виступає контролером. Кожна сторінка має метод `on_show()`, який викликається при переході на неї та оновлює записи [31].

Рівень бізнес-логіки представлений модулем `parser.py`. Він повністю незалежний від інших частин системи:

- Не імпортує жодних модулів проєкту;
- Приймає шлях до CSV-файлу;
- Повертає стандартизований Python-словник з даними зустрічі.

Така ізоляція робить модуль легко тестованим і придатним для повторного використання.

Рівень даних реалізований у модулі `db.py`. Він виконує роль Repository (репозиторію) і надає зручний інтерфейс для роботи з базою даних:

- Ініціалізація таблиць;
- Збереження, оновлення та видалення зустрічей;
- Управління прихованими учасниками;
- Отримання даних для дашборду (агреговані запити).

2.7.3 Принципи проєктування архітектури

Під час проєктування архітектури системи було застосовано низку архітектурних принципів, які забезпечують гнучкість, підтримуваність та тестованість програмного забезпечення. Їх реалізацію наведено в таблиці 2.5.

Крім того, при розробці окремих модулів дотримано принцип єдиної відповідальності (Single Responsibility Principle) [30]: `parser.py` відповідає лише за парсинг CSV-файлів, `db.py` – за роботу з базою даних, а модулі інтерфейсу (`upload.py`, `dashboard.py`, `history.py`) – лише за відображення та взаємодію з користувачем.

2.7.4 Точка входу програми

Точкою входу є файл `main.py`, який:

- Ініціалізує базу даних (створює таблиці, якщо вони відсутні);
- Створює екземпляр головного вікна App;

- Запускає головний цикл програми.

Така структура забезпечує чисте розділення коду та спрощує подальшу розробку.

Таблиця 2.5 – Архітектурні принципи

Принцип	Як реалізований у системі
Separation of Concerns (розділення відповідальностей)	Чітке розділення UI (Tkinter), бізнес-логіки (parser.py) та доступу до даних (db.py)
Dependency Inversion (інверсія залежностей)	Вищі рівні (UI) не залежать від деталей реалізації нижчих рівнів (БД) завдяки використанню db.py як прошарку
Modularity (модульність)	Вищі рівні не залежать від деталей реалізації нижчих рівнів
Testability (тестованість)	Більшість логіки (парсинг, робота з БД) легко тестується в ізоляції завдяки відсутності зв'язків між модулями

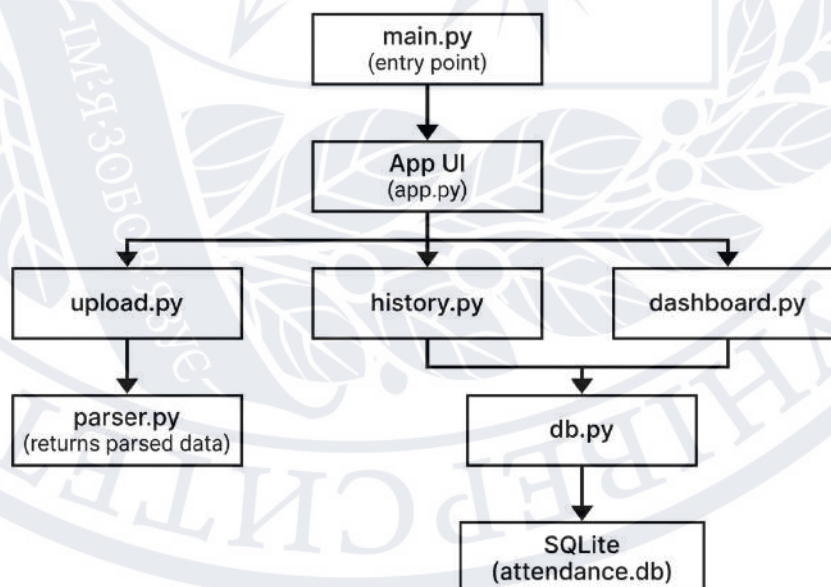


Рисунок 2.8 – Схема взаємодії модулів системи

2.7.5 Переваги обраної архітектури

- Зручність підтримки – зміна інтерфейсу не впливає на логіку парсингу та базу даних.
- Масштабованість – легко додавати нові сторінки, типи звітів або методи аналітики.
- Тестованість – більшість компонентів можна тестувати окремо.
- Простота розгортання – користувачеві достатньо запустити main.py.

У результаті проектування розроблено чітку, модульну та масштабовану архітектуру системи, побудовану за принципами сучасної розробки програмного забезпечення. Тришарова структура з чітким розмежуванням рівнів, використання патерну Repository та модульний підхід забезпечують високу підтримуваність, надійність та можливість подальшого розвитку програмного продукту. Обрана архітектура повністю відповідає сформульованим вимогам і створює надійну основу для практичної реалізації додатку, яка описана в третьому розділі [26].

Висновки до другого розділу

Другий розділ кваліфікаційної роботи присвячено визначенню вимог, проектуванню основних алгоритмів, моделі бази даних та архітектури програмного забезпечення. У ході виконання цього розділу було вирішено низку важливих завдань, необхідних для подальшої успішної реалізації системи.

На етапі визначення вимог (підрозділ 2.1) проведено детальний аналіз функціональних, нефункціональних та системних вимог. Було чітко сформульовано, що система повинна забезпечувати повну автоматизацію процесу обробки CSV-звітів MS Teams, підтримувати освітню специфіку (типи занять, академічні групи, потоки), мати інтуїтивний інтерфейс для нетехнічних користувачів та працювати в повністю локальному режимі. Визначено кількісні показники продуктивності (обробка до 300 файлів, час відгуку дашборду 2-3 секунди) та апаратні вимоги (Python 3.8+, 4 ГБ ОЗУ, підтримка Windows, Linux

та macOS). Такий комплексний підхід дозволив закласти надійне підґрунтя для всіх наступних проєктних рішень.

У підрозділі 2.3 розроблено алгоритм парсингу CSV-файлів, який враховує всі особливості звітів Microsoft Teams, виявлені в першому розділі: варіативність кодування, роздільників, форматів дати та тривалості, а також відсутність чіткого роздільника між метаданими та таблицею учасників. Запропонований алгоритм є гнучким, стійким до помилок і забезпечує високу точність зчитування даних. Його реалізація у вигляді окремого модуля `parser.py` дозволяє легко тестувати та підтримувати цю частину системи.

