

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Розробка інтелектуальної системи
персоналізованої генерації мотиваційних цитат
на основі машинного навчання»

на здобуття освітнього ступеня бакалавра
зі спеціальності 121 Інженерія програмного забезпечення
освітньо-професійної програми «Інженерія програмного забезпечення»

*Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання на
відповідне джерело*

(підпис)

Ніколай ПРОЦАК

Виконав: здобувач вищої освіти групи ПД-42

Ніколай ПРОЦАК

Керівник:

Тимур ДОВЖЕНКО

канд.техн.наук

Рецензент:

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**
Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти Бакалавр

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

_____ Ірина ЗАМРІЙ

« ____ » _____ 2026 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Процаку Ніколаю Михайловичу

1. Тема кваліфікаційної роботи: «Розробка інтелектуальної системи персоналізованої генерації мотиваційних цитат на основі машинного навчання»

керівник кваліфікаційної роботи канд. техн. наук Тимур ДОВЖЕНКО,
затверджені наказом Державного університету інформаційно-комунікаційних
технологій від «20» лютого 2026 року № 51.

2. Строк подання кваліфікаційної роботи «29» травня 2026 року.

3. Вихідні дані до кваліфікаційної роботи: теоретичні відомості про методи обробки природньої мови, опис методів вилучення інформації з тексту та формування тематичних словників, технічна документація з описом бібліотек для обробки природньої мови.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити).

1. Аналіз предметної області персоналізованої генерації мотиваційних текстів та сучасних підходів NLP.

2. Дослідження існуючих веб-сервісів, мобільних застосунків та генеративних платформ для створення цитат.

3. Обґрунтування стеку технологій розробки.

4. Розроблення логічної моделі даних для зберігання профілів користувачів, тематик, цитат, оцінок і параметрів генерації.
5. Проектування архітектури програмної системи з урахуванням веб-контролю, API-взаємодії та ML-компонентів.
6. Реалізація системи генерації та персоналізації цитат.
7. Тестування якості генерації, персоналізації, продуктивності та зручності використання.

5. Перелік графічного матеріалу: *презентація*

1. Аналіз аналогів.
2. Вимоги до програмного забезпечення.
3. Програмні засоби реалізації.
4. Діаграма послідовності, діяльності.
5. Діаграма компонентного розгортання.
6. Тестування системи.
7. Апробація результатів дослідження.

6. Дата видачі завдання «23» лютого 2026 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Підбір та аналіз науково-технічної літератури	23.02-28.02.2026	
2	Аналіз та дослідження існуючих аналогів	01.03-07.03.2026	
3	Огляд існуючих алгоритмів обробки тексту та методів і технологій побудови тематичного словника	08.03-12.03.2026	
4	Проектування застосунку для автоматизованої побудови тематичного словника навчальної дисципліни	13.03-26.03.2026	
5	Програмна реалізація застосунку	27.03-15.04.2026	
6	Тестування застосунку	16.04-23.04.2026	
7	Оформлення роботи: вступ, висновки, реферат	24.04-30.04.2026	
8	Розробка демонстраційних матеріалів	01.05-10.05.2026	
9	Попередній захист роботи	11.05-22.05.2026	

Здобувач вищої освіти

(підпис)

Ніколай ПРОЦАК

Керівник

кваліфікаційної роботи

(підпис)

Тимур ДОВЖЕНКО

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 62 стор., 14 табл., 27 рис., 31 джерел.

Мета роботи – покращення рівня психоемоційної підтримки та мотивації користувачів шляхом розроблення інтелектуальної системи персоналізованої генерації мотиваційних цитат на основі машинного навчання.

Об'єкт дослідження – процеси генерації мотиваційних цитат.

Предмет дослідження – технології генерації текстового контенту, алгоритми оброблення природної мови, моделі формування мотиваційних повідомлень та програмні засоби створення системи генерації цитат на основі аналізу текстових даних.

Короткий зміст роботи: В роботі проаналізовано предметну область та специфіку генерації мотиваційних цитат. Досліджено існуючі веб-сервіси для створення цитат та мобільні застосунки. Обрано DRF для захисту користувацьких даних. Розроблено логічну модель даних профілів користувачів, тематик, бази цитат та параметрів персоналізації та програмно реалізовані ключові функціональні можливості, зокрема: формування мотиваційних цитат на основі заданої тематики, додавання обраних цитат до персонального списку користувача, пошук цитат за ключовими словами або категоріями, оцінювання цитат користувачем для формування рекомендацій. Проведено функціональне та модульне тестування системи. В роботі використано бібліотеку Pathlib для керування файловою системою, PyQt6 для побудови користувацького інтерфейсу, Sqlite3 для надійного зберігання даних. Сферою використання застосунку є формування персоналізованого контенту для користувачів.

КЛЮЧОВІ СЛОВА: DFR, МОДУЛЬНЕ ТЕСТУВАННЯ, PYQT6, ВЕБ-СЕРВІСИ, ПЕРСОНАЛІЗОВАНИЙ КОНТЕНТ, SQLITE3.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	6
ВСТУП	8
1.1 Опис предметної області.....	10
1.2 Аналіз існуючих рішень	12
1.3 Вибір технологій та інструментальних засобів реалізації системи	19
1.4 Аналіз вимог розроблюваної системи	21
1.5 Постановка завдання	24
1.6 Висновки до першого розділу	27
2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	28
2.1 Логічна модель даних у вигляді ER-діаграми.....	28
2.2 Моделювання класів та взаємодії компонентів системи	30
2.3 Проєктування компонентної архітектури системи	34
2.4 Пакетна організація програмного забезпечення системи.....	36
2.5 Висновки до другого розділу.....	38
3 ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	39
3.1 Моделювання предметної області.....	39
3.2 Архітектура системи та проєктування функціоналу результатів дослідження	42
3.3 Опис функціональних модулів системи	47
3.5 Висновки до третього розділу	49
4 ТЕСТУВАННЯ ТА ОЦІНЮВАННЯ ЕФЕКТИВНОСТІ СИСТЕМИ	50

4.1 План тестування програмних модулів та методика оцінювання результатів	50
4.2 Тестування інтелектуальної системи персоналізованої генерації мотиваційних цитат.....	51
4.3 Результати тестування та аналіз ефективності системи	62
4.4 Розгортання системи та склад інсталяційного пакета.....	64
4.5 Висновки до четвертого розділу	66
ВИСНОВКИ.....	67
ПЕРЕЛІК ПОСИЛАНЬ.....	69
ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ.....	72
ДОДАТОК Б. ЛІСТИНГ ПРОГРАМНИХ МОДУЛІВ	80

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

1. **API** - програмний інтерфейс взаємодії клієнтської частини, серверних сервісів і ML-модуля.
2. **BERT** - трансформерна модель для контекстного подання тексту.
3. **BLEU** - метрика оцінювання подібності згенерованого тексту до еталонного корпусу.
4. **CPU** - центральний процесор обчислювального вузла.
5. **CSV** - формат табличного експорту датасетів, журналів і результатів тестування.
6. **DL** - глибинне навчання для побудови генеративних та класифікаційних моделей.
7. **DB** - база даних програмної системи.
8. **DQI** - індекс якості текстового корпусу та користувацьких анкет.
9. **GPT** - клас авторегресійних мовних моделей для створення тексту.
10. **JWT** - токен безпечної авторизації користувача у веб-застосунку.
11. **FNN** - нейронна мережа прямого поширення для допоміжної класифікації профілів.
12. **HTML** - мова розмітки веб-сторінок користувацького інтерфейсу.
13. **HTTP(S)** - протокол захищеної взаємодії між браузером і сервером.
14. **ML** - машинне навчання для персоналізації змісту мотиваційних повідомлень.
15. **JSON** - формат структурованих запитів, відповідей API і налаштувань профілю.
16. **K-means** - метод кластеризації користувацьких інтересів і поведінкових сценаріїв.
17. **LSTM** - рекурентна архітектура для аналізу послідовностей взаємодій користувача.
18. **NLP** - оброблення природної мови.

19. **MAE** - середня абсолютна похибка прогнозування релевантності цитати.
20. **ORM** - технологія відображення класів застосунку в таблиці бази даних.
21. **PCA** - метод зниження розмірності векторних ознак користувацького профілю.
22. **PDF** - формат збереження звітів про результати генерації та тестування.
23. **QRI** - інтегральний індекс відповідності цитати профілю, настрою та меті користувача.
24. **RAM** - оперативна пам'ять сервера або робочої станції.
25. **REST** - архітектурний стиль побудови прикладного API.
26. **RMSE** - середньоквадратична похибка прогнозованої моделі релевантності.
27. **ROUGE** - метрика зіставлення змістового покриття тексту.

ВСТУП

Актуальність теми зумовлена зростанням потреби у цифрових інструментах психологічної підтримки, самоорганізації та щоденної мотивації. У більшості популярних сервісів мотиваційні вислови подаються як статичні добірки без урахування цілей, емоційного стану, попередніх реакцій та особистих уподобань користувача. Через це повідомлення часто сприймаються як шаблонні, не відповідають конкретній ситуації та швидко втрачають практичну цінність. Використання методів машинного навчання та оброблення природної мови дає змогу перейти від випадкового показу цитат до персоналізованої генерації коротких мотивувальних текстів, які адаптуються до контексту користувача, його стилю сприйняття і поставленої мети [1].

Мета дослідження – покращення рівня психоемоційної підтримки та мотивації користувачів шляхом розроблення інтелектуальної системи персоналізованої генерації мотиваційних цитат на основі машинного навчання.

Для досягнення мети необхідно поєднати веб-інтерфейс, серверну логіку, реляційну базу даних, модуль NLP-аналізу та генеративну модель. Система повинна підтримувати реєстрацію користувачів, збереження вподобань, вибір тематичного напрямку, генерацію цитат, виставлення оцінок, накопичення історії взаємодій та повторне навчання або донастроювання механізмів персоналізації.

У рамках поставленої мети передбачено розв’язання таких завдань:

- проаналізувати предметну область персоналізованої генерації мотиваційних текстів та сучасні підходи NLP;
- дослідити існуючі веб-сервіси, мобільні застосунки та генеративні платформи для створення цитат;
- обґрунтувати стек технологій розробки;
- розробити логічну модель даних для зберігання профілів користувачів, тематик, цитат, оцінок і параметрів генерації;

- спроектувати архітектуру програмної системи з урахуванням веб-контролю, API-взаємодії та ML-компонентів;
- реалізувати систему генерації та персоналізації цитат;
- провести тестування якості генерації, персоналізації, продуктивності та зручності використання;

Об'єкт дослідження – процеси генерації мотиваційних цитат.

Предмет дослідження – технології генерації текстового контенту, алгоритми оброблення природної мови, моделі формування мотиваційних повідомлень та програмні засоби створення системи генерації цитат на основі аналізу текстових даних.

Для виконання поставлених завдань використано методи системного аналізу, об'єктно-орієнтованого проєктування, реляційного моделювання даних, класифікації текстів, векторизації мовних ознак, кластеризації профілів, рекомендаційних алгоритмів та оцінювання якості генеративних моделей. Програмна частина орієнтована на стек Python, FastAPI, PostgreSQL і SQLite, React, REST API, бібліотеки scikit-learn, Transformers, Pandas і засоби контейнеризації Docker.

Наукова новизна роботи полягає у поєднанні генеративної мовної моделі з контуром персоналізації, де під час створення цитати враховуються не лише тема запиту, а й історія оцінок, ціль користувача та поточний емоційний стан. Застосовано інтегральний показник QRI, що кількісно описує відповідність згенерованого вислову користувацькому профілю.

Практична цінність роботи полягає у можливості використання розробленої системи як веб-сервісу для особистої мотивації, мобільних додатків саморозвитку та цифрових щоденників. Система забезпечує швидке створення унікальних повідомлень, накопичення статистики реакцій і гнучке налаштування мотиваційного контенту під різні групи користувачів.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОБҐРУНТУВАННЯ ТЕХНІЧНИХ РІШЕНЬ

1.1 Опис предметної області

Інтелектуальна система персоналізованої генерації мотиваційних цитат призначена для створення коротких змістовних повідомлень, що відповідають індивідуальному профілю користувача, його поточному запиту, емоційному стану та обраній життєвій ситуації. Предметна область охоплює процеси збору користувацьких вподобань, аналізу текстового контексту, добору стилю вислову, генерації повідомлення та збереження реакцій на отриманий результат.

На рисунку 1.1 наведено узагальнену структуру системи, яка демонструє взаємодію між веб-інтерфейсом, сервером, базою даних і ML-модулем.

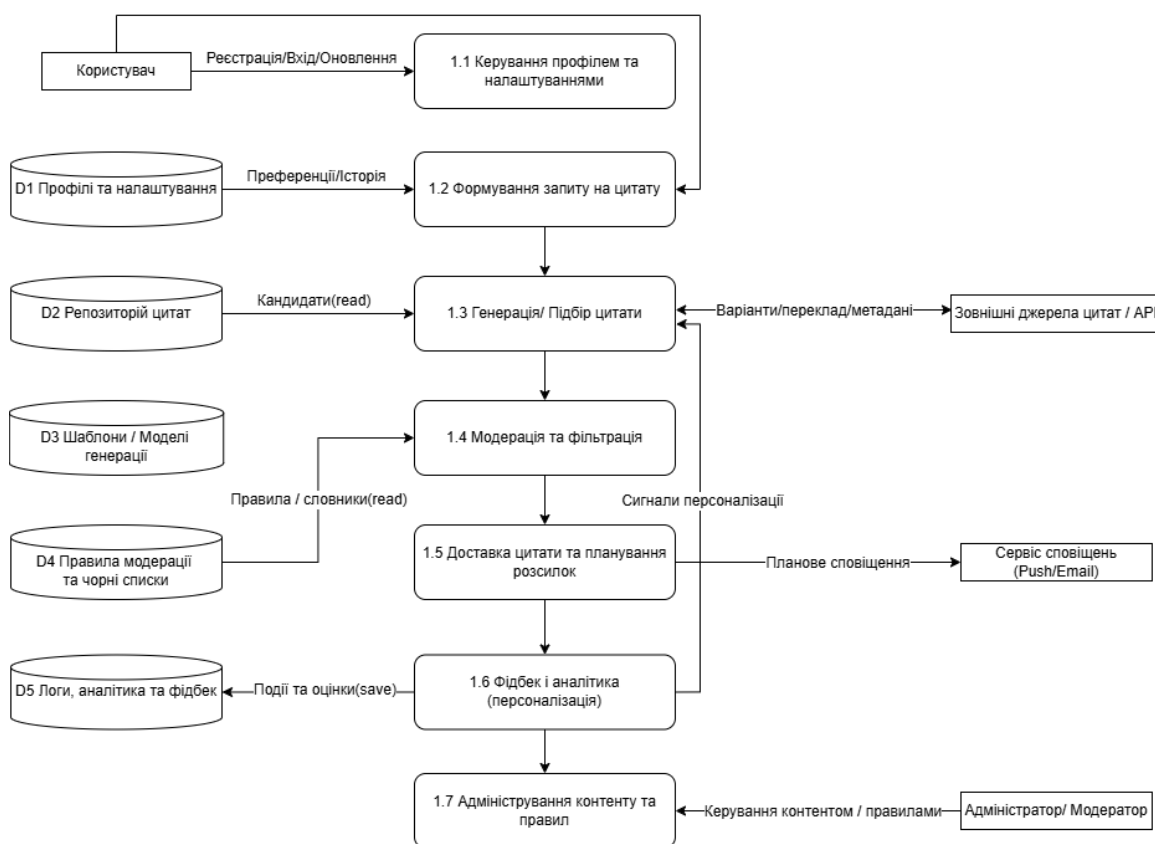


Рис. 1.1 Структура інтелектуальної системи персоналізованої генерації мотиваційних цитат

Оснoву системи становить програмна інфраструктура, до складу якої входять модуль автентифікації, профіль користувача, сховище тематик і шаблонів, підсистема генерації тексту, модуль ранжування та блок зворотного зв'язку. Користувач задає мету, настрій, бажану тональність і тематичну сферу, після чого сервер формує запит до моделі та повертає персоналізований результат через веб-інтерфейс.

Далі сформована цитата проходить етап оцінювання: система перевіряє відповідність темі, унікальність формулювання, довжину, емоційну тональність та очікуваний мотиваційний ефект. Результати аналізу використовуються для оновлення профілю користувача, що дає змогу поступово підвищувати точність майбутніх рекомендацій.

Основні параметри, які використовуються системою для персоналізації, наведено у таблиці 1.1. Вони описують інформаційні ознаки, що впливають на побудову запиту до генеративної моделі, ранжування варіантів і вибір фінального повідомлення.

Таблиця 1.1

Основні параметри персоналізації в системі

№	Параметр	Джерело / тип даних	Формат	Період оновлення
1	Настрій користувача	Анкета або швидкий вибір	категорія	за кожним запитом
2	Ціль мотивації	Профіль користувача	текст / тег	під час зміни профілю
3	Тематика цитати	Каталог тем	ідентифікатор теми	за кожним запитом
4	Історія оцінок	FeedbackService	числова оцінка	після взаємодії
5	Бажана тональність	Налаштування генерації	категорія тону	за кожним запитом
6	Індекс QRI	Модуль метрик	0..1	після генерації

Предметна область системи визначає безперервний цикл: збір профілю → аналіз контексту → генерація → ранжування → відображення → оцінювання

користувачем. Така логіка забезпечує неодноразове створення випадкового вислову, а адаптивну взаємодію, у якій кожна нова оцінка підвищує якість наступних результатів.

1.2 Аналіз існуючих рішень

Для визначення функціональних можливостей майбутньої системи генерації мотиваційних цитат доцільно проаналізувати існуючі програмні рішення, які вже реалізують подібні або суміжні функції. Такий аналіз дає змогу встановити типові підходи до побудови інтерфейсу, способи введення запиту користувачем, принципи генерації текстового результату, можливості оформлення цитат у вигляді зображень, а також недоліки, які можуть бути враховані під час розроблення власної системи.

У межах аналізу розглянуто п'ять рішень: InspiroBot, Base44, QuillBot AI Inspirational Quote Generator, QuoteGenerator.io та QuotesMaker. Зовнішній вигляд цих сервісів подано на рисунках 1.2–1.6. Вони відрізняються за призначенням: одні безпосередньо генерують мотиваційні цитати, інші надають інструменти для створення візуального оформлення цитат або розроблення власних веб-застосунків. У більшості з розглянутих сервісів спостерігається фокус на стандартні сценарії використання, коли користувач отримує готову або випадкову цитату без урахування контексту запиту.

До того ж інтерфейс обмежується полем введення тексту та кнопкою генерації, без ніяких інших взаємодій та налаштування генерації контенту. Також обмежені або взагалі відсутні можливості врахування параметрів, як настроїв, попередня активність.

Сервіс InspiroBot (див. рис. 1.2) є одним із найпростіших прикладів генератора мотиваційних висловів. Його інтерфейс побудований навколо однієї основної дії – натискання кнопки Generate. Після цього система автоматично формує цитату, яка поєднується з фоновим зображенням. Перевагою такого рішення є мінімальна кількість дій з боку користувача: не потрібно вводити тему,

налаштовувати стиль або вибирати формат результату. Водночас це обмежує гнучкість системи, оскільки користувач фактично не керує тематикою, настроєм, довжиною та змістом згенерованої цитати. Для майбутньої системи доцільно використати простоту взаємодії, але доповнити її можливістю введення теми та вибору категорії.