У підрозділі 2.4 детально описано алгоритм агрегації даних відвідуваності. Запропонований алгоритм дозволяє не лише накопичувати дані, але й трансформувати їх у цінну аналітичну інформацію: розрахунок відсотків відвідуваності за формулою $P = \left(\frac{D}{T}\right) \times 100$, категоризацію учасників (висока, середня, низька відвідуваність), зведену статистику по всіх зустрічах. Особлива увага приділена продуктивності – запропоновано зберігати попередньо обчислені значення в базі даних, а також підтримці механізму приховування нерелевантних учасників.

У підрозділі 2.5 спроектовано модель бази даних (SQLite) з восьми таблиць, яка забезпечує нормалізацію даних через довідники типів занять, груп та потоків, а також підтримку користувацьких налаштувань дашборду через таблиці `tab_groups` та `tab_group_members`.

Підрозділ 2.6 містить обґрунтування вибору технологій. Було обрано стек на основі стандартної бібліотеки Python (Tkinter + sqlite3), що забезпечує простоту встановлення, відсутність зовнішніх залежностей, портативність та зручність використання для нетехнічних користувачів. Проведено порівняльний аналіз з альтернативними технологіями (PyQt/Electron та PostgreSQL), який підтвердив оптимальність обраного рішення для поставлених задач.

У підрозділі 2.7 представлено проєктування архітектури системи. Система побудована за тришаровою модульною архітектурою з чітким розмежуванням

рівнів (інтерфейс, бізнес-логіка, дані). Використання патерну Repository, принципу єдиної відповідальності (SRP) та модульної організації коду значно підвищує підтримуваність, тестованість та масштабованість програмного продукту. Описано точку входу програми (main.py) та перелічено переваги обраної архітектури.

Загалом, у другому розділі повністю завершено етап проєктування системи. Сформульовані вимоги, розроблені алгоритми, спроектована модель даних та архітектура створюють цілісне, логічне та технічно обґрунтоване підґрунтя для практичної реалізації програмного забезпечення, яка викладена в третьому розділі.

Отримані результати підтверджують, що обрані підходи та рішення відповідають актуальним потребам користувачів і дозволять створити ефективний, зручний та надійний інструмент для агрегації та аналізу відвідуваності онлайн-зустрічей у MS Teams [31].

РОЗДІЛ 3

РЕАЛІЗАЦІЯ ДОДАТКУ

3.1 Реалізація модуля парсингу CSV-файлів

Модуль парсингу є одним з найбільш критичних компонентів розробленого програмного забезпечення, оскільки від його надійності безпосередньо залежить точність усієї подальшої агрегації та аналітики. Модуль реалізований у файлі `parser.py` відповідно до алгоритму, описаного в підрозділі 2.2.

Основною функцією модуля є `parse_teams_csv()`, яка виконує повний цикл обробки вхідного CSV-файлу. Крім неї, важливу роль відіграють допоміжні функції `decode_file()` та `parse_duration()` [40, 39].

Основні етапи роботи модуля:

1. Автоматичне визначення кодування
Функція `decode_file()` зчитує файл у бінарному режимі та послідовно намагається декодувати його у семи найбільш поширених кодуваннях (`utf-16`, `utf-16-le`, `utf-8-sig`, `utf-8`, `cp1251` тощо). Такий підхід дозволяє коректно обробляти файли, згенеровані в різних регіональних версіях Microsoft Teams.
2. Визначення роздільника полів
Алгоритм аналізує перші рядки файлу та автоматично визначає роздільник (кома або табуляція).
3. Витягування метаданих зустрічі
Здійснюється пошук рядків, що містять “Meeting Title”, “Start Time”, “Meeting Duration”. У разі відсутності метаданих назва зустрічі формується з імені файлу, а дата – з першої події або поточної дати.
4. Пошук заголовка таблиці учасників
Алгоритм гнучко шукає рядок заголовка за наявністю ключових колонок (`Name`, `Email`, `In-Meeting Duration` / `Duration`). Це дозволяє працювати з різними варіантами звітів MS Teams.
5. Парсинг тривалості

Функція `parse_duration()` підтримує різні формати запису тривалості (“1h 30m”, “1:30:00”, “90 min” тощо) за допомогою регулярних виразів.

6. Агрегація сесій учасників

Якщо учасник кілька разів підключався/відключався протягом однієї зустрічі, його сесії автоматично об’єднуються.

Модуль спроектовано максимально незалежним: він не залежить від бази даних чи інтерфейсу і повертає стандартизований словник Python. Це значно полегшує тестування та підтримку коду.

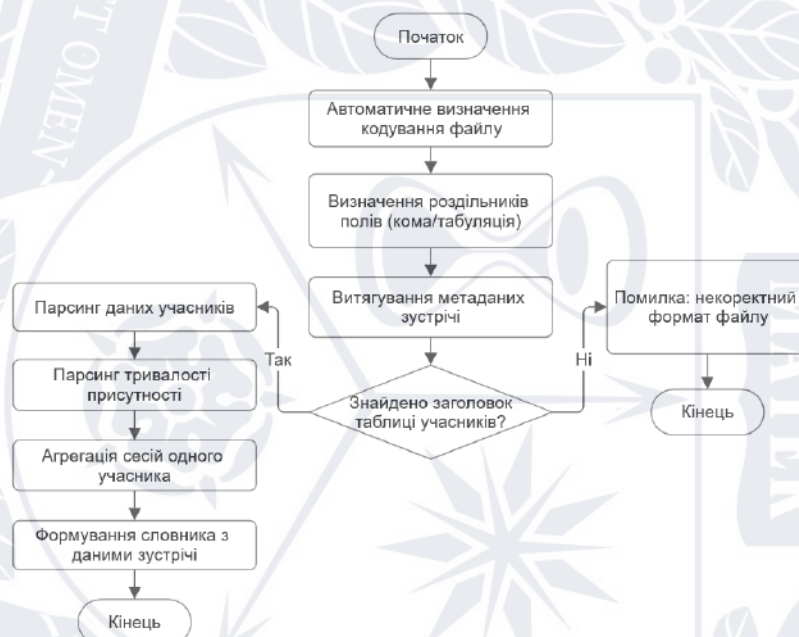


Рисунок 3.1 – Блок-схема роботи модуля парсингу CSV-файлів

Таблиця 3.1 – Підтримувані формати даних модулем парсингу

Параметр	Підтримувані варіанти
Кодування	UTF-8, UTF-16 LE, UTF-8 BOM, Windows-1251 та ін.
Роздільники	«,» (кома), “\t” (слеш т), «;» (крапка з комою)
Формати дати	7 основних регіональних форматів
Формати тривалості	“1h 30m”, “1:30:00”, “90 min”, числове значення