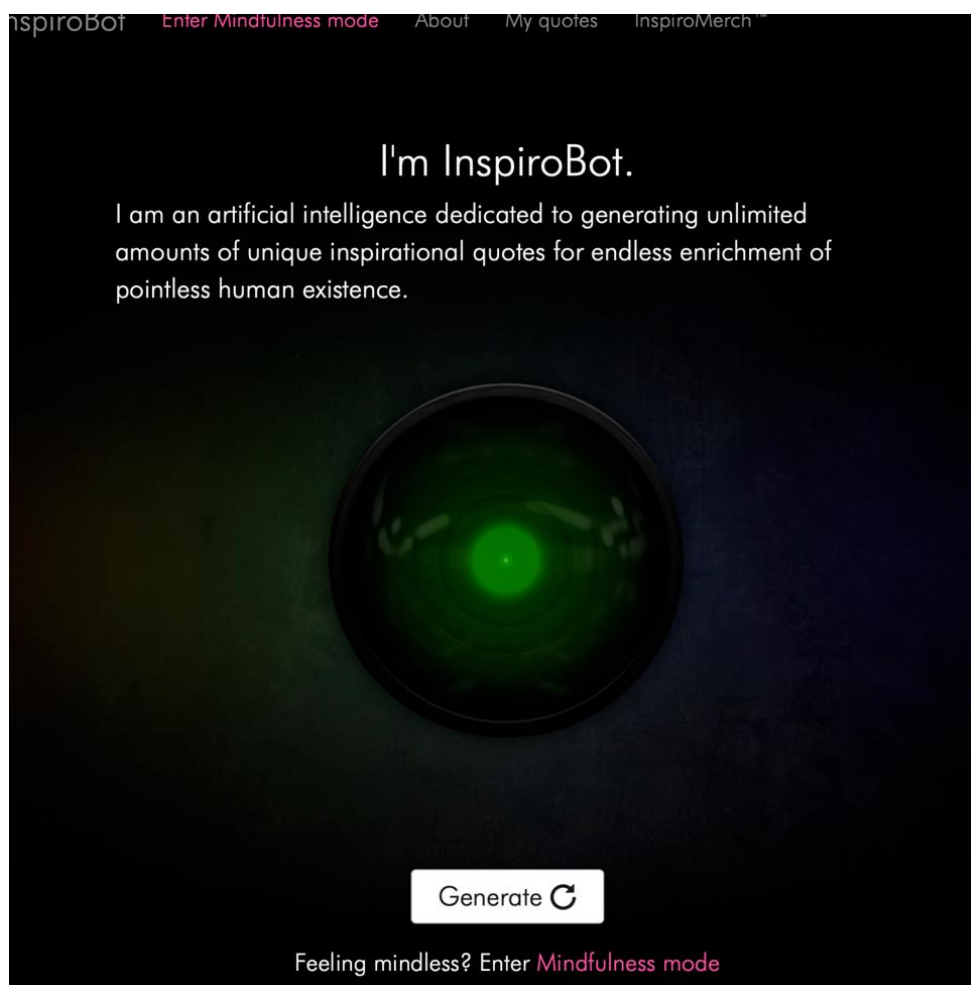


Рис. 1.2 Інтерфейс сервісу InspiroBot

Платформа Base44 (див. рис. 1.3) не є вузькоспеціалізованим генератором цитат, однак її можна розглядати як суміжне рішення, оскільки вона дозволяє створювати вебзастосунки на основі текстового опису. Інтерфейс сервісу орієнтований на введення користувацького запиту, наприклад опису майбутнього застосунку, після чого система формує базову структуру цифрового продукту. Для теми генерації мотиваційних цитат це рішення є важливим як

приклад no-code/AI-платформи, за допомогою якої можна швидко створити прототип вебзастосунку. Недоліком у контексті досліджуваної задачі є те, що Base44 не спеціалізується саме на генерації цитат, тому не містить готової логіки категоризації, стилізації та збереження мотиваційних висловів.

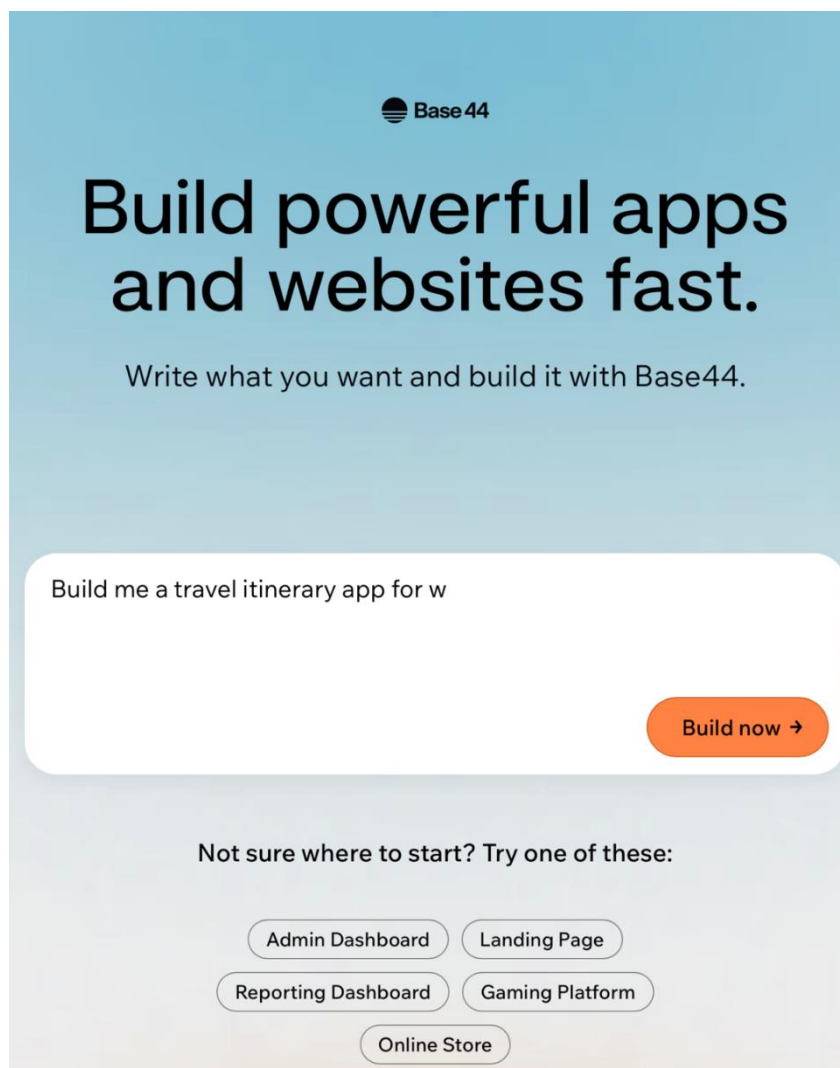


Рис. 1.3 Інтерфейс платформи Base44

QuillBot AI Inspirational Quote Generator (див. рис. 1.4) є спеціалізованим інструментом для створення натхненних цитат на основі текстового запиту. Користувач може ввести тему або короткий опис потрібного вислову, після чого система генерує відповідний текст. Перевагою цього рішення є орієнтація на змістову якість цитати та підтримка конкретного контексту, наприклад командної роботи, лідерства, стійкості або навчання. Крім того, сервіс пропонує

прикладі тем, що спрощує роботу для користувачів, які не мають чітко сформульованого запиту. Недоліком є обмежена візуальна складова: основний акцент зроблено на тексті, а не на створенні готового графічного матеріалу для соціальних мереж або презентацій.

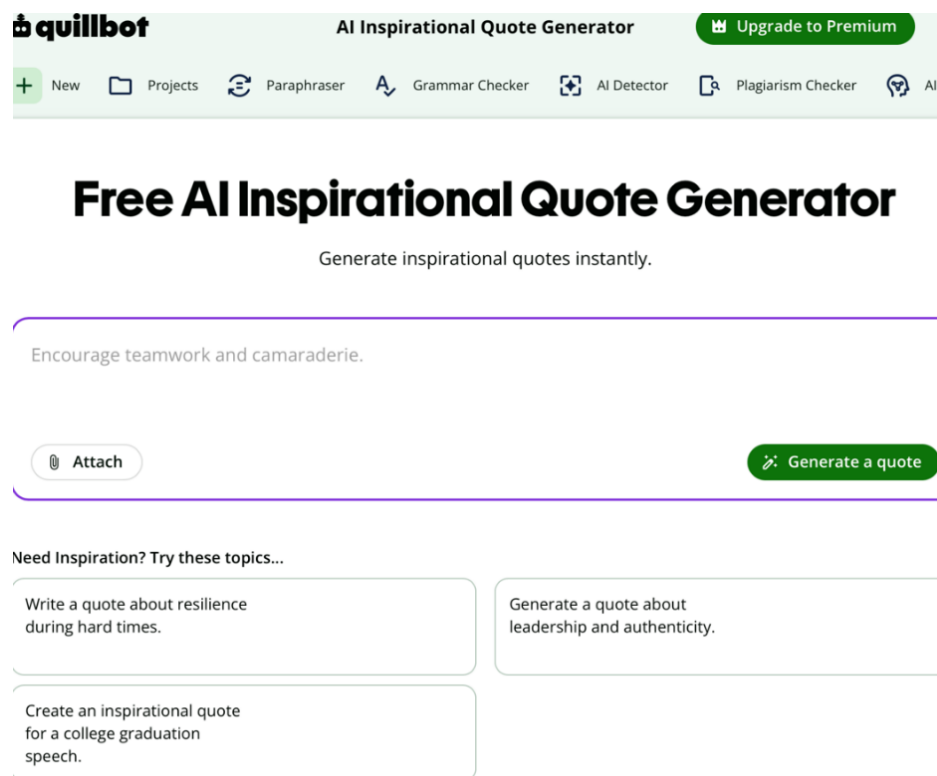


Рис. 1.4 Інтерфейс QuillBot AI Inspirational Quote Generator

QuoteGenerator.io (див. рис. 1.5) поєднує генерацію цитат із вибором тематичних категорій. На головній сторінці користувач може ввести власну тему або скористатися готовими категоріями, зокрема Motivational, Success & Hard Work, Happiness & Positivity. Це підвищує зручність використання, оскільки система не вимагає від користувача точного формулювання запиту. Також важливою особливістю є відображення кількості доступних запитів, що вказує на наявність певних обмежень використання сервісу. Для майбутньої системи такий підхід є корисним у частині категоризації цитат, але доцільно уникати надмірних обмежень для базових функцій.

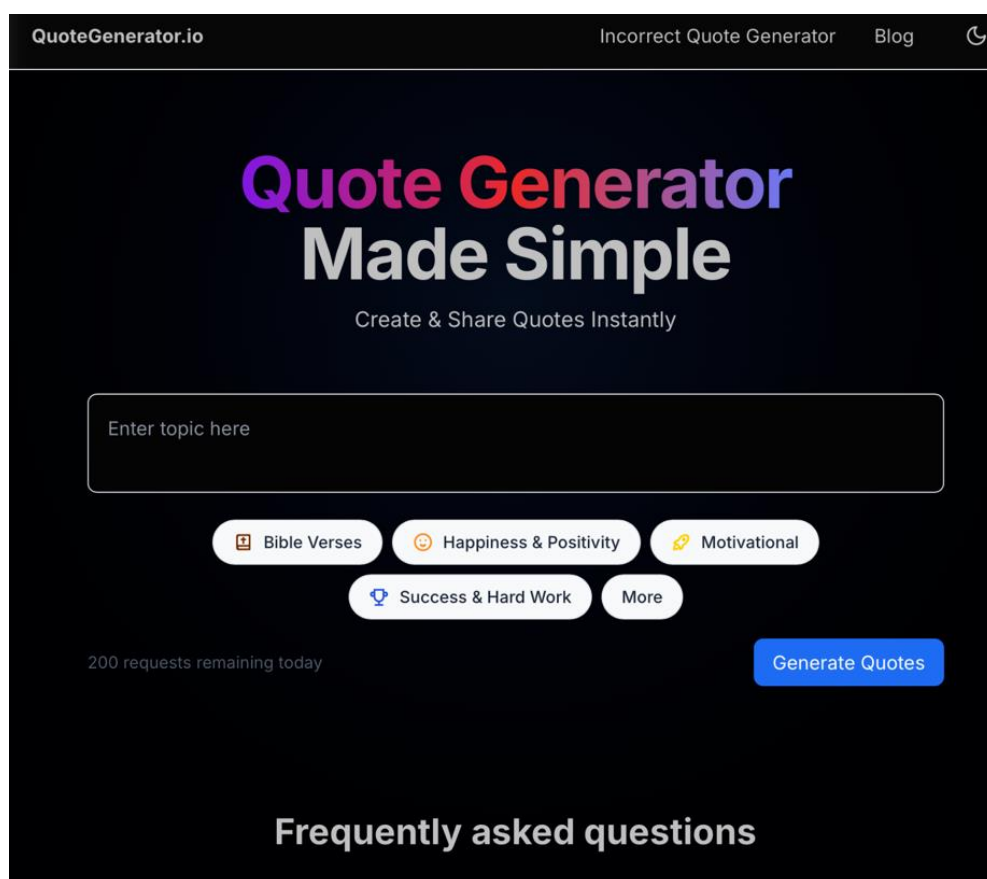
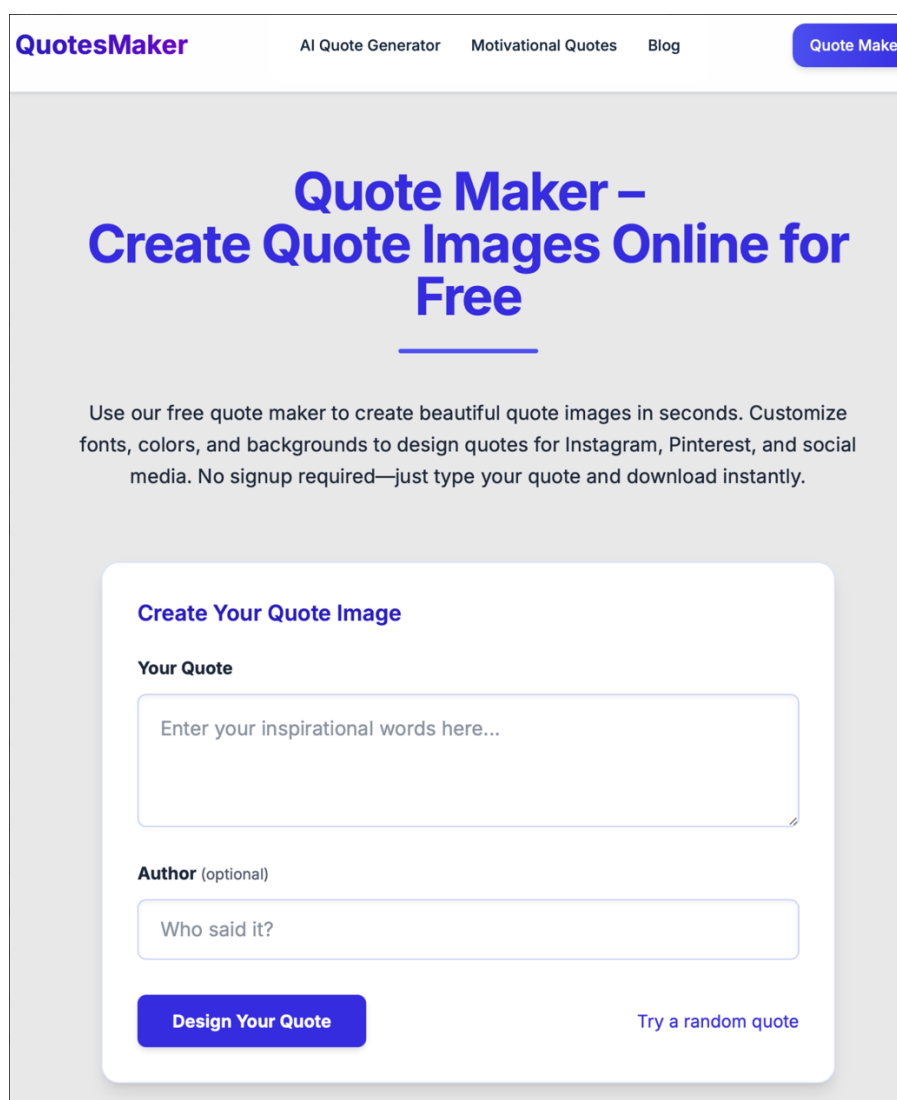


Рис. 1.5 Інтерфейс QuoteGenerator.io

QuotesMaker (див. рис. 1.6) орієнтований не лише на текстову генерацію, а й на створення зображень із цитатами. Користувач може ввести текст цитати, за потреби зазначити автора та перейти до оформлення зображення. Такий підхід є корисним для користувачів, які планують використовувати цитати в соціальних мережах, презентаціях, навчальних матеріалах або рекламних публікаціях. Перевагою сервісу є практична спрямованість на готовий візуальний результат. Водночас генерація змісту тут менш гнучка порівняно зі спеціалізованими AI-генераторами, оскільки основний акцент зроблено на дизайні цитати, а не на інтелектуальному доборі змісту.



The screenshot displays the QuotesMaker website interface. At the top, there is a navigation bar with the 'QuotesMaker' logo on the left, and links for 'AI Quote Generator', 'Motivational Quotes', and 'Blog' in the center. A blue button labeled 'Quote Maker' is positioned on the right. The main content area features a large, bold heading: 'Quote Maker – Create Quote Images Online for Free'. Below this heading, a paragraph describes the service: 'Use our free quote maker to create beautiful quote images in seconds. Customize fonts, colors, and backgrounds to design quotes for Instagram, Pinterest, and social media. No signup required—just type your quote and download instantly.' The central part of the interface is a white form titled 'Create Your Quote Image'. It contains two input fields: 'Your Quote' with a placeholder 'Enter your inspirational words here...' and 'Author (optional)' with a placeholder 'Who said it?'. At the bottom of the form, there is a blue button labeled 'Design Your Quote' and a link that says 'Try a random quote'.

Рис.1.6 Інтерфейс QuotesMaker

Порівняльну характеристику розглянутих рішень наведено в таблиці 1.2.

Таблиця 1.2

Порівняльний аналіз існуючих рішень для генерації мотиваційних цитат

Назва рішення	Основне призначення	Ключові можливості	Переваги	Недоліки
InspiroBot	Автоматична генерація мотиваційних цитат	Генерація цитати одним натисканням, поєднання тексту з фоновим зображенням	Дуже простий інтерфейс, швидке отримання результату, мінімальна кількість дій	Відсутність керування темою, стилем і змістом цитати
Base44	Створення вебзастосунків за текстовим описом	Формування структури застосунку на основі запиту, створення інтерфейсів і сторінок	Можливість швидкого прототипування власної системи	Не є спеціалізованим генератором цитат, потребує додаткового налаштування логіки
QuillBot AI Inspirational Quote Generator	Генерація натхненних цитат за темою	Введення теми, генерація тексту, готові приклади запитів	Якісний текстовий результат, підтримка тематичного контексту	Обмежені можливості візуального оформлення цитат
QuoteGenerator.io	Генерація цитат за темою або категорією	Введення теми, вибір категорій, генерація кількох цитат	Зручні тематичні категорії, простий інтерфейс	Можливі обмеження кількості запитів, недостатня персоналізація результату
QuotesMaker	Створення зображень із цитатами	Введення цитати, зазначення автора, оформлення графічного результату	Орієнтація на готові зображення для соціальних мереж, просте налаштування	Менший акцент на інтелектуальній генерації змісту

Як видно з таблиці 1.2, більшість розглянутих рішень реалізує лише окрему частину задачі. InspiroBot забезпечує швидку автоматичну генерацію, але майже не дає користувачу контролю над результатом. QuillBot і QuoteGenerator.io краще працюють із тематичним запитом, однак мають обмежену візуальну складову. QuotesMaker, навпаки, зосереджений на оформленні цитат, але не забезпечує достатньо гнучкої генерації тексту. Base44 є корисним прикладом AI-платформи для створення застосунків, проте не є готовим рішенням саме для генерації мотиваційних цитат.

Отже, аналіз існуючих рішень показав, що актуальним є розроблення системи, яка поєднує переваги розглянутих сервісів: швидку генерацію цитат, введення власної теми, вибір категорії, можливість персоналізації стилю, збереження результатів і створення готових візуальних матеріалів. На відміну від наявних сервісів, розроблювана система повинна бути не лише генератором випадкових висловів, а повноцінним інструментом для створення, редагування, зберігання та використання мотиваційних цитат у навчальних, особистих і медійних цілях.

1.3 Вибір технологій та інструментальних засобів реалізації системи

Розроблення інтелектуальної системи персоналізованої генерації мотиваційних цитат базується на багаторівневій архітектурі, де кожен рівень виконує окрему функцію: веб-взаємодію, оброблення запиту, збереження профілю, генерацію тексту, ранжування та аналітичну звітність. Такий підхід забезпечує керованість системи й можливість незалежного оновлення ML-компонентів.

Основною мовою програмування для серверної та аналітичної частини обрано Python, оскільки вона має розвинену екосистему бібліотек для NLP, машинного навчання, оброблення даних і створення веб-API. Для реалізації серверного шару доцільним є використання FastAPI, який забезпечить швидку обробку HTTP-запитів і зручне документування API.

Користувацький веб-інтерфейс доцільно реалізувати засобами React. Він забезпечує динамічне оновлення сторінок, роботу з формами профілю, перегляд історії генерацій, виставлення оцінок і візуалізацію статистики без перезавантаження сторінки.