Реалізований модуль парсингу успішно вирішує всі проблеми, виявлені під час аналізу структури CSV-звітів Microsoft Teams. Його гнучкість і стійкість до варіацій формату забезпечують надійну первинну обробку даних, що є основою функціонування всієї системи [26].

3.1.1 Обґрунтування вибору основних технологій

Мова програмування – Python

Основною мовою програмування обрано Python [42, 29]. Цей вибір обумовлений кількома вагомими причинами:

- Простота та читабельність синтаксису, що прискорює розробку та спрощує подальшу підтримку коду.
- Широка стандартна бібліотека, яка дозволяє вирішувати більшість задач без використання сторонніх пакетів.
- Відмінна підтримка роботи з текстовими даними, регулярними виразами та базами даних.
- Висока популярність мови в освітньому середовищі, що полегшує можливе подальше доопрацювання системи іншими розробниками.

Графічний інтерфейс користувача – Tkinter

Для створення десктопного інтерфейсу обрано бібліотеку Tkinter [44], яка входить до складу стандартної бібліотеки Python. Переваги такого вибору:

- Не потребує встановлення додаткових залежностей.
- Дозволяє створювати крос-платформні графічні додатки.
- Підтримує всі необхідні елементи: вікна, кнопки, таблиці (Treeview), вкладки, контекстні меню тощо.
- Має достатню гнучкість для створення функціонального та зручного інтерфейсу.

База даних – SQLite

Для зберігання даних обрано SQLite через модуль sqlite3 [43]. Основні причини:

- Вбудована в Python, не потребує окремого серверного ПЗ.

- Вся база даних зберігається в одному файлі (attendance.db) [46], що забезпечує максимальну портативність.
- Підтримка транзакцій, індексів та складних SQL-запитів.
- Висока швидкодія при роботі з обсягами даних, характерними для одного факультету або кафедри.

Обробка CSV-файлів

Для парсингу CSV використовуються лише вбудовані модулі:

- csv – для читання табличних даних;
- re – для обробки тривалості та складних рядків;
- io (також ручний перебір для визначення кодування файлу).

Таке рішення повністю відповідає вимозі відсутності зовнішніх залежностей.

3.1.2 Переваги обраного технологічного стеку

- Відсутність зовнішніх залежностей – програма запускається на будь-якому комп'ютері з встановленим Python 3.8+ без додаткового встановлення бібліотек.
- Портативність – достатньо скопіювати папку з програмою та файл бази даних для перенесення на інший комп'ютер.
- Конфіденційність – всі дані залишаються на локальному комп'ютері користувача.
- Мінімальні системні вимоги – програма працює на пристроях з 4 ГБ оперативної пам'яті та 300 МБ вільного місця на диску, що дозволяє використовувати її на широкому спектрі обладнання.

Обраний технологічний стек забезпечує виконання вимог, сформульованих у підрозділі 2.1: роботу в офлайн-режимі, відсутність зовнішніх залежностей та локальне зберігання даних. Використання стандартної бібліотеки Python дозволяє запускати програму на будь-якому комп'ютері з Python 3.8+ без додаткового налаштування.

3.2 Реалізація інтерфейсу користувача

Інтерфейс користувача є важливою складовою розробленого програмного забезпечення, оскільки саме від його зручності та інтуїтивності залежить готовність кінцевих користувачів (викладачів, кураторів, методистів) регулярно працювати з програмою.

Інтерфейс реалізований як сучасний десктопний додаток з використанням бібліотеки Tkinter (стандартна бібліотека Python) [44]. Загальна архітектура побудована за принципом односторінкового застосування з бічною навігаційною панеллю. Головне вікно програми керує перемиканням між основними сторінками.

3.2.1 Загальна структура інтерфейсу

Головний файл `app.py` містить клас `App`, який відповідає за:

- Ініціалізацію головного вікна (`borderless` режим з кастомною верхньою панеллю);
- Створення бічної навігаційної панелі (`rail`);
- Управління переходами між сторінками;
- Систему сповіщень та обробку закриття програми.

Кожна сторінка реалізована в окремому модулі та успадковує `tk.Frame`. При переході на сторінку викликається метод `on_show()`, який забезпечує актуалізацію даних [31].

3.2.2 Сторінка завантаження CSV-файлів (`upload.py`)

Це стартова сторінка програми. Її основні можливості включають:

- Вибір одного або кількох CSV-файлів;
- Автоматичний парсинг обраних файлів за допомогою модуля `parser.py`;
- Відображення метаданих зустрічі (назва, дата, тривалість, кількість учасників);
- Можливість вказати тип заняття, академічну групу та потік;

- Таблицю попереднього перегляду учасників з можливістю приховування окремих записів;
- Контекстне меню (права кнопка миші) для швидкого приховування/відображення учасників.

Організатори зустрічей автоматично позначаються як приховані.

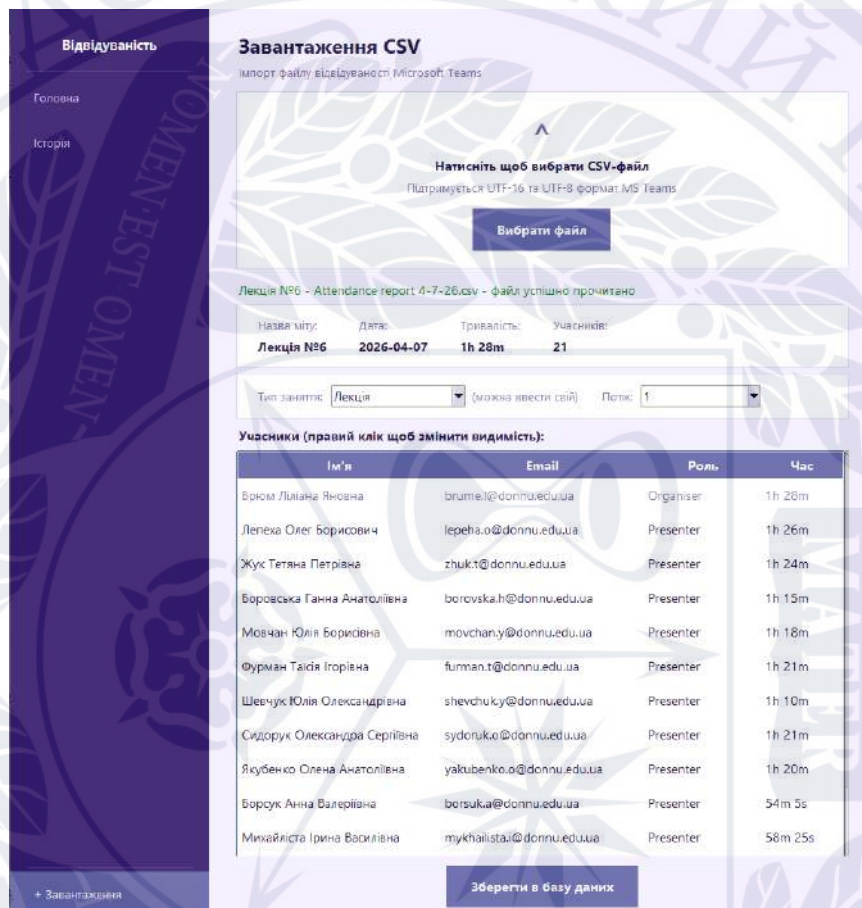


Рисунок 3.2 – Сторінка завантаження CSV-файлів

3.2.3 Сторінка дашборду (аналітики) (dashboard.py)

Це головна робоча сторінка програми. Вона містить:

- Динамічну панель вкладок (автоматично генерується за потоками, типами занять та групами);
- Можливість групування вкладок (drag-and-drop) [48];
- Інформаційні картки з ключовими показниками (кількість мітів, унікальних учасників, середня відвідуваність);
- Дві основні вкладки: «Учасники» та «Зустрічі»;

- Кольорове кодування відсотка відвідуваності (зелений – висока, жовтий – середня, червоний – низька);
- Фільтрацію по групах та зустрічах;
- Функцію експорту поточних даних у CSV.

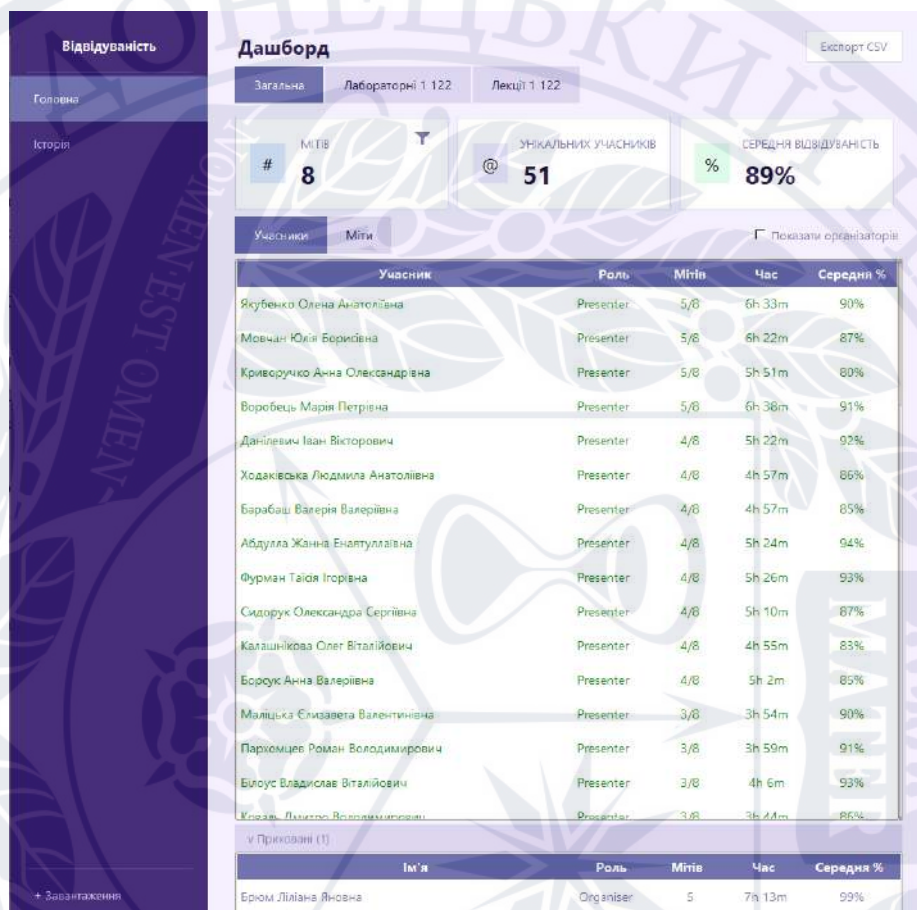


Рисунок 3.3 – Головна сторінка програми

3.2.4 Сторінка історії зустрічей (history.py)

Сторінка призначена для управління збереженими даними і надає такі можливості:

- Таблицю всіх завантажених зустрічей із сортуванням;
- Редагування метаданих (тип заняття, група, потік);
- Пошук та фільтрацію;
- Множинне видалення зустрічей.

3.2.5 Загальні рішення щодо інтерфейсу

- Єдиний файл стилів styles.py (кольорова схема, шрифти, стилі Treeview);

- Адаптивне розміщення елементів;
- Інформативні повідомлення про успішні операції та помилки;
- Повна підтримка української мови;
- Користувачські контекстні меню та інтерактивні елементи.