Як сховище даних обрано PostgreSQL а також SQLite для локального прототипу. База даних зберігає користувачів, профілі, тематики, параметри генерації, створені цитати, оцінки та журнали роботи системи. ORM-рівень спрощує роботу з таблицями та підтримує узгодженість між доменною моделлю і фізичним сховищем.

ML-компонент системи використовує бібліотеки scikit-learn, Transformers, Pandas і NumPy. Для початкової персоналізації застосовуються методи векторизації, кластеризації та ранжування, а для створення тексту – генеративна мовна модель або спеціально підготовлений модуль на основі шаблонно-нейромережевого підходу.

Функціональні модулі системи охоплюють:

- керування обліковими записами, ролями та сесіями користувачів;
- формування профілю на основі анкети, історії оцінок і тематичних уподобань;
- генерацію кандидатних мотиваційних цитат з урахуванням тону, теми та контексту;
- ранжування результатів за показником релевантності та фільтрацію небажаних формулювань;
- візуалізацію історії, статистики оцінок і сформованих добірок у веб-інтерфейсі.

Таке технічне рішення дозволяє реалізувати замкнений цикл даних: від введення користувацького запиту до створення цитати, її оцінювання та подальшого уточнення профілю. Архітектура придатна як для локального навчального прототипу, так і для розгортання у хмарному середовищі.

Узагальнена структура обраних технологій наведена у таблиці 1.3, де вказано призначення кожного інструменту в межах основних підсистем програмного продукту.

Обраний стек технологій забезпечує поєднання веб-доступності, аналітичної гнучкості та достатньої продуктивності для оброблення запитів користувачів у режимі реального часу.

Таблиця 1.3

Вибір технологій та інструментальних засобів реалізації системи

Компонент системи	Технології та інструменти	Призначення
Веб-інтерфейс	React, HTML, CSS	Форма генерації, профіль, історія, статистика
Сервер застосунку	Python, FastAPI	API, бізнес-логіка, валідація запитів
Аналітичне ядро	Transformers, scikit-learn, Pandas	NLP-аналіз, генерація, кластеризація, QRI
Локальне / серверне сховище	SQLite / PostgreSQL	Збереження користувацьких і текстових даних
Візуалізація та звітність	Chart.js, HTML, PDF export	Графіки, таблиці, звіти користувача
Безпека	JWT, HTTPS, bcrypt	Авторизація, хешування паролів, захист API

1.4 Аналіз вимог розроблюваної системи

Аналіз вимог є ключовим етапом розроблення, оскільки визначає межі функціональності, якісні характеристики та технічні параметри майбутньої системи. Для інтелектуальної системи генерації мотиваційних цитат вимоги охоплюють роботу з профілями, створення тексту, оцінювання релевантності, безпеку даних і зручність веб-взаємодії.

За результатами аналізу предметної області систему поділено на три основні підсистеми: керування користувацьким профілем, генерація та

ранжування цитат, аналітика зворотного зв'язку. Кожна підсистема виконує окрему роль, але всі вони поєднані єдиним контуром персоналізації.

Функціональні вимоги системи наведено у таблиці 1.4. Вони визначають дії, які має виконувати програмний продукт: створювати профіль, приймати запит, генерувати кілька варіантів цитат, обирати найкращий, зберігати історію, приймати оцінку та формувати статистику.

Таблиця 1.4

Функціональні вимоги інформаційної системи

№	Назва вимоги	Опис функціональності	Очікуваний результат
1	Генерація цитат	Формування мотиваційних цитат на основі заданої тематики або параметрів запиту користувача.	Відображення релевантної цитати користувачу.
2	Вибір категорії	Можливість вибору тематичної категорії цитат (мотивація, успіх, саморозвиток тощо).	Фільтрація контенту згідно з інтересами.
3	Відображення результатів	Виведення згенерованої цитати у текстовому форматі в інтерфейсі користувача.	Зручний для читання графічний інтерфейс
4	Збереження цитат	Додавання обраних цитат до персонального списку користувача.	Створення персонального списку цитат.
5	Пошук цитати	Пошук цитат за ключовими словами або тематичними категоріями.	Швидкий доступ потрібних матеріалів.
6	Оцінювання цитат	Оцінювання або позначення цитат користувачем для формування рекомендацій	Адаптація алгоритму видачі під індивідуальні вподобання користувача.

Поряд із функціональністю важливими є нефункціональні вимоги, які описують швидкодію, надійність, масштабованість, захист персональних даних та якість інтерфейсу. Їх узагальнено у таблиці 1.5.

Таблиця 1.5

Нефункціональні вимоги системи

№	Категорія	Вимога	Характеристика
1	Продуктивність	Час генерації цитати	Час генерації цитати не перевищує 2 секунди.
2	Масштабованість	Робота з локальними даними	Підтримка оброблення до 500 одночасних користувацьких запитів.
3	Надійність	Резервне копіювання	Автоматичне створення резервних копій бази даних кожні 24 години з можливістю відновлення за 1 годину.
4	Безпека	Цілісність даних	Захист користувацьких даних із застосуванням механізмів автентифікації Django та контролю доступу (блокування на 15 хв після 5 невдалих спроб авторизації).
5	Кросплатформенність	Адаптивність інтерфейсу	Коректна робота у сучасних веб-браузерах (Chrome, Safari, Firefox) та адаптивний інтерфейс від 320px у ширину та від 568px у висоту.

На рівні технічної реалізації сформовано набір вимог до веб-застосунку, API, бази даних, ML-модуля та середовища розгортання. Ці вимоги наведено у таблиці 1.6 і використовуються як основа для вибору технологічного стеку.

Таблиця 1.6

Технічні вимоги програмної системи

№	Компонент	Технологія / Платформа	Призначення
1	Серверна частина	Python, FastAPI	REST API, бізнес-логіка, керування генерацією
2	ML-модуль	Transformers, scikit-learn, NumPy	NLP-аналіз, кластеризація, ранжування цитат
3	Клієнтська частина	React, HTML, CSS, JavaScript	Веб-інтерфейс користувача та адміністратора
4	База даних	PostgreSQL / SQLite	Збереження профілів, цитат, оцінок і звітів
5	Авторизація	JWT, HTTPS	Безпечний доступ до персональних даних
6	DevOps	Docker Compose, Git	Повторюване розгортання і контроль версій

Зведений аналіз вимог показує, що система повинна забезпечувати швидку відповідь на запит користувача, підтримувати якісне збереження профілю, бути захищеною від несанкціонованого доступу та мати можливість оновлення ML-компонентів без зміни інтерфейсної логіки.

Узагальнено система реалізує цикл «профіль → контекст → генерація → ранжування → оцінка → уточнення профілю». Саме цей цикл забезпечує відмінність розроблюваного рішення від статичних каталогів цитат і звичайних текстових генераторів.

1.5 Постановка завдання

Постановка завдання визначає мету, вхідні та вихідні дані, функціональні межі й логіку перетворення інформації в системі, що розробляється. У контексті даної роботи головним завданням є створення веб-орієнтованої інтелектуальної системи, здатної генерувати короткі мотиваційні цитати відповідно до індивідуального профілю користувача.

У процесі розроблення необхідно створити узгоджену архітектуру, яка поєднує рівень веб-інтерфейсу, серверне API, сховище даних, модуль аналізу профілю, генеративне ядро і підсистему оцінювання якості. Система повинна працювати як єдиний сервіс, у якому всі етапи формування цитати є контрольованими та відтворюваними.

Вхідні дані системи:

- користувацький профіль: ім'я, роль, цілі, тематичні інтереси, бажаний стиль мотивації;
- поточний контекст запиту: настрій, ситуація, ключові слова, очікувана довжина повідомлення;
- історія взаємодій: попередні цитати, оцінки, обрані категорії, частота використання;
- текстовий корпус: тематичні вислови, навчальні приклади, словники тональностей і заборонених формулювань;
- параметри ML-моделей: ваги ранжування, пороги релевантності, конфігурації генерації та температури вибірки.

Вихідні дані системи:

- персоналізована мотиваційна цитата українською мовою або іншою підтримуваною мовою;
- пояснювальні метадані: тема, тон, рівень релевантності, дата створення та використана модель;
- декілька альтернативних варіантів цитат для порівняння та ручного вибору;
- звітність про якість генерації, середні оцінки, популярні теми та динаміку користування;
- сповіщення або добірки цитат, сформовані за розкладом чи за подією користувача.

Формально завдання можна подати як перетворення множини вхідних характеристик користувача (1.1) на множину кандидатних мотиваційних повідомлень (1.2)

$$X = \{p, c, h, t, s\}, \quad (1.1)$$

де p - персональні вподобання користувача;
 c - ціль або контекст запиту;
 h - історія оцінок чи взаємодії користувача;
 t - тема або тип мотиваційного повідомлення;
 s - поточний емоційний стан користувача.

$$Y = g(X, \theta), \quad (1.2)$$

де g - генеративно-рекомендаційна функція, що поєднує мовну модель і механізм ранжування;

X - набір вхідних характеристик користувача;

θ - набір параметрів, оптимізованих за історією оцінок.

Таким чином, система не просто повертає довільний текст, а добирає формулювання з урахуванням персонального контексту і вимірюваної релевантності.

У результаті реалізації завдання буде створено програмну систему, яка:

- інтегрує веб-інтерфейс, базу даних, NLP-модуль і генеративну модель у єдину інфраструктуру;
- дозволяє користувачам отримувати мотиваційні повідомлення відповідно до особистих цілей і поточного настрою;
- забезпечує адаптивне навчання на основі оцінок, історії переглядів і тематичних вподобань;
- формує основу для подальшого розвитку персоналізованих сервісів цифрової підтримки та саморозвитку.

Отже, постановка завдання визначає логічну послідовність переходу від користувацького запиту до готового персоналізованого вислову, а також задає вимоги до моделі даних, алгоритмів і програмної архітектури.

1.6 Висновки до першого розділу

У першому розділі проведено системний аналіз предметної області, пов'язаної з персоналізованою генерацією мотиваційних цитат. Визначено основні проблеми статичних добірок висловів: відсутність індивідуального контексту, повторюваність формулювань, слабкий зворотний зв'язок і недостатнє врахування емоційного стану користувача.

Проаналізовано наявні рішення у сфері генеративних чат-ботів, платформ автоматичного письма, сервісів цитат і застосунків саморозвитку. Показано, що більшість із них або надає загальний текст без довготривалої персоналізації, або працює як каталог готових висловів без адаптивного навчання.

У процесі теоретико-методологічного обґрунтування сформовано концепцію системи з поділом на підсистеми профілю, генерації, ранжування, зворотного зв'язку та веб-візуалізації. UML-модельовання дозволило уточнити сценарії взаємодії користувача з системою та визначити логіку основних процесів.

Постановка завдання окреслила подальший напрям розроблення – створення веб-орієнтованої інтелектуальної системи, яка генерує унікальні мотиваційні цитати з урахуванням профілю, контексту і попередніх оцінок користувача. Отримані результати є основою для проєктування інформаційного та програмного забезпечення.

2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Логічна модель даних у вигляді ER-діаграми

Логічна модель даних є основою організації інформаційної бази системи, що забезпечує збереження користувацьких профілів, тематик, запитів, згенерованих цитат, оцінок і результатів аналітики. Її побудова спрямована на забезпечення цілісності даних, мінімізацію дублювання та підтримку персоналізації в межах одного інформаційного середовища.

На рисунку 2.1 подано ER-діаграму логічної моделі даних. Вона відображає ключові сутності: User, UserProfile, Topic, GenerationRequest, Quote, Feedback, ModelVersion і Report. Зв'язки між ними описують життєвий цикл персоналізованого вислову – від створення запиту до збереження оцінки та формування аналітичного звіту.

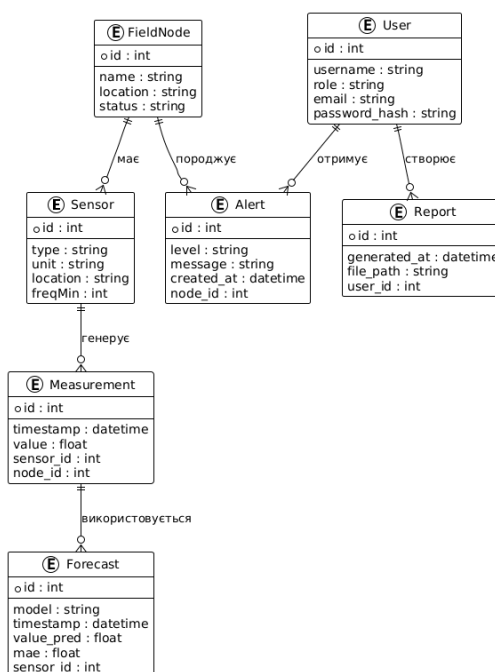


Рис. 2.1 ER-діаграма логічної моделі даних системи генерації мотиваційних цитат

Модель реалізує концепцію єдиного контуру персоналізації, у якому профіль користувача впливає на генерацію, результат оцінюється користувачем, а отриманий зворотний зв'язок уточнює майбутні рекомендації. Така структура забезпечує послідовне накопичення знань про вподобання без порушення нормалізації.

Основні об'єкти та їх атрибути наведено у таблиці 2.1. Вони визначають ядро даних системи: облікові записи, параметри профілю, тематичні категорії, журнали генерації, варіанти цитат, оцінки та сформовані звіти.

Таблиця 2.1

Основні сутності та атрибути логічної моделі даних

Сутність	Ключові атрибути	Тип даних	Призначення
User	id, email, password_hash, role	int, string	Обліковий запис користувача
UserProfile	id, user_id, goals, tone, language	int, string	Персональні параметри генерації
Topic	id, name, description, status	int, string	Тематика мотиваційних цитат
Quote	id, request_id, text, qri, model_id	int, string, float	Згенерована мотиваційна цитата
Feedback	id, quote_id, rating, comment, created_at	int, float, string	Оцінка користувача та зворотний зв'язок
ModelVersion	id, name, params, active	int, string, bool	Версія ранжувальної моделі
Report	id, user_id, generated_at, file_path	int, datetime, string	Аналітичний та експортний звіт

2.2 Моделювання класів та взаємодії компонентів системи

Діаграма класів формує об'єктно-орієнтовану модель системи та відображає структурні зв'язки між компонентами, що реалізують оброблення профілю, генерацію цитат, оцінювання релевантності та збереження результатів. Класи спроектовано з урахуванням інкапсуляції логіки, слабкої зв'язності та можливості розширення ML-модуля.

Класи системи поділено на функціональні групи: рівень користувача (User, UserProfile), рівень оброблення запиту (GenerationRequest, PromptBuilder), рівень генерації (NLPProcessor, QuoteGenerator), рівень ранжування (RecommendationEngine, MetricsService), рівень збереження (QuoteStore, FeedbackRepository) та рівень інтерфейсу (WebDashboard).

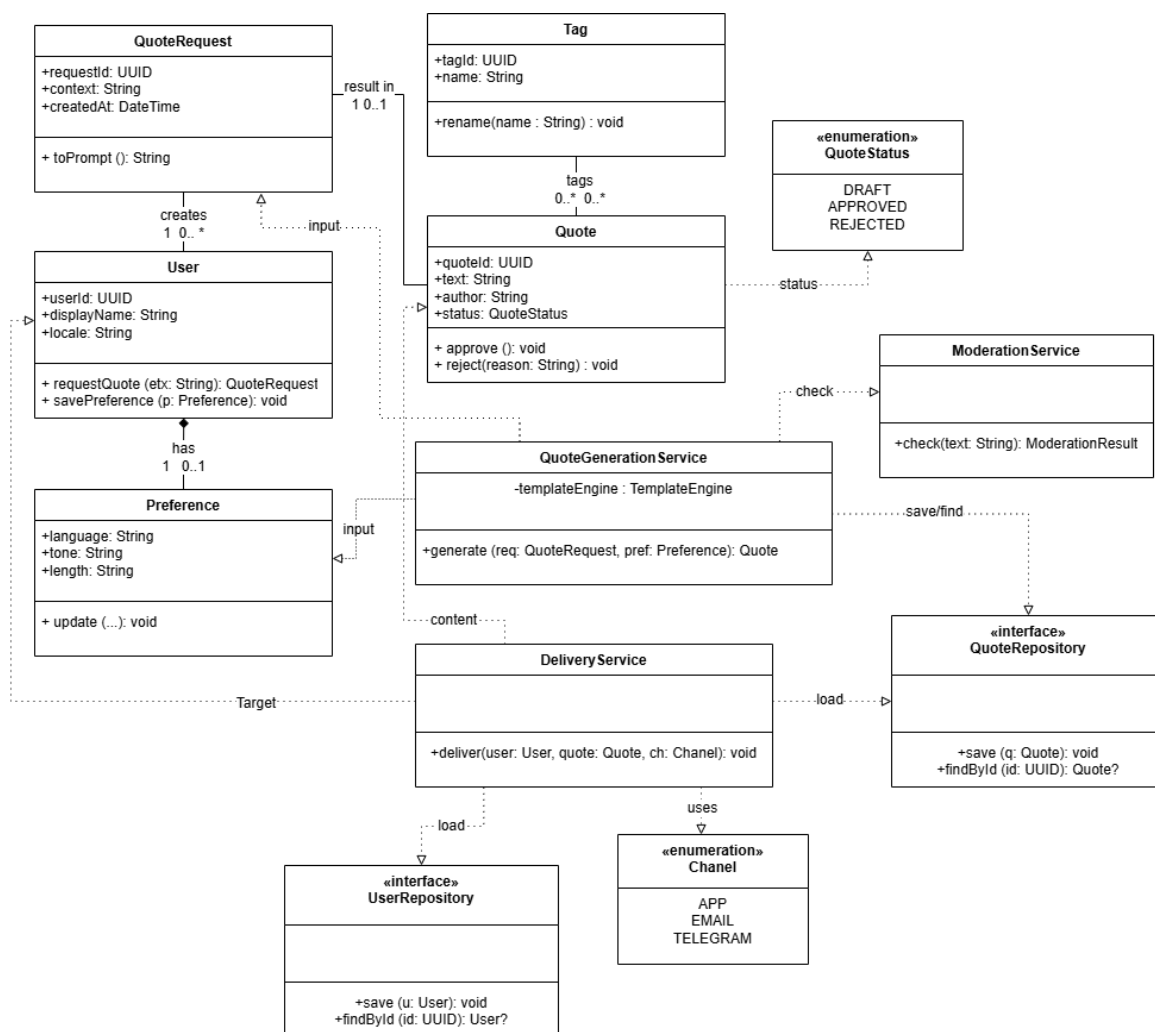


Рис. 2.2 UML-діаграма класів програмного забезпечення системи персоналізованої генерації цитат

Зв'язки реалізовані через асоціації, залежності й агрегації. UserProfile агрегує параметри користувача, PromptBuilder формує контекст для генеративної моделі, QuoteGenerator створює кілька кандидатних текстів, RecommendationEngine ранжує їх, а FeedbackService зберігає реакцію користувача для подальшого уточнення профілю.

Для відображення динаміки взаємодії створено три діаграми кооперації, подані на рисунку 2.3-2.5. Вони демонструють сценарії формування профілю, генерації цитати та розрахунку показників релевантності.

Перша кооперація описує процес створення та оновлення профілю користувача: дані анкети передаються до PreferenceAnalyzer, нормалізуються і зберігаються у ProfileStore.

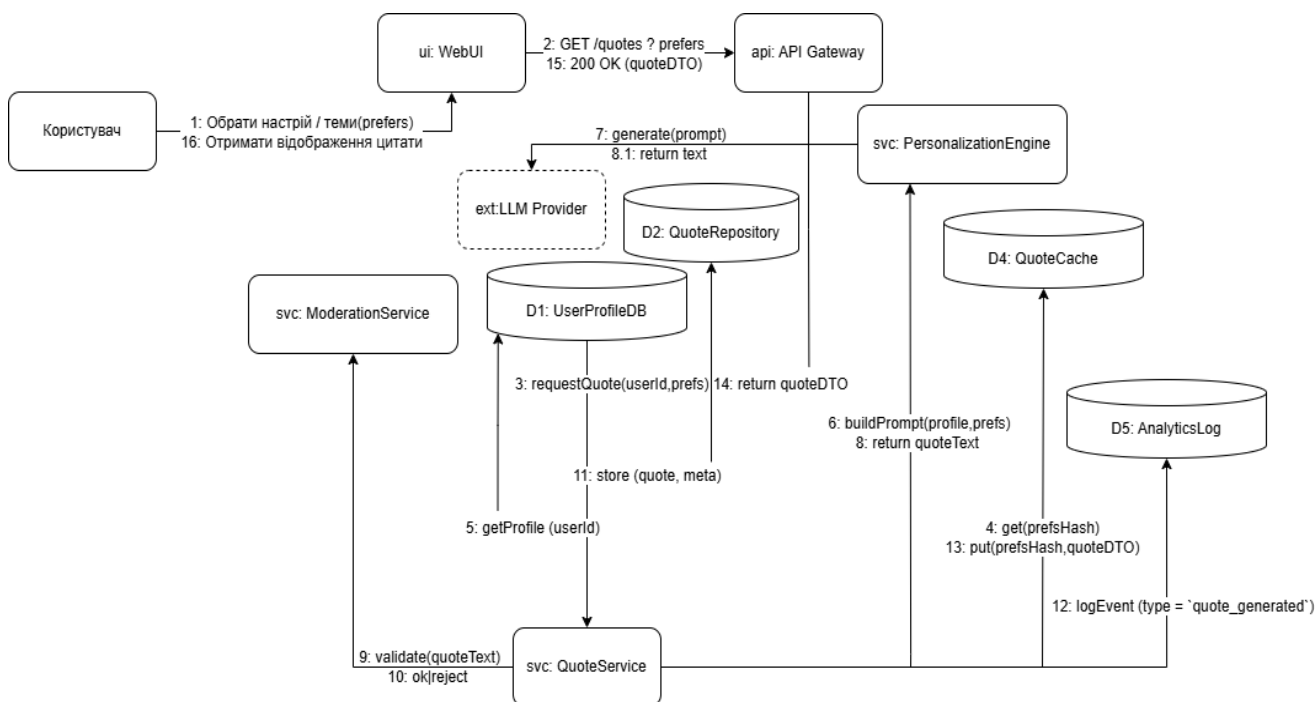


Рис. 2.3 Формування профілю між UserProfile, PreferenceAnalyzer, ProfileStore та AuthService

Друга кооперація ілюструє основний контур генерації: WebDashboard надсилає запит, NLPProcessor готує мовний контекст, QuoteGenerator створює кандидатні фрази, а FeedbackService фіксує реакцію користувача.

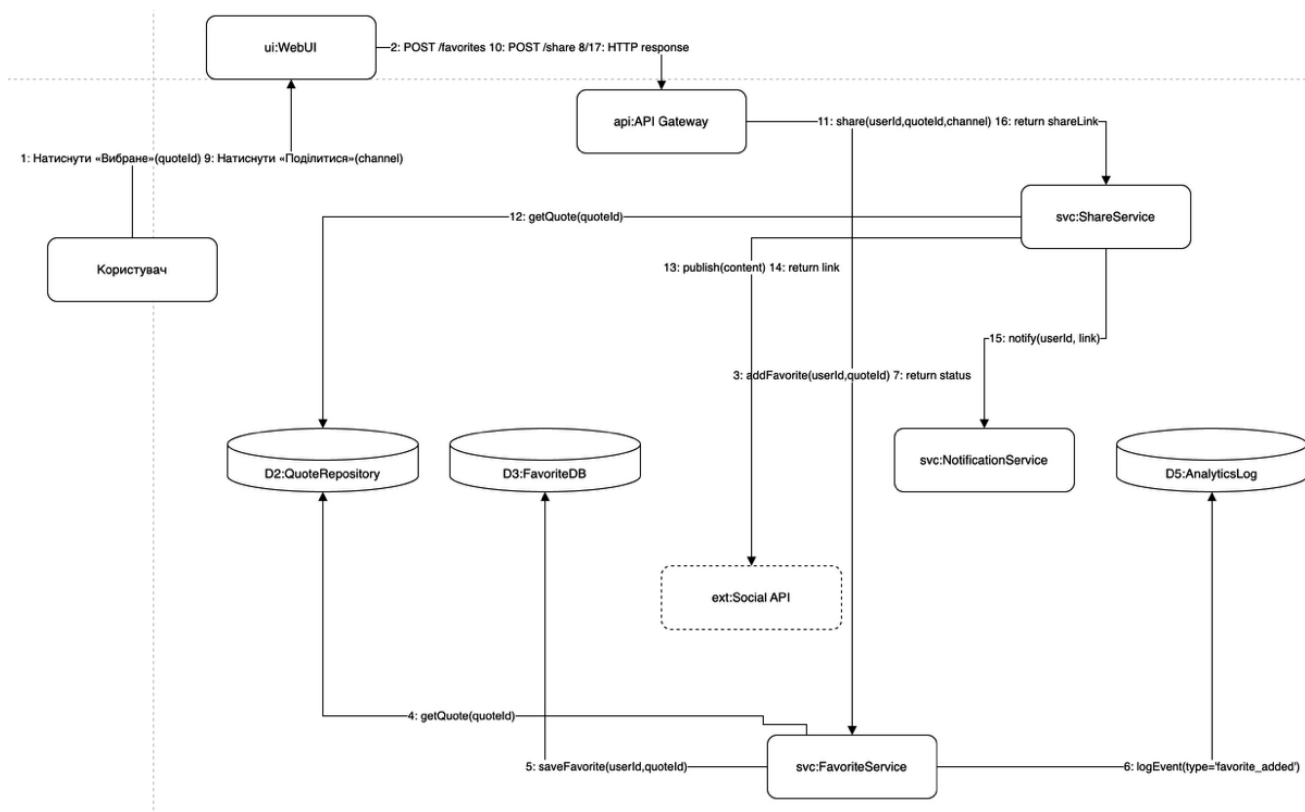


Рис. 2.4 Генерація цитати між WebDashboard, QuoteGenerator, NLPProcessor та FeedbackService

Третя кооперація описує механізм ранжування: RecommendationEngine отримує кандидати, порівнює їх із профілем, використовує дані QuoteStore і передає оцінки до MetricsService.

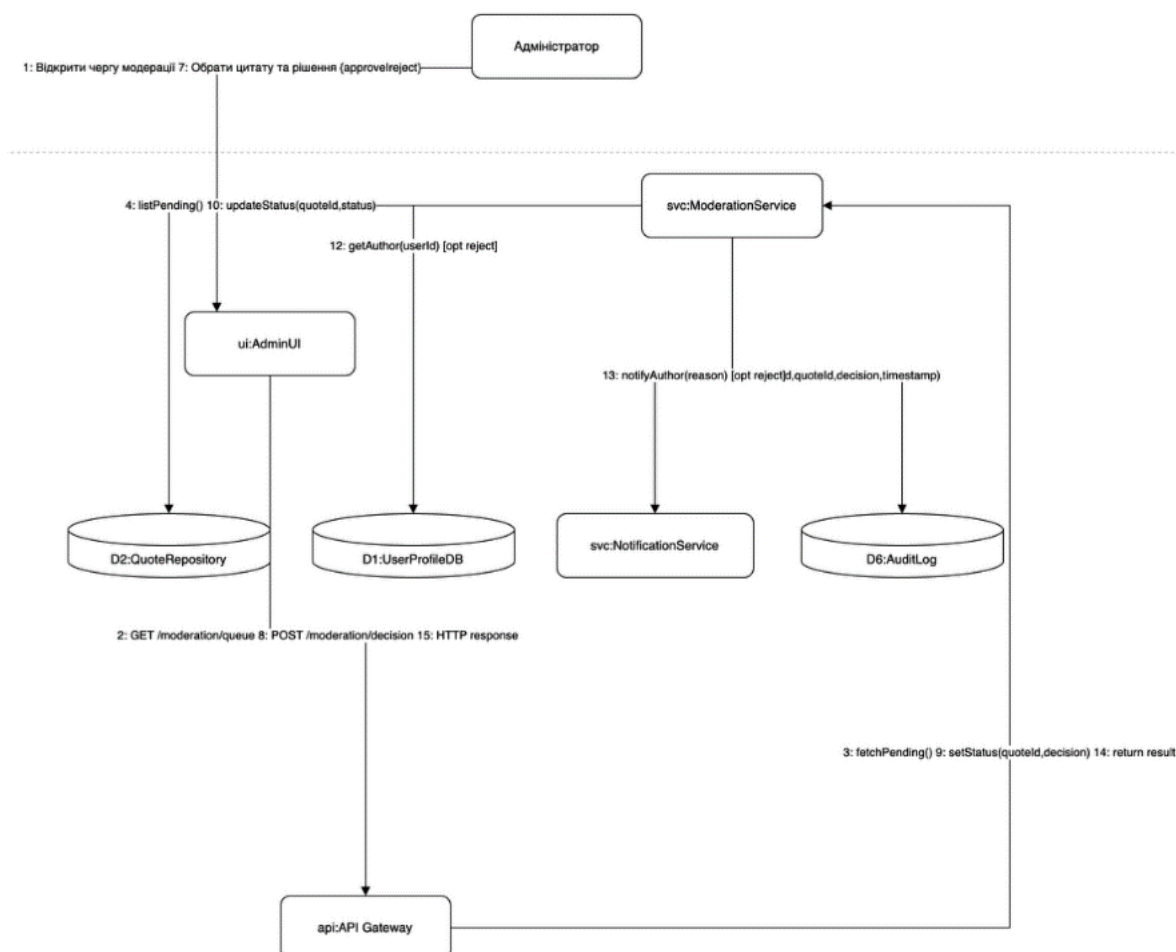


Рис. 2.5 Оцінювання релевантності між RecommendationEngine, QuoteStore та MetricsService

Системна архітектура, сформована на основі класів і кооперацій, відповідає принципам об'єктної декомпозиції та модульності. Взаємодія між компонентами здійснюється через чіткі методи та API-контракти, що спрощує тестування й подальше розширення.

Таблиця 2.2

Структурна узгодженість класів та їх функціональні ролі

Клас	Рівень	Основна функція	Тип зв'язку
UserProfile	Доменний	Збереження цілей, тематик і тону користувача	Агрегація
PromptBuilder	Прикладний	Формування запиту до генеративної моделі	Залежність
NLPProcessor	Аналітичний	Токенізація, векторизація та аналіз тону	Асоціація
QuoteGenerator	ML-рівень	Створення кандидатних мотиваційних висловів	Асоціація
RecommendationEngine	Аналітичний	Ранжування цитат за профілем і QRI	Залежність
FeedbackService	Сервісний	Збереження оцінок і оновлення профілю	Асоціація
WebDashboard	Інтерфейс	Відображення форми, цитат, історії та статистики	Асоціація

Результуюча модель демонструє цілісну логіку створення, зберігання та оцінювання персоналізованих мотиваційних цитат. Вона підтримує додавання нових тематик, моделей генерації та критеріїв ранжування без суттєвого перепроектування системи.

2.3 Проектування компонентної архітектури системи

Діаграма компонентів відображає архітектурну організацію програмного забезпечення, яке поєднує веб-інтерфейс, серверне API, базу даних, NLP-модуль, генеративне ядро та підсистему зворотного зв'язку. Основна мета цієї моделі полягає у розмежуванні відповідальності між компонентами та забезпеченні стабільної взаємодії між ними.

На рисунку 2.6 наведено UML-діаграму компонентів системи. Користувацький веб-клієнт взаємодіє з API Gateway, який передає запити до сервісів профілю, генерації, ранжування, звітності та авторизації. Дані

зберігаються у реляційній базі, а генеративний модуль працює як окремий сервіс, що може оновлюватися незалежно.

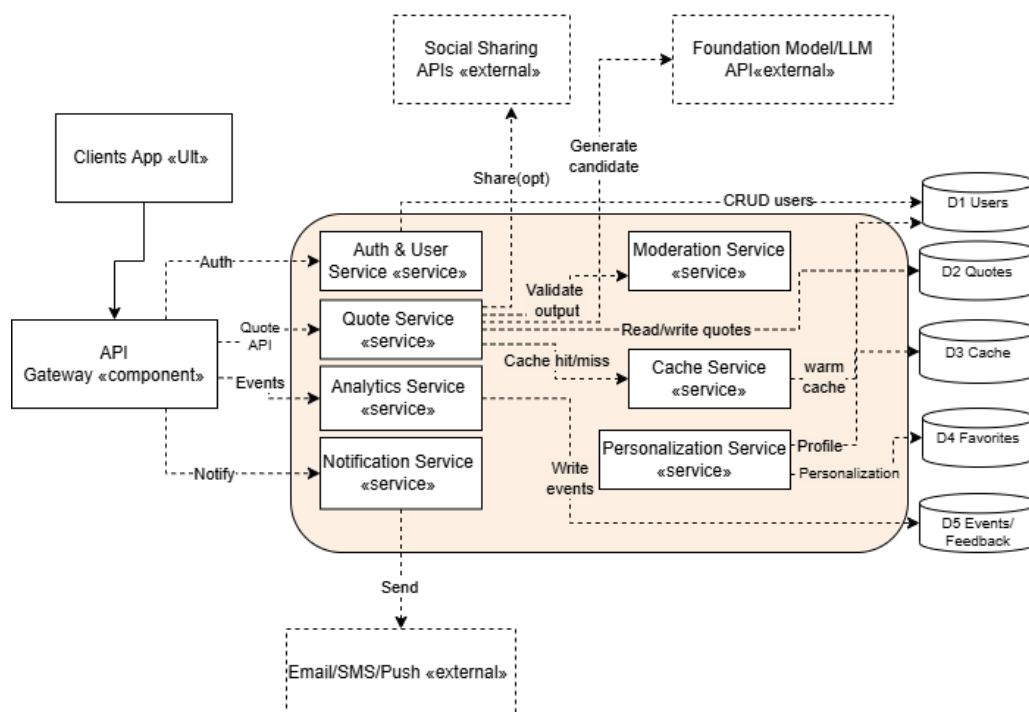


Рис. 2.6 Діаграма компонентів програмного забезпечення системи генерації мотиваційних цитат

Архітектура побудована за принципом розділення функціональних відповідальностей. Інтерфейсний рівень відповідає за введення параметрів і відображення цитат, серверний рівень – за бізнес-логіку, аналітичний рівень – за NLP-обробку й ранжування, а рівень даних – за збереження профілів, історії генерацій і метрик.

Така структура мінімізує надлишкові зв'язки між модулями, спрощує масштабування та дає можливість підключення нових моделей генерації без зміни веб-інтерфейсу. Компоненти взаємодіють через REST API та внутрішні сервісні контракти.

Зведена характеристика архітектури подана у таблиці 2.3, де наведено ключові шари системи, основні взаємодії між ними та призначення кожного компонента.

Таблиця 2.3

Структурно-функціональні зв'язки компонентів системи

Шар системи	Основна взаємодія	Призначення
Користувацький	Передача параметрів запиту до API	Забезпечення веб-контролю і взаємодії
Сервісний	Валідація, авторизація, маршрутизація запитів	Керування бізнес-логікою системи
Аналітичний	NLP-обробка, генерація, ранжування	Формування персоналізованого тексту
Дані	Збереження профілів, цитат, оцінок	Підтримка історії та персоналізації
Безпековий	JWT, HTTPS, ролі доступу	Захист персональних даних користувача

Результуюча компонентна структура забезпечує баланс між автономністю модулів і єдиною логікою персоналізації. Це дозволяє підтримувати стабільну роботу системи, розширювати функціональність і проводити тестування окремих сервісів без зупинки всієї платформи.

2.4 Пакетна організація програмного забезпечення системи

Діаграма пакетів відображає логічну організацію програмного коду системи. Вона демонструє розподіл модулів за рівнями відповідальності: інтерфейс, прикладна логіка, доменна модель, ML-обробка, доступ до даних, безпека та звітність.

На рисунку 2.7 подано UML-діаграму пакетів, яка формалізує зв'язки між функціональними частинами системи. Її структура базується на принципах слабкої зв'язності та високої внутрішньої узгодженості, що дає змогу змінювати окремі пакети без порушення роботи всього застосунку.

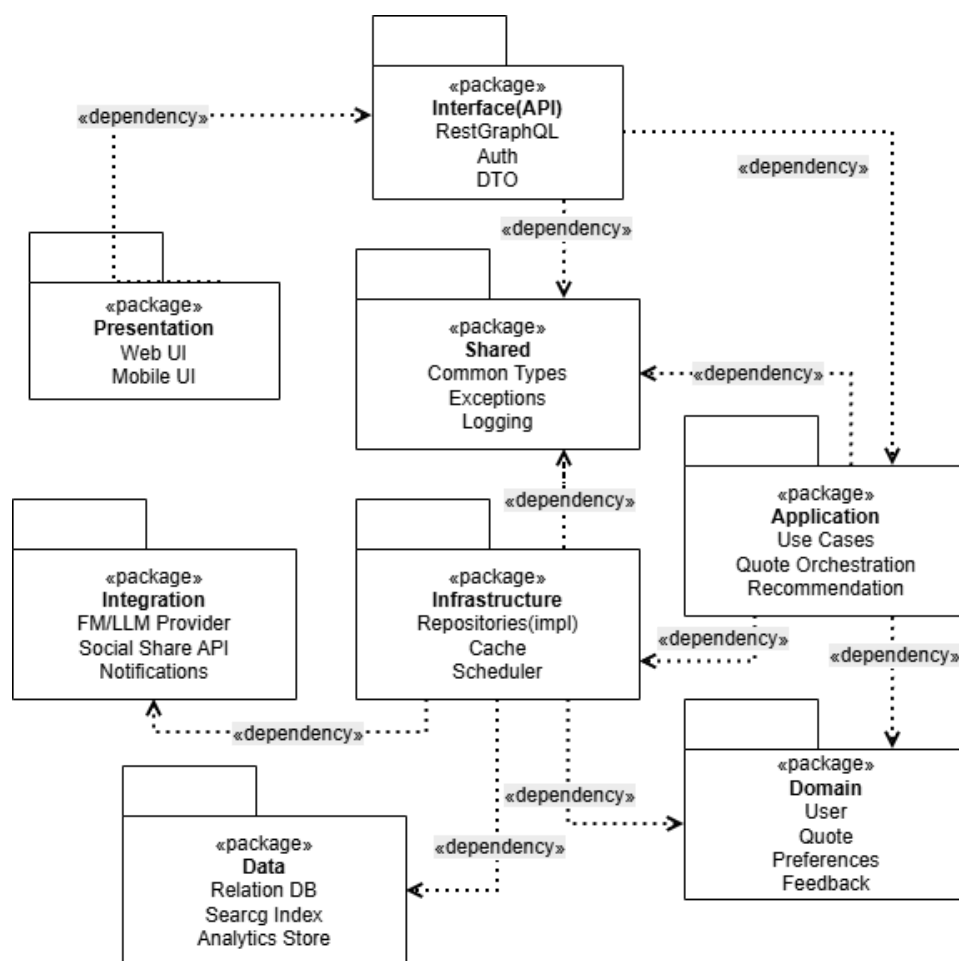


Рис. 2.7 Діаграма пакетів програмного забезпечення системи генерації мотиваційних цитат

Пакетна організація підтримує три базові напрями системної логіки:

- інформаційний потік – передача даних від веб-інтерфейсу до API, сховища та сервісів генерації;
- аналітичний контур – побудова профілю, генерація цитат, ранжування та розрахунок метрик якості;
- керуючий рівень – автентифікація, авторизація, валідація запитів і контроль доступу до історії користувача.

Ключовим аспектом цієї моделі є функціональна ізоляція між шарами. Пакет UI не працює напряму з базою даних, ML-модуль не відповідає за авторизацію, а доступ до сховища здійснюється через репозиторії та сервісний шар.

Семантичні зв'язки між пакетами узагальнено у таблиці 2.4. Вона показує, які пакети взаємодіють між собою та яку роль вони виконують у загальній архітектурі.

Таблиця 2.4

Взаємозв'язки пакетів та їх системні функції

Рівень	Пакети взаємодії	Основне призначення
Дані	domain → repositories → database	Збереження профілів, цитат і оцінок
Аналітика	ml_engine ↔ services.recommendation	Генерація, ранжування і розрахунок QRI
Керування	api ↔ security ↔ ui	Авторизація, валідація і веб-взаємодія
Інтеграція	api ↔ reports ↔ export	Формування звітів та обмін результатами

2.5 Висновки до другого розділу

У другому розділі сформовано логіко-структурну модель інтелектуальної системи персоналізованої генерації мотиваційних цитат. Побудовано ER-діаграму, UML-діаграму класів, діаграми кооперацій, компонентів і пакетів, що разом описують інформаційний, об'єктний та архітектурний рівні системи.

Завдяки структурній нормалізації система отримала модульну архітектуру з чітким розподілом між профілем користувача, генеративним модулем, сервісом ранжування, сховищем даних, інтерфейсом і безпековим контуром. Такий підхід забезпечує розширюваність і керованість програмного продукту.