#	Назва міту	Дата	Тип	Група	Потік	Учасників	Файл CSV
1	Лекція №10	2026-05-05	Лекція			20	Лекція №10 - A
2	Лекція №9	2026-04-28	Лекція			23	Лекція №9 - At
3	Лекція №8	2026-04-21	Лекція			21	Лекція №8 - At
4	Лекція №7	2026-04-14	Лекція			19	Лекція №7 - At
5	Лабораторна Б	2026-04-12	Лабораторна	Б		8	Лабораторна Б
6	Лабораторна В	2026-04-11	Лабораторна	В		7	Лабораторна В
7	Лабораторна А	2026-04-10	Лабораторна	А		8	Лабораторна А
8	Лекція №6	2026-04-07	Лекція			21	Лекція №6 - At

Рисунок 3.4 – Сторінка історії

Реалізований інтерфейс користувача повністю відповідає вимогам, сформульованим у розділі 2. Він є інтуїтивним, функціонально насиченим та орієнтованим на нетехнічного користувача. Використання сучасних підходів (динамічні вкладки, групування, drag-and-drop, кольорове кодування) у поєднанні з чистим Tkinter дозволило створити зручний та ергономічний інструмент для щоденної роботи [48].

3.3 Реалізація модуля аналітики

Модуль аналітики є центральною частиною програмного забезпечення, оскільки саме він перетворює накопичені сирі дані в зручну, наочну та корисну

аналітику. Модуль реалізований переважно у файлі dashboard.py [44] і тісно взаємодіє з модулем доступу до даних (db.py).

3.3.1 Загальна архітектура модуля

Модуль аналітики виконує такі основні задачі:

- Отримання даних з бази даних з урахуванням поточних фільтрів;
- Обчислення зведених показників;
- Формування та відображення таблиць «Учасники» та «Зустрічі»;
- Застосування кольорового кодування рівня відвідуваності;
- Підтримка динамічних вкладок і групування [49];
- Підготовка даних для експорту.

3.3.2 Система фільтрації даних

Дашборд підтримує багаторівневу фільтрацію:

- Динамічні вкладки – автоматично генеруються за потоками, типами занять та групами;
- Групування вкладок – користувач може об'єднувати кілька вкладок у групи за допомогою drag-and-drop;
- Фільтр по зустрічах – можливість вибору конкретних зустрічей у межах вкладки.

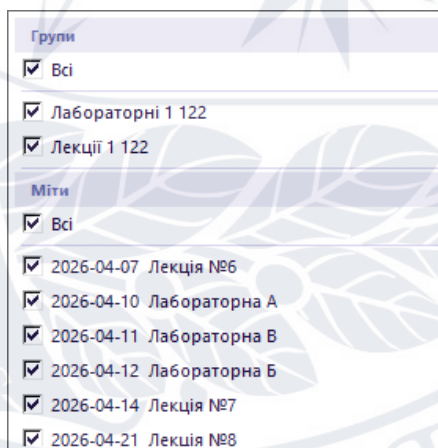


Рисунок 3.5 – Фільтрація мітів

Фільтрація виконується на рівні Python після отримання даних з бази, що забезпечує гнучкість і швидкодію.

3.3.3 Формування зведених показників

У верхній частині дашборду відображаються три інформаційні картки:

- Кількість зустрічей;
- Кількість унікальних учасників;
- Середній відсоток відвідуваності.

Показники динамічно перераховуються при зміні фільтрів.

3.3.4 Таблиця «Учасники»

Таблиця є основним інструментом аналізу. Для кожного унікального учасника (комбінація ПІБ + Email) відображається:

- Кількість зустрічей, на яких був присутній;
- Загальний час присутності;
- Середній відсоток відвідуваності;
- Кольорове кодування (зелений / жовтий / червоний).

Учасники сортуються за кількістю відвіданих зустрічей у спадному порядку [49, 50]. Підтримується контекстне меню для швидкого приховування/відображення учасників.

3.3.5 Таблиця «Зустрічі»

Таблиця відображає список усіх зустрічей з урахуванням активних фільтрів. Для кожної зустрічі показується:

- Назва та дата проведення;
- Тип заняття, група, потік;
- Кількість учасників;
- Середній відсоток відвідуваності.

Подвійний клік по рядку відкриває модальне вікно з детальною інформацією про зустріч та повним списком її учасників.

3.3.6 Функціонал експорту

Кнопка «Експорт CSV» дозволяє зберегти поточні дані таблиці учасників (з урахуванням усіх фільтрів) у файл з зрозумілими заголовками.

Реалізований модуль аналітики повністю відповідає функціональним і нефункціональним вимогам, сформульованим у розділі 2. Поєднання динамічної системи вкладок, гнучкої фільтрації, кольорового кодування та зручного експорту даних робить модуль потужним і комфортним інструментом для щоденного аналізу відвідуваності. Використання ефективних алгоритмів агрегації та оптимізованих запитів до бази даних забезпечує високу швидкодію навіть при значному обсязі даних [41].

3.4 Тестування програмного забезпечення

Тестування є обов'язковим етапом розробки програмного забезпечення, метою якого є перевірка відповідності реалізованого продукту вимогам, сформульованим у другому розділі, а також оцінка його стабільності, надійності та зручності використання.

Тестування проводилося за методом «чорна скриня» [51, 52] на реальних даних, експортованих з Microsoft Teams протягом березня–травня 2026 року. Загальна кількість тестових файлів – 47 CSV-файлів різного обсягу, кодування, форматів дати та тривалості.

3.4.1 Методика тестування

- Функціональне тестування основних модулів;
- Тестування на граничних значеннях (порожні файли, пошкоджені файли, великі файли);
- Регресійне тестування після внесення змін;
- Тестування інтерфейсу (usability);
- Тестування продуктивності.

Тестування здійснювалося в середовищі Windows 10, Python 3.11.