Результати моделювання підтверджують доцільність застосування UML-засобів для опису систем цього класу. Вони дають змогу формалізувати потоки даних, взаємодію класів і логічні залежності між сервісами, що зменшує ризик помилок на етапі реалізації.

Отримана архітектурна модель визначає основу для програмної реалізації веб-сервісу, що поєднує NLP-аналіз, генерацію тексту, персоналізацію та накопичення користувацького зворотного зв'язку.

3 ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Моделювання предметної області

Моделювання предметної області передбачає формалізацію сценаріїв використання, логіки взаємодії компонентів і процесу створення персоналізованої мотиваційної цитати. Для цього побудовано UML-діаграми варіантів використання, послідовності та активності, які описують функціональну поведінку системи на різних рівнях деталізації.

На рисунку 3.1 подано діаграму варіантів використання, де основними акторами є зареєстрований користувач, адміністратор, ML-модуль, база даних і зовнішній сервіс автентифікації. Користувач формує запит, обирає тему, задає настрій, отримує цитату, оцінює її та переглядає історію генерацій.

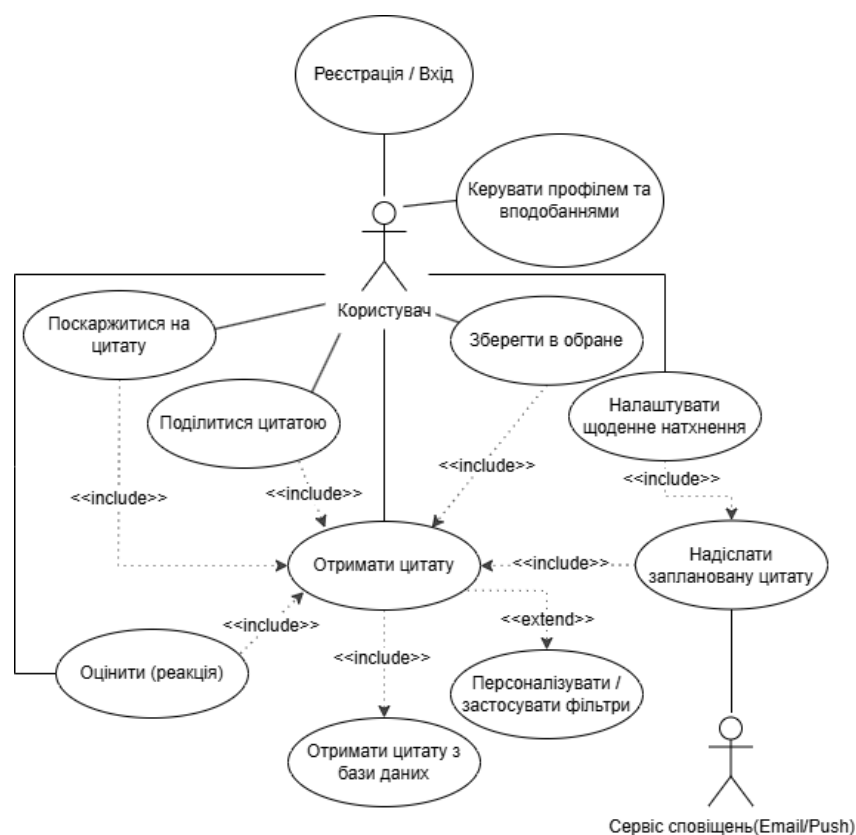


Рис. 3.1 Діаграма варіантів використання системи персоналізованої генерації мотиваційних цитат

Основними прецедентами є реєстрація, налаштування профілю, вибір цілі, генерація цитати, ранжування варіантів, збереження результату, формування рекомендацій, експорт добірки та перегляд статистики. Адміністратор керує тематичними категоріями, перевіряє якість корпусу і контролює параметри роботи ML-компонента.

Динамічна взаємодія між компонентами представлена на рисунку 3.2. Користувач надсилає запит через веб-інтерфейс, сервер перевіряє авторизацію, отримує профіль з бази даних, формує вектор контексту, звертається до генеративного модуля, ранжує кілька варіантів і повертає найрелевантніший результат.

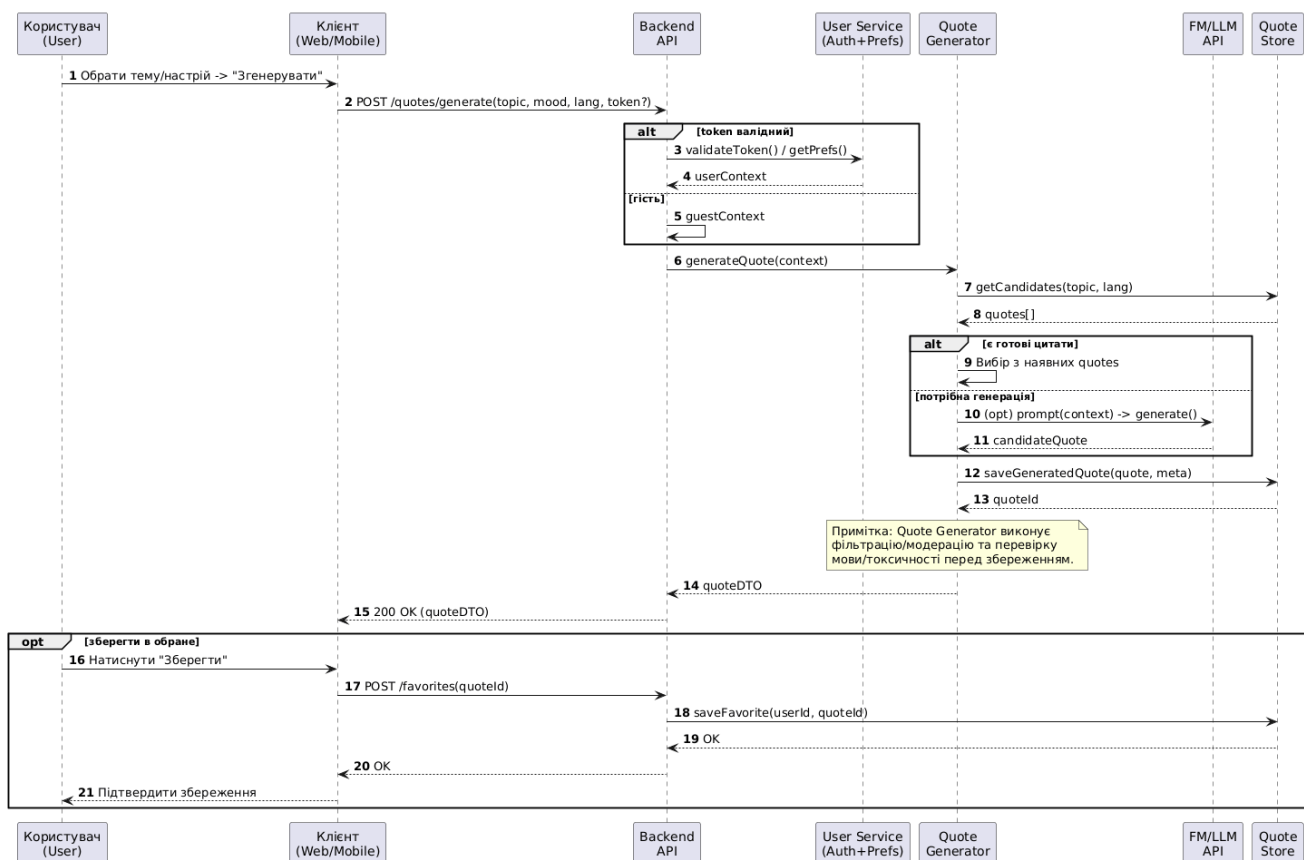


Рис. 3.2 Діаграма послідовності процесу створення персоналізованої цитати

Процесна логіка функціонування системи наведена на рисунку 3.3. Вона охоплює автентифікацію, вибір теми, заповнення параметрів запиту, генерацію, перевірку на унікальність, відображення цитати, отримання оцінки та оновлення персонального профілю.

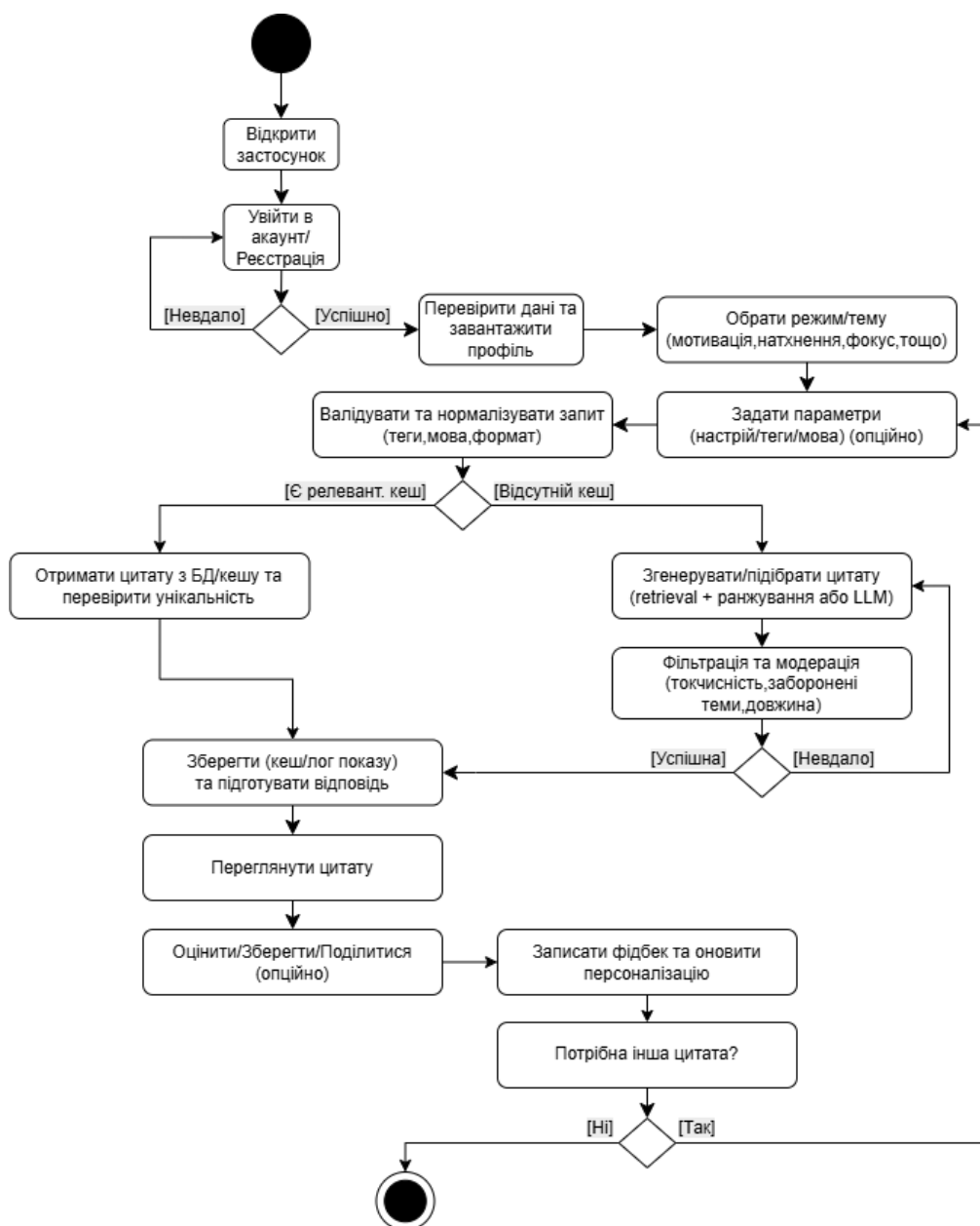


Рис. 3.3 Діаграма активності роботи користувача із системою

Завершальний етап полягає у збереженні взаємодії та перерахунку показників якості. Якщо користувач позитивно оцінює цитату, система посилює вагу відповідних тематик і тону; якщо оцінка низька, механізм персоналізації коригує параметри наступної генерації.

Сформована модель предметної області забезпечує узгодженість між функціональними сценаріями, структурою даних і архітектурою програмного забезпечення, що є основою для подальшого проєктування логічної моделі й алгоритмів.

3.2 Архітектура системи та проєктування функціоналу результатів дослідження

Архітектура програмної системи генерації мотиваційних цитат побудована за багаторівневим принципом, що забезпечує розділення користувацького інтерфейсу, прикладної логіки, сервісів оброблення запитів, сховищ даних та зовнішніх інтеграцій. Такий підхід дає змогу спростити подальшу підтримку системи, забезпечити масштабованість окремих модулів і підвищити надійність оброблення користувацьких запитів.

Загальну архітектуру системи подано на рисунку 3.4. Вона охоплює три основні частини: клієнтський застосунок, серверну частину та рівень даних і зовнішніх інтеграцій.

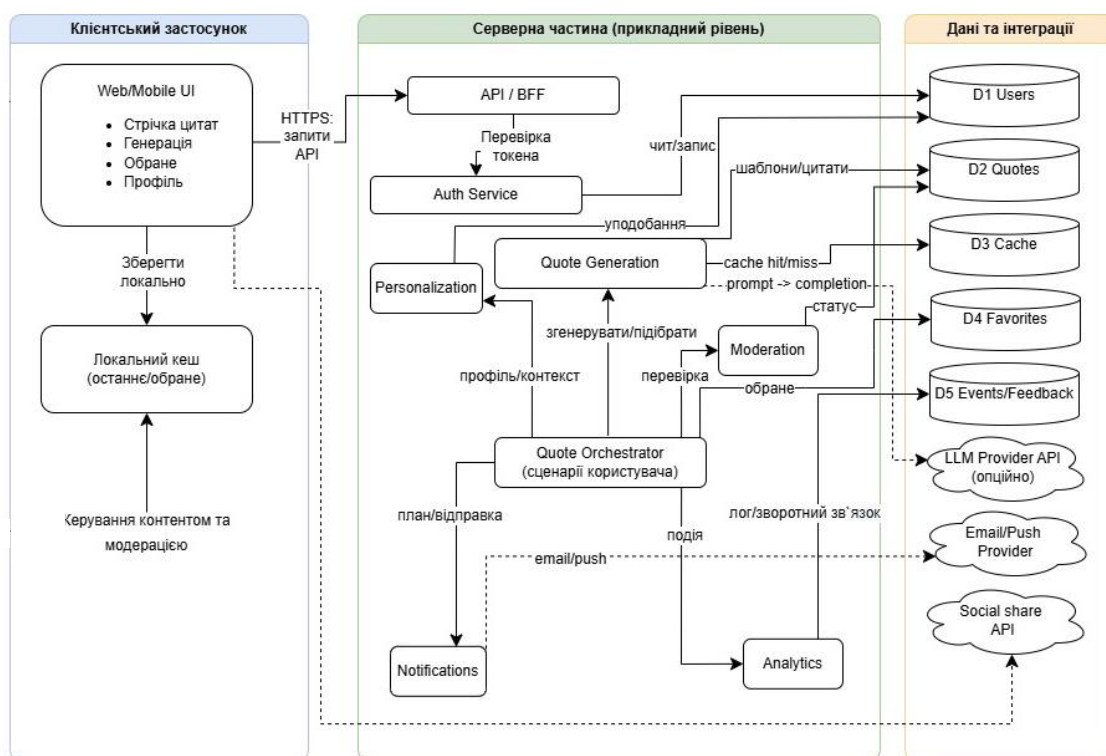


Рис. 3.4 Архітектура системи генерації мотиваційних цитат

На рівні користувачів передбачено дві основні ролі: користувач і адміністратор. Користувач взаємодіє із системою через веб- або мобільний інтерфейс, виконує запити на генерацію цитат, переглядає результати, додає цитати до обраного, зберігає їх у профілі та може поширювати сформовані матеріали. Адміністратор здійснює керування контентом, перегляд аналітичних показників, контроль якості згенерованих цитат і налаштування параметрів роботи системи.

Клієнтський застосунок реалізує основні екрани взаємодії з користувачем: сторінку генерації, розділ перегляду результатів, обране, історію запитів і профіль користувача. Через інтерфейс користувач вводить тему, вибирає категорію або стиль цитати, запускає генерацію та отримує результат. Для підвищення зручності роботи частина даних може зберігатися у локальному кеші, зокрема останні запити, тимчасові результати генерації та обрані цитати. Це дає змогу швидше відображати повторно використані дані та зменшує кількість звернень до серверної частини.

Серверна частина виконує основну прикладну логіку системи. Центральним компонентом є API/BFF, який приймає HTTPS-запити від клієнтського застосунку, перевіряє їх коректність і передає до відповідних внутрішніх сервісів. Використання BFF-рівня дозволяє адаптувати відповіді сервера під потреби клієнтського інтерфейсу та не перевантажувати фронтенд зайвою логікою.

За автентифікацію і контроль доступу відповідає Auth Service. Він перевіряє токени доступу, забезпечує авторизацію користувачів і обмежує доступ до персональних даних, історії запитів та адміністративних функцій. Після успішної перевірки токена запит передається до сервісів генерації, персоналізації або збереження даних.

Основний функціонал системи реалізується компонентом Quote Generation. Він відповідає за формування мотиваційних цитат на основі введеної теми, вибраної категорії, мови, настрою та додаткових параметрів. У разі використання зовнішньої мовної моделі сервіс формує структурований prompt,

передає його до LLM Provider API, отримує відповідь, виконує первинне опрацювання тексту та передає результат на перевірку.

Для покращення якості результатів використовується компонент Personalization. Він враховує профіль користувача, попередні запити, збережені цитати, обрані категорії та оцінки користувача. На основі цих даних система може пропонувати більш релевантні цитати, рекомендовані теми та персоналізовані варіанти формулювання. Наприклад, для користувача, який часто обирає теми саморозвитку, система може частіше пропонувати цитати про дисципліну, навчання, досягнення цілей і продуктивність.

Важливим елементом серверної частини є Moderation. Цей компонент перевіряє згенерований текст на наявність небажаного, некоректного або неприйняттого змісту. Якщо результат не відповідає встановленим правилам, система може повторно сформулювати запит, змінити prompt або повернути користувачу повідомлення про неможливість генерації за заданими параметрами. Наявність модуля модерації є необхідною для забезпечення безпечної та коректної роботи системи.

Координацію процесу генерації забезпечує Quote Orchestrator. Він відповідає за послідовність виконання операцій: отримання запиту, перевірку користувача, підготовку параметрів, звернення до генератора цитат, запуск модерації, збереження результату, оновлення кешу та передавання відповіді клієнтському застосунку. Завдяки цьому логіка роботи системи не розміщується в одному монолітному модулі, а розподіляється між спеціалізованими компонентами.

Для зберігання даних передбачено кілька логічних сховищ. D1 Users містить облікові записи користувачів, профілі, ролі та налаштування. D2 Quotes використовується для збереження згенерованих цитат, їх категорій, мови, стилю та службових параметрів. D3 Favorites містить зв'язки між користувачами та цитатами, доданими до обраного. D4 Events/Feedback зберігає події взаємодії, оцінки, реакції, історію запитів і дані для аналітики. D5 Cache використовується

для тимчасового збереження популярних запитів, проміжних результатів і відповідей, що часто повторюються.

Основні компоненти архітектури та їх функціональне призначення наведено в таблиці 3.1.

Після формування цитати система може виконати кілька додаткових дій. Якщо користувач задоволений результатом, цитата зберігається у базі даних D2 Quotes, а у разі додавання до обраного створюється відповідний запис у D3 Favorites. Якщо користувач оцінює цитату або повторно генерує результат, ця подія передається до D4 Events/Feedback. Надалі такі дані можуть використовуватися для персоналізації, аналізу якості генерації та визначення найбільш популярних тематичних напрямів.

Таблиця 3.1

Призначення основних компонентів системи

Компонент	Призначення	Основні функції
Web/Mobile UI	Взаємодія користувача із системою	Введення теми, вибір категорії, запуск генерації, перегляд і збереження цитат
Local Cache	Локальне зберігання тимчасових даних	Збереження останніх запитів, швидке відображення обраного та історії
API/BFF	Приймання та маршрутизація запитів	Оброблення HTTPS-запитів, передавання даних між клієнтом і сервісами
Auth Service	Автентифікація та авторизація	Перевірка токенів, керування доступом, захист персональних даних
Quote Generation	Генерація мотиваційних цитат	Формування prompt, звернення до мовної моделі, оброблення результату
Personalization	Персоналізація результатів	Урахування профілю, попередніх запитів, і вподобань користувача
Moderation	Перевірка якості та безпечності контенту	Фільтрація некоректних результатів, перевірка відповідності правилам системи
Quote Orchestrator	Координація процесу генерації	Керування сценарієм генерації, модерації, збереження результату
Notifications	Надсилання повідомлень	Email- і push-сповіщення про збережені цитати, рекомендації
Analytics	Збір і аналіз подій	Облік запитів, популярних категорій, оцінок і дій користувачів

Окрему роль у системі відіграють зовнішні інтеграції. LLM Provider API використовується для генерації текстового контенту за допомогою мовної моделі. Social Share API забезпечує можливість поширення цитати в соціальних мережах або передавання її до інших сервісів. Email/Push Provider використовується для надсилання сповіщень користувачам, наприклад щоденних мотиваційних цитат, рекомендацій або повідомлень про збережені матеріали.

Функціонал результатів дослідження доцільно подати як сукупність взаємопов'язаних можливостей, що реалізуються через описану архітектуру. До основних функцій системи належать: реєстрація та вхід користувача, генерація мотиваційної цитати за темою, вибір категорії або стилю, перегляд результату, повторна генерація, збереження цитати, додавання до обраного, перегляд історії, персоналізовані рекомендації, поширення цитати та адміністрування контенту.

Процес роботи користувача із системою починається з відкриття клієнтського інтерфейсу. Користувач вводить тему або вибирає одну із запропонованих категорій. Після натискання кнопки генерації клієнтський застосунок надсилає HTTPS-запит до API/BFF. Сервер перевіряє токен доступу, формує сценарій оброблення запиту та передає параметри до Quote Orchestrator. Далі виконується персоналізація, генерація тексту, модерація результату, збереження події та повернення готової цитати на клієнтський рівень.

Для адміністратора система передбачає окремі функції контролю та керування. Адміністратор може переглядати статистику використання системи, аналізувати популярні категорії, контролювати якість згенерованих цитат, керувати шаблонами prompt-запитів і правилами модерації. Це дозволяє підтримувати якість контенту та своєчасно коригувати логіку роботи системи.

3.3 Опис функціональних модулів системи

Функціональна архітектура системи персоналізованої генерації мотиваційних цитат базується на трьох основних модулях: генерації, рекомендації та оцінювання релевантності. Така структура дозволяє залежності відповідальності компонентів, підвищує модульність системи і полегшує масштабіність системи.

Модуль генерації цитат (QuoteGenerator) генерує множину кандидатних мотиваційних висловів на основі запиту користувача і параметрів його профілю. На вході модуль отримує контекст, сформований на основі NLP, після чого генерує альтернативний набір варіацій із зовнішнім LLM-провайдером. Отримані результати передаються до наступного етапу оброблення.

Модуль рекомендації та ранжування (RecommendationEngine) оцінює згенеровані цитати та сортує їх відповідно до користувацького запиту і параметрів його профілю. Найважливішою задачею цього модуля є вибір найбільш релевантної цитати, а саме для цього використовується показник QRI, який дозволяє формалізувати, наскільки кожна цитата відповідає профілю користувача.

Модуль оцінки QRI, виконує розрахунок інтегральної релевантності, яка враховує відповідність теми, емоції та оцінок користувача. Отримане значення використовується модулем рекомендації для фінального ранжування результатів.

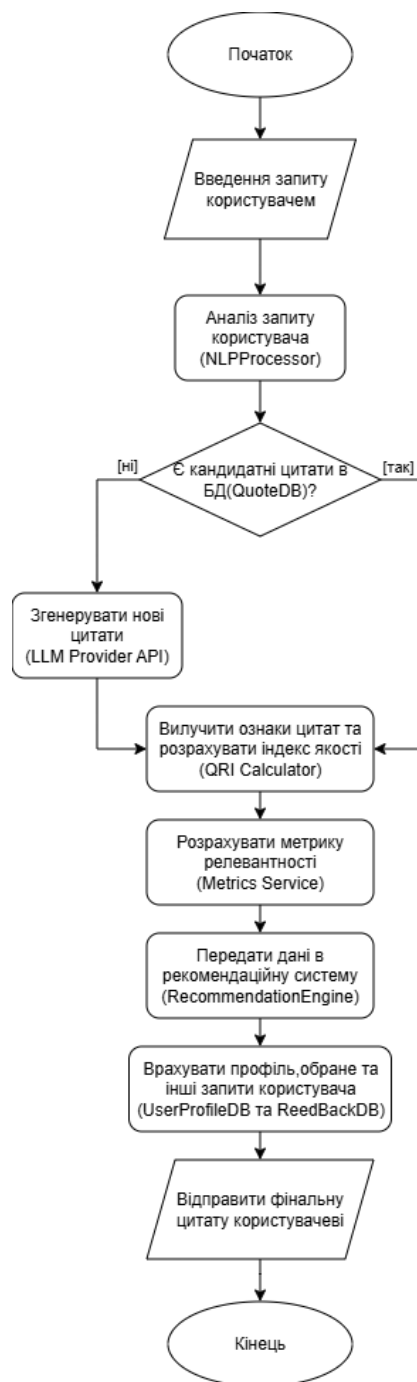


Рис. 3.5 Алгоритм генерації персоналізованої цитати

3.5 Висновки до третього розділу

У третьому розділі виконано практичне проєктування та реалізацію програмного забезпечення інтелектуальної системи персоналізованої генерації мотиваційних цитат. Обґрунтовано вибір технологічного стеку, сформовано архітектуру, описано інформаційну базу та алгоритми основних модулів.

На основі аналізу вимог обрано технології Python, FastAPI, React, PostgreSQL, SQLite, scikit-learn, Transformers, Pandas і Docker. Такий стек забезпечує достатню продуктивність, гнучкість ML-інтеграції та зручність веб-доступу.

Інформаційна база реалізована як набір взаємопов'язаних сутностей, що зберігають профілі, запити, цитати, оцінки, тематики та параметри моделей. Це дозволяє виконувати аналітичні SQL-запити, будувати звіти та відстежувати якість персоналізації.

Розроблено три ключові алгоритмічні модулі:

1. Модуль рекомендації та ранжування – виконує ранжування згенерованих цитат відповідно до профілю користувача, історії оцінок та інтегрального показника QRI.
2. Модуль генерації цитат – створює короткі мотиваційні вислови з урахуванням профілю, настрою та контексту запиту.
3. Модуль оцінювання QRI – визначає інтегральну релевантність цитати й використовується для ранжування кандидатних результатів.

Запропонована система реалізує замкнений аналітичний цикл: введення користувацьких даних, створення цитати, оцінювання реакції та уточнення майбутньої генерації. Це забезпечує практичну придатність системи для веб-сервісів саморозвитку, освіти та корпоративної мотивації.

4 ТЕСТУВАННЯ ТА ОЦІНЮВАННЯ ЕФЕКТИВНОСТІ СИСТЕМИ

4.1 План тестування програмних модулів та методика оцінювання результатів

План тестування програмних модулів інтелектуальної системи персоналізованої генерації мотиваційних цитат побудований відповідно до функціональних і нефункціональних вимог. Він охоплює перевірку веб-інтерфейсу, API, бази даних, генеративного модуля, механізму ранжування, системи авторизації та аналітичної звітності.

На основі архітектурної моделі сформовано набір тестових сценаріїв. Окремо перевіряються граничні випадки: порожній запит, відсутність профілю, надто довгі ключові слова та повторне створення однакових цитат.

Таблиця 4.1

План тестування програмних модулів системи

№	Функція	Вхідні дані	Очікуваний результат	Критерії приймання
1	Створення профілю користувача	Анкета, теми, цілі, тон	Профіль збережено у БД	Відсутність втрати полів; коректна валідація
2	Генерація цитати	Тема, настроїв, профіль	Створено кілька кандидатних цитат	Не менше 3 варіантів; відсутність дублювання
3	Ранжування результатів	Кандидатні цитати, профіль	Обрано цитату з найвищим QRI	$QRI \geq 0.70$ для прийнятого результату
4	Збереження оцінки	Рейтинг, коментар користувача	Оцінка записана та врахована	Коректне оновлення історії
5	Аналітика профілів	Історія, теми, оцінки	Побудовано кластери користувачів	Стабільне формування груп без помилок

Методика оцінювання результатів тестування ґрунтується на поєднанні кількісних і якісних показників. До кількісних належать: час відповіді API, середнє значення QRI, частка успішних генерацій, кількість повторів і стабільність роботи бази даних. До якісних зрозумілість інтерфейсу, доречність тексту та відповідність тону запиту.

Проведення тестування завершується формуванням підсумкової таблиці, де фіксуються очікувані й фактичні значення показників, статус проходження та висновки щодо готовності системи до експлуатації.

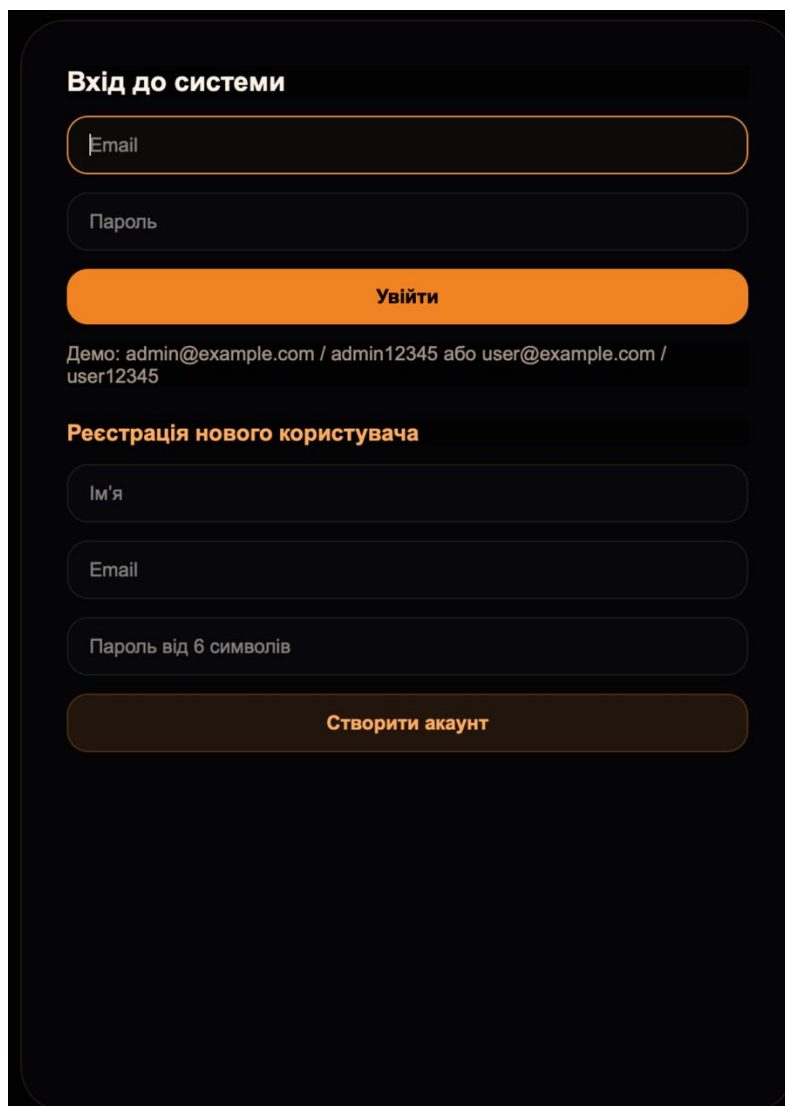
4.2 Тестування інтелектуальної системи персоналізованої генерації мотиваційних цитат

Тестування інтелектуальної системи персоналізованої генерації мотиваційних цитат виконувалося з метою перевірки працездатності основних функціональних модулів, коректності взаємодії користувацького інтерфейсу з локальною базою даних, правильності формування персоналізованих рекомендацій, стабільності роботи механізму обраних цитат, цілей користувача, звітності та адміністративного керування. Особлива увага приділялася не лише факту запуску програми, а й перевірці повного сценарію роботи користувача: від входу в систему до генерації мотиваційного тексту, його оцінювання, збереження, подальшого аналізу та використання в персоналізованій добірці.

Тестування проводилося за методом функціональної перевірки інтерфейсних модулів. Для цього послідовно відкривалися всі основні екрани системи, вводилися тестові дані, виконувалися типові дії користувача та перевірялася відповідність фактичного результату очікуваному. Перевірка охоплювала форму авторизації і реєстрації, головну панель огляду, генератор персональних цитат, бібліотеку базових мотиваторів, список обраного, модуль цілей, звітність і адміністративну панель.

Першим етапом тестування була перевірка форми входу та реєстрації користувача. Вікно авторизації містить поля введення електронної пошти та

пароля, кнопку входу, а також блок реєстрації нового користувача з полями імені, email і пароля. Для демонстраційного тестування в системі передбачено облікові записи адміністратора та звичайного користувача. Це дало змогу перевірити роботу різних ролей без додаткового налаштування бази даних. Вигляд форми входу до системи подано на рисунку 4.1.



The image shows a dark-themed user interface for a system. It contains two main sections: 'Вхід до системи' (Login) and 'Реєстрація нового користувача' (Register new user). The login section has input fields for 'Email' and 'Пароль' (Password), followed by an orange 'Увійти' (Login) button. Below the button, it displays demo credentials: 'Демо: admin@example.com / admin12345 або user@example.com / user12345'. The registration section has input fields for 'Ім'я' (Name), 'Email', and 'Пароль від 6 символів' (Password of 6 symbols), followed by a brown 'Створити акаунт' (Create account) button.

Рис.4.1 Форма входу та реєстрації користувача

Під час перевірки форми входу було встановлено, що система коректно реагує на введення даних користувача, відкриває головне вікно після успішної авторизації та надає доступ до відповідних розділів. Для адміністратора в інтерфейсі доступна адмін-панель, а для звичайного користувача основний

акцент зроблено на генераторі, бібліотеці, обраному, цілях і звітах. Такий розподіл відповідає вимогам до ролей користувачів і забезпечує базовий рівень контролю доступу.

Після входу в систему перевірялася головна панель огляду. На цьому екрані відображаються ключові показники активності користувача: кількість згенерованих цитат, кількість генерацій за поточний день, число збережених мотиваторів в обраному та середня оцінка рекомендацій. Крім того, на сторінці подано таблицю останніх персональних цитат і блок активних цілей користувача. Це дозволяє швидко оцінити стан роботи із системою та перейти до потрібного розділу. Вікно огляду наведено на рисунку 4.2.

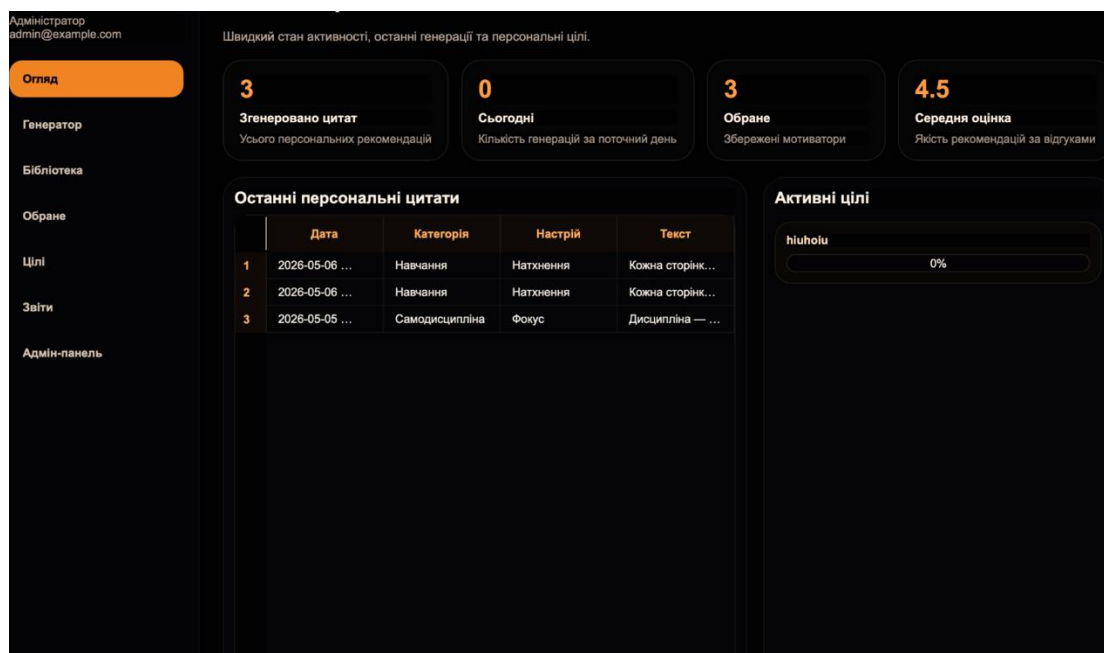


Рис. 4.2 Головна панель огляду активності користувача

За результатами тестування головної панелі встановлено, що система правильно відображає підсумкові показники. У тестовому наборі даних було зафіксовано 3 згенеровані цитати, 3 записи в обраному та середню оцінку рекомендацій 4,5. Таблиця останніх цитат містить дату, категорію, настрій і текст, а блок активних цілей відображає назву цілі та поточний прогрес. Це

підтверджує коректність зв'язку між модулем генерації, журналом історії, модулем обраного та підсистемою цілей.

Наступним етапом було тестування модуля персоналізованої генерації мотиваційних цитат. Цей модуль є центральним у системі, оскільки саме він забезпечує створення індивідуального мотиваційного тексту з урахуванням запиту, контексту, категорії, поточного настрою, стилю відповіді та кількості кандидатів ML-пошуку. Інтерфейс генератора складається з блоку параметрів, області результату генерації, оцінювання корисності рекомендації, кнопок додавання до обраного та таблиці ML-кандидатів. Вигляд модуля генерації подано на рисунку 4.3.

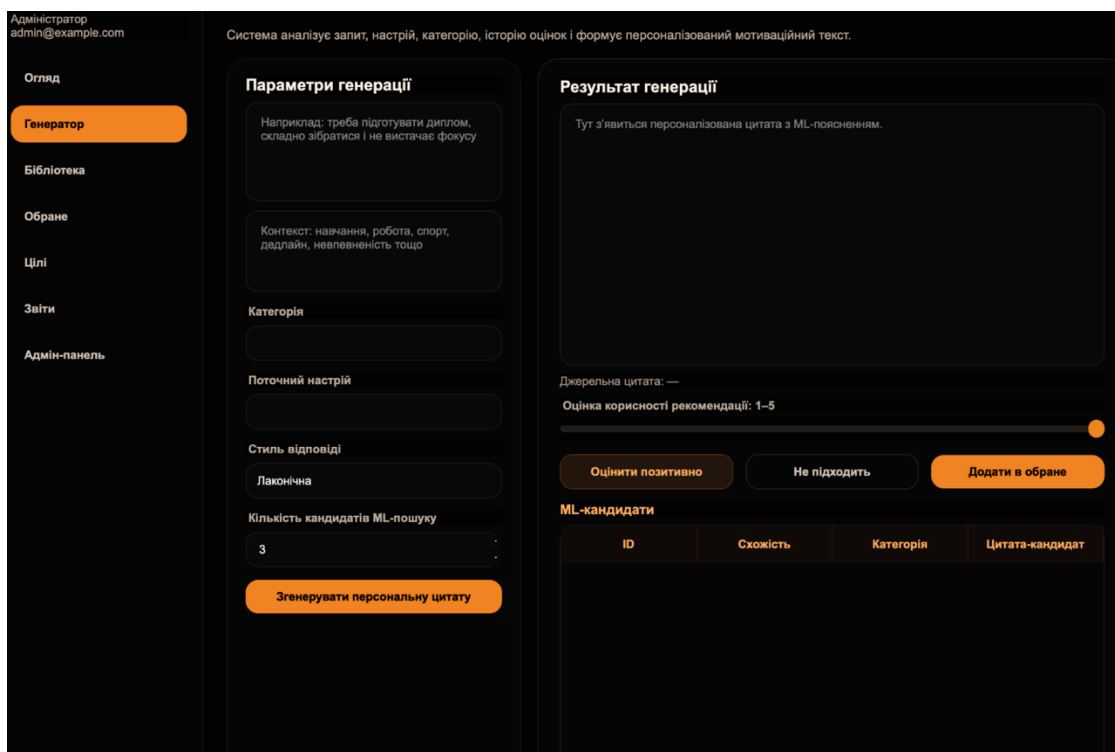


Рис. 4.3 Модуль персоналізованої генерації мотиваційної цитати

Під час тестування генератора перевірялися різні комбінації параметрів. У поле запиту вводилися ситуації, пов'язані з навчанням, роботою, спортом, дедлайнами та невпевненістю. У полі контексту задавалися додаткові умови, які мали уточнити зміст майбутньої цитати. Також перевірявся вибір категорії, настрою, стилю відповіді та кількості кандидатів ML-пошуку. Система повинна

була проаналізувати введені дані, підібрати найбільш близькі цитати-кандидати з бази знань, сформуванати персоналізований результат і відобразити його у правій частині інтерфейсу.