3.4.2 Результати тестування

Таблиця 3.2 – Результати тестування модуля парсингу

№	Тестовий випадок	Кількість тестів	Примітки
1	Різні кодування файлів (UTF-16, UTF-8)	12	Успішно
2	Різні формати дати	9	Успішно
3	Різні формати тривалості	11	Успішно
4	Файли без метаданих	5	Успішно
5	Пошкоджені та порожні файли	4	Коректна обробка помилок

Таблиця 3.3 – Результати тестування модуля збереження даних

№	Тестовий випадок	Результат
1	Збереження зустрічі з повною метаінформацією	Пройдено
2	Редагування типу заняття, групи, потоку	Пройдено
3	Приховування/відновлення учасників	Пройдено
4	Видалення зустрічі	Пройдено
5	Автоматичне приховування організаторів	Пройдено

Таблиця 3.4 – Результати тестування модуля аналітики

№	Тестовий випадок	Результат
1	Фільтрація за потоком та групою	Пройдено
2	Кольорове кодування відвідуваності	Пройдено
3	Групування вкладок (drag-and-drop)	Пройдено
4	Експорт даних у CSV	Пройдено
5	Відкриття детальної інформації про зустріч	Пройдено
6	Продуктивність при більш ніж 300 зустрічах	Пройдено

3.4.3 Загальні результати тестування

Під час тестування було перевірено 41 тестових сценарії. Усі критичні функції працюють стабільно. Виявлені незначні недоліки інтерфейсу (відображення дуже довгих назв зустрічей, масштабування на екранах з високою роздільною здатністю) були виправлені на фінальному етапі розробки.

Проведене комплексне функціональне тестування підтвердило, що розроблене програмне забезпечення повністю відповідає вимогам, сформульованим у розділі 2. Програма стабільно працює з різними типами CSV-файлів Microsoft Teams, забезпечує коректне збереження та обробку даних, а також надає зручний і функціональний інтерфейс. Отримані результати дозволяють зробити висновок про готовність програмного продукту до практичного використання в навчальних закладах та організаціях [26].

3.5 Системні вимоги до програмного забезпечення

Після завершення розробки та тестування програмного забезпечення було визначено мінімальні технічні вимоги до середовища його експлуатації. Програма розроблена з використанням стандартної бібліотеки Python, що забезпечує її кросплатформеність.

У таблиці 3.5 наведено основні системні вимоги.

Таблиця 3.5 – Системні вимоги до програмного забезпечення

Вимога	Значення
Операційна система	Windows 10/11, Linux (Ubuntu 20.04+), macOS 11+
Python	Версія 3.8 або новіша
Оперативна пам'ять	від 2 ГБ (рекомендовано 4 ГБ)
Вільне місце на диску	від 300 МБ
Підключення до Інтернету	не потрібне (офлайн-режим)

Програмне забезпечення працює в автономному режимі, не потребує активного підключення до мережі Інтернет та не передає жодних даних на

зовнішні сервери. Усі обчислення виконуються локально на пристрої користувача, що гарантує конфіденційність персональних даних відповідно до вимог чинного законодавства [36].

3.6 Перспективи розвитку

Розроблене програмне забезпечення є повністю функціональним і готовим до використання в освітніх установах та організаціях. Однак у процесі роботи було визначено кілька напрямків для подальшого вдосконалення системи.

Перспективним напрямком є розширення підтримки інших форматів вхідних даних. Наразі програма орієнтована виключно на CSV-звіти Microsoft Teams. Додавання можливості імпортувати дані з інших джерел (наприклад, Excel-файлів або безпосередньо через API) значно розширило б аудиторію користувачів.

Другим важливим напрямком є інтеграція з хмарними сервісами. Локальне зберігання даних забезпечує конфіденційність, але ускладнює спільний доступ до аналітики для кількох викладачів або адміністраторів. Створення веб-версії додатку з централізованою базою даних дозволило б формувати звіти для цілих факультетів або інститутів.

Третім напрямком є вдосконалення аналітичних можливостей. Можна додати автоматичне прогнозування відвідуваності на основі історичних даних, виявлення аномалій (різке падіння відвідуваності в окремих студентів) та автоматичне формування сповіщень для кураторів або викладачів.

Крім того, доцільним є розширення експортних форматів. Окрім CSV, користувачі могли б отримувати звіти у форматах PDF (для друку) або Excel (з попередньо налаштованим форматуванням). Це спростило б подання звітів керівництву або в навчальні відділи.

Реалізація зазначених напрямків дозволить перетворити розроблений інструмент на універсальне рішення для контролю відвідуваності в закладах освіти різного рівня.

Висновки до третього розділу

У третьому розділі кваліфікаційної роботи розглянуто практичну реалізацію програмного забезпечення для агрегації та аналізу відвідуваності онлайн-зустрічей у MS Teams.

Була успішно реалізована модульна архітектура системи, що включає:

- Модуль парсингу CSV-файлів (`parser.py`), який забезпечує надійну обробку файлів з різними кодуваннями, форматами дати та тривалості;
- Інтерфейс користувача (`app.py`, `upload.py`, `dashboard.py`, `history.py`), орієнтований на нетехнічного користувача;
- Модуль аналітики, що підтримує динамічну фільтрацію, кольорове кодування, групування вкладок та експорт даних;
- Модуль роботи з базою даних (`db.py`), що забезпечує ефективне зберігання та швидке отримання інформації.

Особливу увагу було приділено зручності використання: інтуїтивна навігація, контекстні меню, динамічні вкладки, можливість групування даних та кольорове кодування показників відвідуваності. Програмне забезпечення повністю працює в офлайн-режимі та не вимагає встановлення додаткових залежностей.

Проведене функціональне тестування на реальних даних (47 CSV-файлів) підтвердило високу надійність та стабільність роботи всіх модулів. Усі ключові сценарії використання були успішно пройдені.

Розроблений програмний продукт повністю відповідає вимогам, сформульованим у другому розділі, і готовий до практичного застосування в навчальних закладах та організаціях, які активно використовують платформу Microsoft Teams для проведення онлайн-зустрічей.

Отримані результати реалізації свідчать про успішне досягнення поставленої мети кваліфікаційної роботи та створення програмного інструменту, який суттєво спрощує контроль відвідуваності [26].

ВИСНОВОК

У кваліфікаційній роботі вирішено актуальне науково-практичне завдання – розроблено програмне забезпечення для автоматизації процесів агрегації та аналізу даних відвідуваності онлайн-зустрічей у MS Teams на основі CSV-звітів.