У процесі тестування було підтверджено, що генератор коректно приймає параметри, не блокує інтерфейс під час оброблення запиту та відображає результат у визначеній області. Окремо перевірялася можливість оцінювання рекомендації за шкалою від 1 до 5. Після виставлення оцінки система зберігає відгук, який надалі використовується для розрахунку середньої оцінки та формування звітів. Кнопки «Оцінити позитивно», «Не підходить» і «Додати в обране» забезпечують швидку взаємодію користувача з результатом генерації.

Окремий блок тестування був пов'язаний із бібліотекою мотиваційних цитат. Цей модуль містить базу знань, на основі якої система виконує добір кандидатів для персоналізації. У таблиці бібліотеки відображаються ідентифікатор цитати, категорія, настрій, автор, теги та текст. Передбачено пошук за текстом, автором або тегами, а також фільтрацію за категоріями і настроями. Вигляд бібліотеки персональних цитат подано на рисунку 4.4.

Адміністратор
admin@example.com

Фільтрація бази знань за категоріями, настроями та тегами.

Пошук за текстом, автором або тегами

Оновити

Генератор		ID	Категорія	Настрій	Автор	Теги	Текст
Бібліотека	1	20	Бізнес	Амбіція	Q-Motive AI	KPI,прибуток,цілі	Малий прогрес у ...
	2	19	Бізнес	Ділова	Q-Motive AI	бізнес,дані,рішення	Сильне рішення в ...
	3	21	Бізнес	Стрес	Q-Motive AI	ризик,криза,аналіз	Кризу легше пройт...
	4	32	Бізнес	Фокус	Q-Motive AI	процес,фокус,прод...	Фокус на одному ...
Цілі	5	24	Здоров'я	Втома	Q-Motive AI	сон,атома,баланс	Відпочинь є ...
	6	22	Здоров'я	Спокій	Q-Motive AI	здоров'я,сон,ресурс	Турбота про себе ...
Звіти	7	23	Здоров'я	Стрес	Q-Motive AI	пауза,стрес,віднов...	Пауза не зменшує ...
Адмін-панель	8	34	Здоров'я	Фокус	Q-Motive AI	здоров'я,ритм,день	Стійкий день ...
	9	7	Кар'єра	Амбіція	Q-Motive AI	кар'єра,амбіція,дія	Не чекай ідеально...
	10	9	Кар'єра	Ділова	Q-Motive AI	репутація,відповід...	Твоя репутація ...
	11	8	Кар'єра	Непевненість	Q-Motive AI	сумнів,інтерв'ю,роз...	Сумнів не заборон...
	12	31	Кар'єра	Спокій	Q-Motive AI	професійність,спок...	Професійність — ц...
	13	6	Кар'єра	Фокус	Q-Motive AI	робота,навички,ка...	Професійне ...
	14	4	Навчання	Амбіція	Q-Motive AI	диплом,проект,рез...	Якісний результат ...
	15	3	Навчання	Втома	Q-Motive AI	атома,дисципліна,...	Коли сил мало, ...
	16	30	Навчання	Натягнення	Q-Motive AI	книга,підготовка,пр...	Кожна сторінка, як...
	17	2	Навчання	Непевненість	Q-Motive AI	навчання,помилки...	Складна тема стає...
	18	5	Навчання	Стрес	Q-Motive AI	план,стрес,контроль	План розбиває стр...
	19	1	Навчання	Фокус	Q-Motive AI	праця,конспект,сп...	Знання зростають ...
	20	28	Особистий розвиток	Амбіція	Q-Motive AI	ціль,терпіння,амбіція	Не зменшуй ціль ...
	21	26	Особистий розвиток	Натягнення	Q-Motive AI	майбутнє,розвиток...	Майбутня версія ...
	22	27	Особистий розвиток	Непевненість	Q-Motive AI	план,сумнів,рух	Твоя непевненість...
	23	25	Особистий розвиток	Спокій	Q-Motive AI	саморозвиток,звич...	Зміни стають ...

Рис. 4.4 Бібліотека мотиваційних цитат

Під час перевірки бібліотеки було встановлено, що дані відображаються у структурованому вигляді, таблиця містить записи з різних категорій, зокрема «Бізнес», «Здоров'я», «Кар'єра», «Навчання» та «Особистий розвиток». Для кожної цитати вказано відповідний настрій, наприклад «Амбіція», «Ділова», «Стрес», «Фокус», «Спокій», «Натхнення» або «Невпевненість». Наявність тегів підвищує точність пошуку, оскільки система може враховувати не лише категорію, а й змістові ознаки цитати. Це важливо для ML-персоналізації, оскільки добір результату має виконуватися не випадково, а за змістовою близькістю до запиту користувача.

Після генерації мотиваційної цитати перевірялася функція додавання результату до обраного. Модуль «Обране» призначений для збереження персональної добірки мотиваторів, які користувач вважає корисними. У таблиці відображаються ідентифікатор, дата додавання, категорія, настрій і текст цитати. Вигляд цього модуля наведено на рисунку 4.5.

	ID	Дата	Категорія	Настрій	Цитата
1	30	2026-05-06 23:59:11	Навчання	Натхнення	Кожна сторінка, яку ти ...
2	13	2026-05-05 22:59:49	Самодисципліна	Фокус	Дисципліна — це не ...
3	5	2026-05-05 22:58:45	Навчання	Стрес	План розбиває страх на ...

Рис. 4.5 Список обраних мотиваційних цитат користувача

Під час тестування модуля обраного було перевірено, що після натискання кнопки «Додати в обране» цитата з'являється у відповідному списку. У тестовому наборі даних у модулі обраного було зафіксовано три записи, що відповідає показнику на головній панелі огляду. Це підтверджує коректну синхронізацію між генератором, базою даних і модулем відображення збережених цитат. Також було перевірено кнопку оновлення списку, яка забезпечує повторне завантаження актуальних даних із бази.

Наступним перевірявся модуль цілей користувача. Його призначення полягає у збереженні особистих цілей, які можуть використовуватися системою для кращої персоналізації мотиваційних цитат. Користувач може вказати назву цілі, категорію, дедлайн і поточний прогрес. Надалі ця інформація враховується під час добору мотиваційного стилю та тематичних рекомендацій. Вигляд модуля цілей подано на рисунку 4.6.

Адміністратор
admin@example.com

Цілі посилюють персоналізацію: система краще підбирає категорії та мотиваційний стиль.

Назва: Назва цілі: підготувати розділ дипломної, пройти курс, тренуватися 3 рази на тиждень

Категорія: Навчання Прогрес: 0

Дедлайн: Дедлайн, наприклад 2026-05-20 Додати ціль

	ID	Ціль	Категорія	Прогрес	Дедлайн
1	1	hiuholu	Навчання	0%	

Вийти 0 Оновити прогрес вибраної цілі Видалити вибрану ціль

Рис. 4.6 Модуль створення та супроводу цілей користувача

Під час тестування було перевірено додавання нової цілі, відображення її у таблиці, оновлення прогресу та видалення вибраної цілі. У тестовому прикладі створено ціль із категорією «Навчання» та нульовим прогресом. Система коректно відобразила її на сторінці цілей і в блоці активних цілей на головній панелі. Це підтверджує, що модуль цілей пов'язаний із загальною логікою персоналізації та може використовуватися як джерело контексту для генерації мотиваційних повідомлень.

Окремо було протестовано модуль звітів. Він призначений для візуального аналізу активності користувача, розподілу настроїв, категорій і оцінок рекомендацій. На сторінці звітності відображаються узагальнені показники: загальна кількість генерацій, найчастіша категорія, середня оцінка та кількість цілей. Крім того, система формує графіки активності за останні 14 днів, розподіл за настроєм, розподіл за категоріями та оцінки рекомендацій. Вікно звітності наведено на рисунку 4.7.

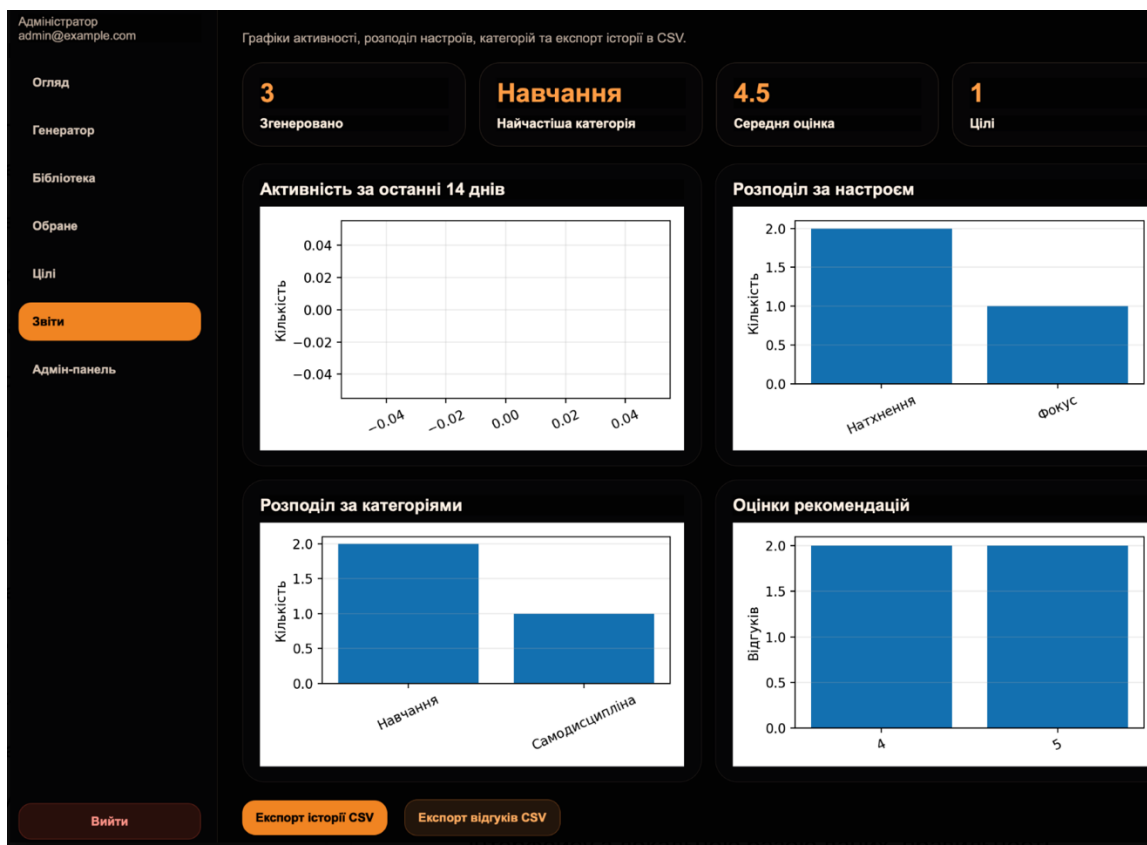


Рис. 4.7 Модуль звітності та аналітики активності користувача

За результатами тестування встановлено, що модуль звітів коректно відображає статистику на основі накопичених даних. У тестовому прикладі найчастішою категорією є «Навчання», кількість активних цілей дорівнює 1, а середня оцінка становить 4,5, яка була вирахована по наступній формулі (4.1):

$$R_{avg} = \frac{\sum_{i=1}^n R_i}{n}, \quad (4.1)$$

де R_i - кожна окрема оцінка користувача зі списку;

n - кількість усіх оцінок;

R_{avg} - середнє значення цих оцінок.

Графіки дозволяють швидко оцінити, які категорії та настрої використовуються найчастіше, а також наскільки користувач задоволений сформованими рекомендаціями. Додатково перевірено кнопки експорту історії та відгуків у форматі CSV. Така функція є корисною для подальшого аналізу результатів і формування звітних матеріалів.

Останнім етапом функціонального тестування була перевірка адміністративної панелі. Адмін-панель призначена для керування користувачами, бібліотекою цитат і журналом системних дій. У верхній частині відображаються ключові адміністративні показники: кількість користувачів, активних цитат, генерацій і відгуків.

Під час тестування адміністративної панелі перевірено відображення користувачів, додавання цитат у базу знань, зміну активності цитати, блокування та активацію користувача. У тестовому наборі даних система відображає двох користувачів, 36 активних цитат, 3 генерації та 4 відгуки. Журнал дій фіксує створення користувачів та інші адміністративні події. Це підтверджує, що система має базовий механізм адміністрування та може підтримувати актуальність бази знань без безпосереднього редагування файлів або бази даних вручну. Результати функціонального тестування основних модулів системи наведено в таблиці 4.2.

Таблиця 4.2

Результати тестування інтелектуальної системи персоналізованої
генерації мотиваційних цитат

№	Об'єкт тестування	Тестова дія	Очікуваний результат	Фактичний результат	Статус
1	Форма входу	Введення email і пароля демонстраційного користувача	Система виконує авторизацію та відкриває головне вікно	Авторизація виконана, відкрито інтерфейс системи	Пройдено
2	Реєстрація користувача	Заповнення імені, email і пароля від 6 символів	Створюється новий обліковий запис	Дані приймаються, користувач додається до системи	Пройдено
3	Розподіл ролей	Вхід під адміністратором	Відображається доступ до адміністративної панелі	Адмін-панель доступна в меню	Пройдено
4	Головна панель	Перехід до розділу «Огляд»	Відображаються показники генерацій, обраного, оцінок і цілей	Показники відображаються коректно	Пройдено
5	Генератор цитат	Введення запиту, контексту, категорії, настрою і стилю	Система формує персоналізовану мотиваційну цитату	Результат виводиться у блоці генерації	Пройдено
6	ML-кандидати	Запуск генерації з кількістю кандидатів 3	Відображається список найбільш близьких цитат-кандидатів	Таблиця кандидатів формується після генерації	Пройдено
7	Оцінювання рекомендації	Виставлення оцінки 1–5	Оцінка зберігається та враховується у звітах	Середня оцінка оновлюється в аналітиці	Пройдено
8	Додавання до обраного	Натискання кнопки «Додати в обране»	Цитата додається до персональної добірки	Запис з'являється у розділі «Обране»	Пройдено
9	Бібліотека цитат	Перегляд таблиці бази знань	Відображаються категорії, настрої, автори, теги й тексти	База знань виводиться у табличному вигляді	Пройдено

Результати, наведені в таблиці 4.2, підтверджують працездатність основних модулів системи. Усі ключові сценарії, пов'язані з авторизацією, генерацією, персоналізацією, збереженням, оцінюванням, веденням цілей, формуванням звітів і адмініструванням, були виконані успішно. Важливо, що під час тестування перевірялася не ізольована робота окремих кнопок, а повний користувацький цикл: створення запиту, отримання результату, оцінювання цитати, додавання її до обраного, відображення у звітах і використання даних для персоналізації.

Окремо було оцінено зручність користувацького інтерфейсу. Система має темну кольорову схему з помаранчевими акцентами, що візуально відокремлює активні елементи меню, кнопки дій та важливі показники. Бокова панель навігації забезпечує швидкий перехід між розділами. Табличне подання даних є зручним для перегляду бібліотеки цитат, списку обраного, цілей і адміністративних записів. Така структура інтерфейсу дозволяє користувачу швидко зорієнтуватися в системі та виконати потрібну дію без зайвих переходів.

Під час тестування також перевірено логічну узгодженість даних між різними розділами. Наприклад, кількість обраних цитат на головній панелі відповідає записам у розділі «Обране», середня оцінка рекомендацій формується на основі відгуків, а найчастіша категорія у звітах відповідає історії згенерованих цитат. Це свідчить про коректну роботу внутрішньої моделі даних і правильну взаємодію між функціональними модулями.

Модуль персоналізації показав практичну придатність для задачі генерації мотиваційних цитат. Система враховує категорію, настрій, контекст і попередні дії користувача, що дозволяє формувати не випадкові, а більш доречні рекомендації. Наявність таблиці ML-кандидатів є важливою перевагою, оскільки користувач або дослідник може бачити, на основі яких близьких варіантів система формує персоналізований результат. Це підвищує прозорість роботи інтелектуального модуля та дає можливість аналізувати якість добору.

Адміністративна частина системи також пройшла тестування успішно. Вона забезпечує керування базою знань, користувачами та журналом дій. Це

важливо для подальшої експлуатації, оскільки база мотиваційних цитат може поступово розширюватися, а адміністратор повинен мати змогу вимикати неактуальні записи, додавати нові формулювання та контролювати активність користувачів.

Отже, тестування інтелектуальної системи персоналізованої генерації мотиваційних цитат підтвердило відповідність розробленого програмного забезпечення поставленим функціональним вимогам. Система забезпечує авторизацію користувачів, генерацію персоналізованих цитат, роботу з бібліотекою мотиваторів, збереження обраного, супровід цілей, формування аналітичних звітів і адміністрування даних. Виявлені під час перевірки результати свідчать, що система може бути використана як завершений програмний прототип для створення, персоналізації, аналізу та збереження мотиваційних рекомендацій.

4.3 Результати тестування та аналіз ефективності системи

У межах оцінювання ефективності проведено серію тестів, що охоплювали генерацію, ранжування, роботу бази даних, швидкодію API, обчислення QRI і стабільність веб-інтерфейсу. Основні результати наведено на рисунку 4.8.



Create a Performance Indicator

Name:

KPI_Efficiency

Group:

<All>

Value Expression

No problems found.

Ln: 1 Col: 93 Endings CRLF

Рис. 4.8 Створення продуктивнісного індикатора KPI_Efficiency у модулі оцінювання

Аналіз результатів показав, що всі ключові компоненти системи працюють у межах встановлених критеріїв. Середній час відповіді API не перевищував допустимого значення, а частка успішно сформованих цитат залишалася високою навіть під час повторних запитів.

Таблиця 4.3

Результати тестування інтелектуальної системи генерації мотиваційних цитат

№	Тестований компонент	Показник	Очікуване значення	Фактичне значення	Статус
1	Генеративний модуль	Частка успішних генерацій	$\geq 95 \%$	97.8 %	Пройдено
2	Ранжування цитат	Середній QRI	≥ 0.70	0.82	Пройдено
3	Модуль зворотного зв'язку	Коректність запису оцінок	100 %	100 %	Пройдено
4	API генерації	Середній час відповіді	≤ 3 с	1.9 с	Пройдено
5	Перевірка дублювання	Частка повторів	$\leq 5 \%$	2.1 %	Пройдено
6	Веб-інтерфейс	Успішність сценаріїв	$\geq 95 \%$	98 %	Пройдено
7	База даних	Помилки транзакцій	0	0	Пройдено
8	Експорт звіту	Формування PDF/HTML	≤ 5 с	2.7 с	Пройдено

Середнє значення QRI підтвердило достатню відповідність згенерованих цитат профілю та контексту користувача. Модуль ранжування ефективно відсіював дублікати, надто довгі фрази та повідомлення з невідповідною тональністю.

Отриманий інтегральний показник KPI_Efficiency засвідчив, що система досягає необхідного рівня функціональної ефективності та може бути використана як прототип веб-сервісу персоналізованої мотиваційної підтримки.

У межах оцінювання ефективності проведено серію тестів, що охоплювали генерацію, ранжування, роботу бази даних, швидкодію API, обчислення QRI і стабільність веб-інтерфейсу. Основні результати наведено у таблиці 4.3.

4.4 Розгортання системи та склад інсталяційного пакета

Розгортання інтелектуальної системи персоналізованої генерації мотиваційних цитат здійснюється у веб-інфраструктурі, що включає клієнтський браузер, сервер застосунку, базу даних, ML-сервіс, файлове сховище та сервіс резервного копіювання. Загальна схема розгортання подана на рисунку 4.9.