У першому розділі роботи проведено аналіз предметної області та існуючих систем. Розглянуто платформу Microsoft Teams, її можливості для відстеження відвідуваності та виявлено основні обмеження, зокрема 30-денний термін зберігання звітів та відсутність засобів для автоматичної агрегації даних з багатьох зустрічей. Виконано порівняльний аналіз існуючих інструментів обробки даних (Excel, MS Access, Power BI, Viva Insights), який показав, що жоден з них не забезпечує оптимального поєднання простоти використання, автоматизації та підтримки освітньої специфіки. Результати аналізу стали підґрунтям для формування вимог до власного програмного забезпечення.

У другому розділі визначено функціональні та нефункціональні вимоги до системи. Розроблено алгоритм парсингу CSV-файлів MS Teams, який враховує варіативність кодувань, роздільників, форматів дати та тривалості. Запропоновано алгоритм агрегації даних відвідуваності з розрахунком ключових метрик (відсоток присутності, категорія відвідуваності). Спроектовано модель реляційної бази даних (SQLite) з восьми таблиць, що забезпечує нормалізацію даних та підтримку освітньої специфіки. Обґрунтовано вибір технологічного стеку на основі стандартної бібліотеки Python (Tkinter, sqlite3). Розроблено тришарову модульну архітектуру системи (рівень інтерфейсу, бізнес-логіки та доступу до даних), яка забезпечує підтримуваність та масштабованість продукту.

У третьому розділі реалізовано основні модулі програмного забезпечення. Створено модуль парсингу CSV-файлів (parser.py), який підтримує різні кодування та формати даних. Розроблено графічний інтерфейс користувача з використанням бібліотеки Tkinter, що включає сторінку завантаження файлів, сторінку історії зустрічей та головний дашборд з динамічними вкладками та кольоровим кодуванням показників відвідуваності. Реалізовано модуль аналітики з системою фільтрації за групами, потоками та типом занять, а також

функцію експорту даних у CSV-формат. Проведено функціональне тестування на 47 реальних CSV-файлах, яке підтвердило стабільну роботу всіх модулів та відповідність продукту визначеним вимогам.

Таким чином, поставлену мету роботи досягнуто в повному обсязі. Розроблене програмне забезпечення може бути впроваджене в навчальних закладах та організаціях, які використовують Microsoft Teams, для автоматизованого контролю відвідуваності онлайн-заходів. Перспективами подальших досліджень є розширення підтримки інших форматів вхідних даних, створення веб-версії додатку для спільної роботи кількох викладачів та впровадження методів прогнозування відвідуваності на основі історичних даних.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ha Y. et al. Effects of class attendance on academic performance: Evidence from synchronous online courses. // Computers & Education. 2023. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0738059323002286> (дата звернення: 30.04.2026).
2. Microsoft Teams grows to 320 million monthly active users. // Microsoft Community Hub. 2023. URL: <https://techcommunity.microsoft.com/discussions/microsoftteams/teams-grows-to-320-million-monthly-active-users/3964746> (дата звернення: 24.04.2026).
3. Microsoft Teams Statistics & Usage Report (2025–2026). // The VoIP Shop. 2026. URL: <https://www.thevoipshop.co.uk/blog/microsoft-teams-statistics-usage-adoption> (дата звернення: 24.04.2026).
4. Microsoft Teams – Features overview. // Microsoft Learn. URL: <https://learn.microsoft.com/en-us/microsoftteams/teams-overview> (дата звернення: 03.05.2026).
5. Introduction to Moodle LMS – Teams. // Microsoft Learn. URL: <https://learn.microsoft.com/en-us/microsoftteams/platform/resources/moodle-overview> (дата звернення: 02.05.2026).
6. Manage meeting attendance reports in Microsoft Teams. // Microsoft Support. URL: <https://support.microsoft.com/en-us/office/manage-meeting-attendance-reports-in-microsoft-teams-ae7cf170-530c-47d3-84c1-3aedac74d310> (дата звернення: 24.04.2026).
7. Teams Meeting Attendance Report Available. // Office365ITPros. 2021. URL: <https://office365itpros.com/2021/06/11/teams-meeting-attendance-report/> (дата звернення: 29.04.2026).
8. How to get attendance report from teams. // Microsoft Q&A. 2025. URL: <https://learn.microsoft.com/en-us/answers/questions/5686724/how-to-get-attendance-report-from-teams> (дата звернення: 24.04.2026).

9. Teams Meeting Attendance Reports Get 1-Year Retention Policy. // Office365ITPros. 2025. URL: <https://office365itpros.com/2025/03/11/attendance-report-retention-policy/> (дата звернення: 29.04.2026).
10. Антонов Ю.С. Оцінка повноти відповідей в автоматизованих системах контролю знань. // Наукові праці Донецького національного технічного університету. Серія: Інформатика, кібернетика та обчислювальна техніка. 2012. Вип. 15. С. 113-117.
11. Гнатюк М.А., Антонов Ю.С. Розробка системи тестування знань для платформи Android. // Прикладні інформаційні технології. 2021. С. 38-40.
12. Чуприна І.А., Антонов Ю.С. Особливості розробки системи автоматичного створення та валідації тестових питань на основі взаємодії з ChatGPT. // Комп'ютерні технології обробки даних. 2025. С. 66-69.
13. Штовба С.Д., Панкевич О.Д., Філіпов М.О. Тестовий дистанційний контроль знань студентів з урахуванням самооцінки впевненості. // Вісник Вінницького політехнічного інституту. 2011. С. 200-203.
14. Римар П.В., Войтко Б.С. Розробка автоматизованої системи формування розкладу навчальних занять з використанням 1С: «Підприємство». // Вісник Хмельницького національного університету. Серія «Технічні науки». 2021. № 5. С. 36-40.
15. Щербак В.В., Римар П.В. Розробка телеграм боту для отримання розкладу занять. // Прикладні інформаційні технології. 2025. С. 131-133.
16. Римар П.В., Слободянюк К.О. Розробка системи формування розкладу навчальних занять: користувачі, права доступу та розмежування даних. // Прикладні аспекти сучасних міждисциплінарних досліджень. 2021. С. 114-116.
17. Захарова К.В., Ніколюк П.К. Програмний додаток «Деканат». // Прикладні інформаційні технології. 2021. С. 38-40.

- 18.Радзіховська А.О., Антонов Ю.С. Ризик-менеджмент під час розробки додатка для вибору наукового керівника. // Прикладні інформаційні технології. 2025. С. 83-85.
- 19.Штовба С.Д., Петричко М.В. Об'єктивізація експертизи наукових творів: автоматичний підбір опонентів дисертацій. // Міжнародна науково-практична конференція «Відновлення України у повоєнні ...». 2024.
- 20.Штовба С.Д., Петричко М.В. Алгоритми для призначення зовнішніх рецензентів для захисту PhD-дисертацій. // System research and information technologies. 2024. С. 127-145.
- 21.Ротштейн О.П., Кашканов А.А., Зелінська О.В., Кательніков Д.І. Fuzzy scenario analysis of generalized dynamic systems. // Cybernetics and Systems Analysis. 2025.
- 22.About Power Query in Excel. // Microsoft Support. URL: <https://support.microsoft.com/en-us/office/about-power-query-in-excel-7104fbee-9e62-4cb9-a02e-5bfb1a6c536a> (дата звернення: 03.05.2026).
- 23.Microsoft Teams Admin Center. // Microsoft. URL: <https://admin.teams.microsoft.com/> (дата звернення: 03.05.2026).
- 24.Microsoft Power BI documentation. // Microsoft Learn. URL: <https://learn.microsoft.com/en-us/power-bi/> (дата звернення: 29.04.2026).
- 25.Attendee exclusion rules for Viva Insights. // Microsoft Learn. URL: <https://learn.microsoft.com/en-us/viva/insights/use/attendee-exclusion-rules> (дата звернення: 30.04.2026).
- 26.Pressman R.S., Maxim B. Software Engineering: A Practitioner's Approach. 8th ed. McGraw-Hill, 2014. 976 с.
- 27.АНТОНОВ Ю. С., СМОКТИЙ К. В. Expert systems using for answers analysis in automated knowledge control systems. // Herald of Khmelnytskyi National University. Technical Sciences. 2024. Vol. 339, № 4. P. 323–331. DOI: 10.31891/2307-5732-2024-339-4-51.
- 28.Silberschatz A., Korth H.F., Sudarshan S. Database System Concepts. 7th ed. McGraw-Hill, 2020. 1376 с.

29. Connolly T., Begg C. Database Systems: A Practical Approach to Design, Implementation, and Management. 7th ed. Pearson, 2021. 1440 с.
30. Martin R.C. Clean Architecture: A Craftsman's Guide to Software Structure and Design. Prentice Hall, 2017. 432 с.
31. Sommerville I. Software Engineering. 10th ed. Pearson, 2015. 816 с.
32. Martin R.C. Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall, 2008. 431 с.
33. IEEE Std 830-1998. IEEE Recommended Practice for Software Requirements Specifications. IEEE, 1998. URL: <https://ieeexplore.ieee.org/document/720574> (дата звернення: 29.04.2026).
34. Python csv – CSV File Reading and Writing. // Python Software Foundation. URL: <https://docs.python.org/3/library/csv.html> (дата звернення: 29.04.2026).
35. Python Standard Library. // Python Software Foundation. URL: <https://docs.python.org/3/library/index.html> (дата звернення: 29.04.2026).
36. Закон України «Про захист персональних даних» від 01.06.2010 № 2297-VI. URL: <https://zakon.rada.gov.ua/laws/show/2297-17> (дата звернення: 30.04.2026).
37. Allen G., Owens M. The Definitive Guide to SQLite. 2nd ed. Apress, 2010. 368 с.
38. Shafranovich Y. RFC 4180: Common Format and MIME Type for Comma-Separated Values (CSV) Files. // IETF. 2005. URL: <https://www.rfc-editor.org/rfc/rfc4180.html> (дата звернення: 29.04.2026).
39. Lutz M. Programming Python. 4th ed. O'Reilly Media, 2010. 1584 с.
40. Python re – Regular expression operations. // Python Software Foundation. URL: <https://docs.python.org/3/library/re.html> (дата звернення: 29.04.2026).
41. Few S. Information Dashboard Design: The Effective Visual Communication of Data. 2nd ed. Analytics Press, 2013. 280 с.
42. Python Success Stories. // Python Software Foundation. URL: <https://www.python.org/success-stories/> (дата звернення: 24.04.2026).

43. Python sqlite3 – DB-API 2.0 interface for SQLite databases. // Python Software Foundation. URL: <https://docs.python.org/3/library/sqlite3.html> (дата звернення: 30.04.2026).
44. Python tkinter – Python interface to Tcl/Tk. // Python Software Foundation. URL: <https://docs.python.org/3/library/tkinter.html> (дата звернення: 24.04.2026).
45. Gaffney K.P. et al. SQLite: Past, Present, and Future. // Proceedings of the VLDB Endowment. 2022. Vol. 15, № 12. URL: <https://dl.acm.org/doi/10.14778/3554821.3554842> (дата звернення: 30.04.2026).
46. Python os – Miscellaneous operating system interfaces. // Python Software Foundation. URL: <https://docs.python.org/3/library/os.html> (дата звернення: 03.05.2026).
47. Python – Comparison of GUI frameworks. // Python Wiki. URL: <https://wiki.python.org/moin/GuiProgramming> (дата звернення: 24.04.2026).
48. Python tkinter – DnD support. // Python Software Foundation. URL: <https://docs.python.org/3/library/tkinter.dnd.html> (дата звернення: 02.05.2026).
49. Munzner T. Visualization Analysis and Design. CRC Press, 2014. 404 с.
50. Knuth D.E. The Art of Computer Programming, Vol. 3: Sorting and Searching. 2nd ed. Addison-Wesley, 1998. 780 с.
51. Beizer B. Black-Box Testing: Techniques for Functional Testing of Software and Systems. Wiley, 1995. 305 с.
52. IEEE Std 829-2008. IEEE Standard for Software and System Test Documentation. IEEE, 2008. URL: <https://ieeexplore.ieee.org/document/4578383> (дата звернення: 03.05.2026).