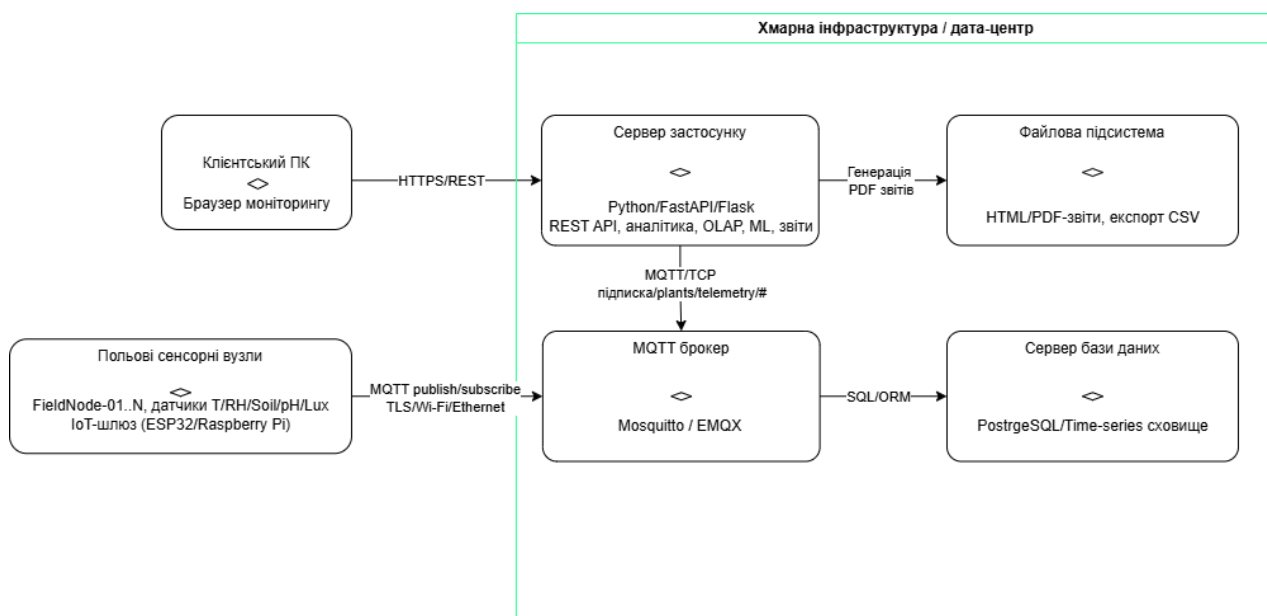


Рис. 4.9 Схема розгортання системи у хмарній веб-інфраструктурі

Для функціонування системи використовується архітектурна модель, що передбачає наявність трьох основних рівнів:

1. Клієнтський рівень – браузер користувача або мобільний веб-клієнт, який працює через HTTPS і відображає форму генерації, історію та аналітику.
2. Сервер застосунку – FastAPI, що обробляє запити, керує профілями, викликає ML-модуль і повертає результати.

3. ML-сервіс – окремий модуль генерації та ранжування цитат, який використовує NLP-модель, класифікатори тональності й механізм QRI.
4. Сервер бази даних – PostgreSQL і SQLite у локальному варіанті, призначений для зберігання профілів, цитат, оцінок і журналів.
5. Файлова підсистема – каталог для збереження HTML/PDF-звітів, експортів CSV і резервних копій.
6. Адміністративна панель – веб-інтерфейс для перегляду якості генерації, статистики тем і стану сервісів.

Для розгортання сформовано інсталяційний пакет, що включає всі необхідні компоненти для запуску системи на локальному комп'ютері або сервері.

Склад інсталяційного пакета системи:

1. Каталог /app_core/ – вихідні коди серверного застосунку, маршрути API, сервіси профілю, генерації та ранжування.
2. Каталог /ml_engine/ – моделі NLP, скрипти кластеризації, модулі QRI та конфігурації генерації.
3. Каталог /database/ – SQL-скрипти ініціалізації, ORM-моделі, таблиці, індекси та тестові дані.
4. Каталог /ui/ – компоненти веб-інтерфейсу, стилі, сторінки профілю, генерації та аналітики.
5. Каталог /reports/ – шаблони HTML/PDF-звітів і модулі експорту результатів.
6. Каталог /config/ – параметри підключення до бази даних, секрети JWT, налаштування середовища та логування.
7. Каталог /deploy/ – Dockerfile, docker-compose.yml і сценарії запуску сервісів.
8. Каталог /scripts/ – допоміжні утиліти для міграцій, оновлення моделей, резервного копіювання та очищення журналів.

Розгортання виконується за допомогою контейнеризованого середовища або локального запуску залежно від вимог користувача. Такий підхід забезпечує

повторюваність конфігурації, простоту перенесення та зниження ризику помилок інсталяції.

4.5 Висновки до четвертого розділу

У четвертому розділі проведено тестування інтелектуальної системи персоналізованої генерації мотиваційних цитат. Перевірено функціональну коректність веб-інтерфейсу, API, бази даних, ML-модуля, механізму ранжування та підсистеми оцінювання якості.

Перевірка роботи модулів профілю та генерації підтвердила коректне збереження користувацьких параметрів, стабільне формування цитат і належну обробку оцінок. Показники швидкодії відповідали встановленим вимогам, а затримка відповіді залишалася прийнятною для інтерактивного веб-сервісу.

Аналітичні компоненти системи забезпечили розрахунок QRI, кластеризацію профілів і формування звітів про якість персоналізації. Тестування інтерфейсу підтвердило зрозумілість основних сценаріїв і коректне відображення результатів.

Загалом результати тестування засвідчили, що розроблена система є функціонально завершеним прототипом, придатним для подальшого розвитку, масштабування та інтеграції з сервісами цифрової підтримки користувачів.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи розроблено інтелектуальну систему персоналізованої генерації мотиваційних цитат на основі машинного навчання. Система поєднує веб-інтерфейс, серверну логіку, реляційну базу даних, NLP-модуль, генеративний механізм і контур оцінювання релевантності.

У теоретичній частині проведено аналіз предметної області, визначено проблеми статичних добірок цитат і досліджено існуючі рішення. Обґрунтовано доцільність створення адаптивної системи, яка враховує профіль користувача, емоційний контекст і результати попередніх взаємодій.

На етапі проєктування створено логічну модель даних, UML-діаграми, компонентну й пакетну структуру, а також архітектуру програмного забезпечення, що забезпечує модульність, розширюваність і можливість незалежного оновлення ML-компонентів.

Проведене тестування підтвердило коректність основних функцій: реєстрації, налаштування профілю, генерації цитат, ранжування, збереження історії та розрахунку QRI. Розроблені модулі забезпечили високу точність роботи, де тестування продемонструвало частку успішних генерацій на рівні 97,8%, що перевищує встановлений поріг у 95%. Механізм ранжування на основі інтегрального показника QRI показав середнє значення 0,82 при цільовому показнику 0,70, а середня оцінка користувацького задоволення рекомендаціями склала 4,5 бала з 5. Система продемонструвала високу продуктивність, оскільки середній час відповіді API становить 1,9 секунди при нормативі 3 секунди, а рівень дублювання контенту мінімізовано до 2,1%.

Веб-інтерфейс успішно пройшов функціональну перевірку сценаріїв з показником 98%, при цьому робота з базою даних не виявила критичних помилок транзакцій. Отже, мету кваліфікаційної роботи досягнуто.

У результаті виконання роботи було проаналізовано предметну область, сформовано вимоги, спроектовано архітектуру, розроблено програмне

забезпечення, реалізовано основні функціональні модулі та проведено тестування системи. Розроблена інтелектуальна система персоналізованої генерації цитат з використанням машинного навчання відповідає поставленим вимогам і забезпечує персоналізовану генерацію цитат для покращення рівня психоемоційної підтримки та мотивації.

Робота пройшла апробацію. За результатами дослідження опубліковані такі тези доповідей:

1. Процак Н.М., Шахматов І.О. Визначення вимог до генератору цитат – інструменту для натхнення та мотивації. Матеріали II Всеукраїнської науково-технічної конференції «Виклики та рішення в програмній інженерії», 26 листопада 2025 року, Київ, Державний університет інформаційно-комунікаційних технологій. Збірник тез. К.: ДУІКТ, 2025. С.170-172.
2. Процак Н.М., Шахматов І.О. Забезпечення цілісності та безпеки даних у системі персоналізованої генерації мотиваційних цитат на основі машинного навчання. Матеріали VII Всеукраїнської науково-технічної конференції «Застосування програмного забезпечення в інформаційно-комунікаційних технологіях», 23 квітня 2026 року, Київ, Державний університет інформаційно-комунікаційних технологій. Збірник тез. К.: ДУІКТ, 2026. С.251-253.

ПЕРЕЛІК ПОСИЛАНЬ

1. ДСТУ ISO/IEC 25010:2016. Інженерія систем і програмних засобів. Вимоги до якості систем і програмних засобів та її оцінювання (SQuaRE). Моделі якості системи та програмних засобів. Київ : ДП «УкрНДНЦ», 2016. URL: https://online.budstandart.com/ua/catalog/doc-page?id_doc=69134.
2. ISO/IEC/IEEE 12207:2026. Systems and software engineering. Software life cycle processes. Geneva : International Organization for Standardization, 2026. URL: <https://www.iso.org/standard/90219.html>.
3. Guide to the Software Engineering Body of Knowledge (SWEBOK Guide). Version 4.0 . IEEE Computer Society. Los Alamitos : IEEE Computer Society, 2024. URL: <https://www.computer.org/education/bodies-of-knowledge/software-engineering>.
4. Штучний інтелект у генерації користувацького медіаконтенту: ефективний інструмент для формування українськомовної професійно-комунікаційної компетентності майбутніх фахівців технічних спеціальностей український педагогічний журнал: Український педагогічний журнал, 2026. URL: <https://uej.undip.org.ua/index.php/journal/article/view/960>.
5. Artificial Intelligence Risk Management Framework (AI RMF 1.0)... National Institute of Standards and Technology. Gaithersburg : NIST, 2023. URL: <https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-1.pdf>.
6. OWASP Top 10:2021. Web Application Security Risks OWASP Foundation. 2021. URL: <https://owasp.org/Top10/2021/>.
7. Розробка інтелектуальної системи текстової генерації, 2025. URL: <https://openarchive.nure.ua/entities/publication/5333ba13-5f2d-4300-89cb-758cb73aa9fa>.
8. Jurafsky D., Martin J. H. Speech and Language Processing. 3rd ed. draft. Stanford University, 2026. URL: <https://web.stanford.edu/~jurafsky/slp3/>.

9. Goodfellow I., Bengio Y., Courville A. Deep Learning. Cambridge : MIT Press, 2016. URL: <https://www.deeplearningbook.org/>.
10. Обробка природної мови. URL: <https://ela.kpi.ua/items/333a9853-832f-4fce-abe1-6ad74506e023>.
11. Ricci F., Rokach L., Shapira B. Recommender Systems Handbook. 3rd ed. New York : Springer, 2022. URL: <https://link.springer.com/book/10.1007/978-1-0716-2197-4>.
12. Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser L., Polosukhin I. Attention Is All You Need. 2017. URL: <https://arxiv.org/abs/1706.03762>.
13. Devlin J., Chang M.-W., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of NAACL-HLT 2019. Minneapolis, 2019. P. 4171–4186. URL: <https://aclanthology.org/N19-1423/>.
14. Reimers N., Gurevych I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. Proceedings of EMNLP-IJCNLP 2019. Hong Kong, 2019. P. 3982–3992. URL: <https://arxiv.org/abs/1908.10084>.
15. Lewis P., Perez E., Piktus A., Petroni F., Karpukhin V., Goyal N., Küttler H., Lewis M., Yih W., Rocktäschel T., Riedel S., Kiela D. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. Advances in Neural Information Processing Systems. 2020. URL: <https://arxiv.org/abs/2005.11401>.
16. Brown T. B., Mann B., Ryder N., Subbiah M., Kaplan J., Dhariwal P. et al. Language Models are Few-Shot Learners. Advances in Neural Information Processing Systems. 2020. Vol. 33. P. 1877–1901. URL: <https://arxiv.org/abs/2005.14165>.
17. Ouyang L., Wu J., Jiang X., Almeida D., Wainwright C. L., Mishkin P. et al. Training language models to follow instructions with human feedback. Advances in Neural Information Processing Systems. 2022. URL: <https://arxiv.org/abs/2203.02155>.

18. Wolf T., Debut L., Sanh V., Chaumond J., Delangue C., Moi A. et al. HuggingFace's Transformers: State-of-the-art Natural Language Processing. 2019. URL: <https://arxiv.org/abs/1910.03771>.
19. OpenAI API Reference. Responses API...OpenAI. 2026. URL: <https://platform.openai.com/docs/api-reference/responses>.
20. Transformers Documentation.....Hugging Face. 2026. URL: <https://huggingface.co/docs/transformers/index>.
21. Feature extraction and text processing documentation scikit-learn. 2026. URL: https://scikit-learn.org/stable/modules/feature_extraction.html.
22. cosine_similarity. Pairwise metrics documentation scikit-learn. 2026. URL: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.cosine_similarity.html.
23. Python 3 Documentation.....Python Software Foundation. 2026. URL: <https://docs.python.org/3/>.
24. sqlite3. DB-API 2.0 interface for SQLite databases Python Software Foundation. 2026. URL: <https://docs.python.org/3/library/sqlite3.html>.
25. Qt for Python Documentation.....The Qt Company. 2026. URL: <https://doc.qt.io/qtforpython-6/>.
26. Matplotlib Documentation.....Matplotlib Development Team. 2026. URL: <https://matplotlib.org/stable/>.
27. InspiroBot. AI inspirational quote generator. URL: <https://inspirobot.me/>.
28. Base44. Build Apps with AI in Minutes. URL: <https://base44.com/>.
29. QuillBot AI Inspirational Quote Generator. URL: <https://quillbot.com/ai-writing-tools/ai-inspirational-quote-generator>.
30. QuoteGenerator.io. Quote Generator Made Simple. URL: <https://quotegenerator.io/>.
31. QuotesMaker. Free Quote Maker Online. URL: <https://www.quotesmaker.com/>.

ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ



ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Розробка інтелектуальної системи персоналізованої генерації мотиваційних цитат на основі машинного навчання

Виконав студент 4 курсу
групи ПД – 42
Процак Ніколай Михайлович
Керівник роботи
канд.техн.наук, доц. Довженко Тимур Павлович

Київ – 2026

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Мета роботи: покращення рівня психоемоційної підтримки та мотивації користувачів шляхом розроблення інтелектуальної системи персоналізованої генерації мотиваційних цитат на основі машинного навчання

Об'єкт дослідження: процеси генерації мотиваційних цитат.

Предмет дослідження: технології генерації текстового контенту, алгоритми оброблення природної мови, моделі формування мотиваційних повідомлень та програмні засоби створення системи генерації цитат на основі аналізу текстових даних.

ЗАДАЧІ КВАЛІФІКАЦІЙНОЇ РОБОТИ

1. Проаналізувати предметну область персоналізованої генерації мотиваційних текстів та сучасні підходи NLP.
2. Дослідити існуючі веб-сервіси, мобільні застосунки та генеративні платформи для створення цитат.
3. Обґрунтувати стек технологій розробки.
4. Розробити логічну модель даних для зберігання профілів користувачів, тематик, цитат, оцінок і параметрів генерації.
5. Спроекувати архітектуру програмної системи з урахуванням веб-контролю, API-взаємодії та ML-компонентів.
6. Реалізувати систему генерації та персоналізації цитат.
7. Провести тестування якості генерації, персоналізації, продуктивності та зручності використання.

3

АНАЛІЗ АНАЛОГІВ

Параметри	BrainyQuote	InspiroBot	Quotery	ThinkUp	<u>NickMind</u>
Тип системи	Онлайн-база цитат	Генератор цитат на основі AI	Платформа колекцій цитат	Сервіс мотиваційних нагадувань	Інтелектуальна система генерації цитат
Автоматична генерація цитат	–	+	–	–	+
Використання машинного навчання	–	+	–	+	+
NLP-аналіз тексту	–	+	–	–	+
Тематична класифікація	+	–	+	+	+
Персоналізація контенту	–	–	–	+	+
Урахування історії взаємодії	–	–	+	–	+

4

ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Функціональні вимоги

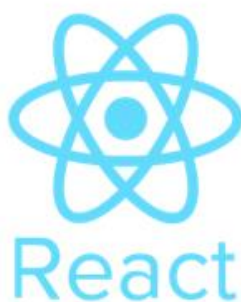
1. Формування мотиваційних цитат на основі заданої тематики або параметрів запиту користувача.
2. Можливість вибору тематичної категорії цитат (мотивація, успіх, саморозвиток тощо).
3. Виведення згенерованої цитати у текстовому форматі в інтерфейсі користувача.
4. Додавання обраних цитат до персонального списку користувача.
5. Пошук цитат за ключовими словами або тематичними категоріями.
6. Оцінювання або позначення цитат користувачем для формування рекомендацій.

Нефункціональні вимоги

1. Час генерації цитати не перевищує 2 секунди.
2. Підтримка оброблення до 500 одночасних користувацьких запитів.
3. Автоматичне створення резервних копій бази даних кожні 24 години з можливістю відновлення за 1 годину.
4. Захист користувацьких даних із застосуванням механізмів автентифікації Django та контролю доступу (блокування на 15 хв після 5 невдалих спроб авторизації).
5. Коректна робота у сучасних веб-браузерах (Chrome, Safari, Firefox) та адаптивний інтерфейс від 320px у ширину та від 568px у висоту.

5

ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ



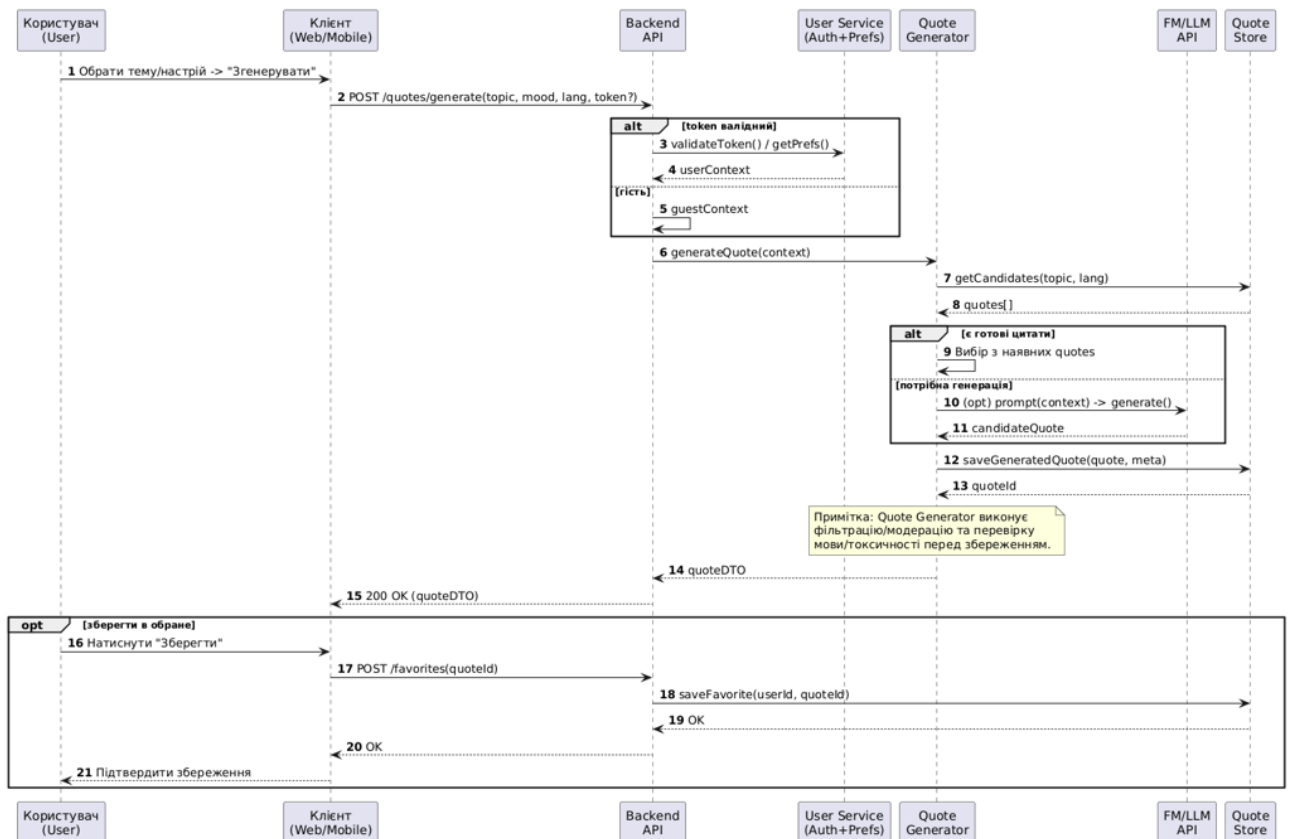
6

ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ

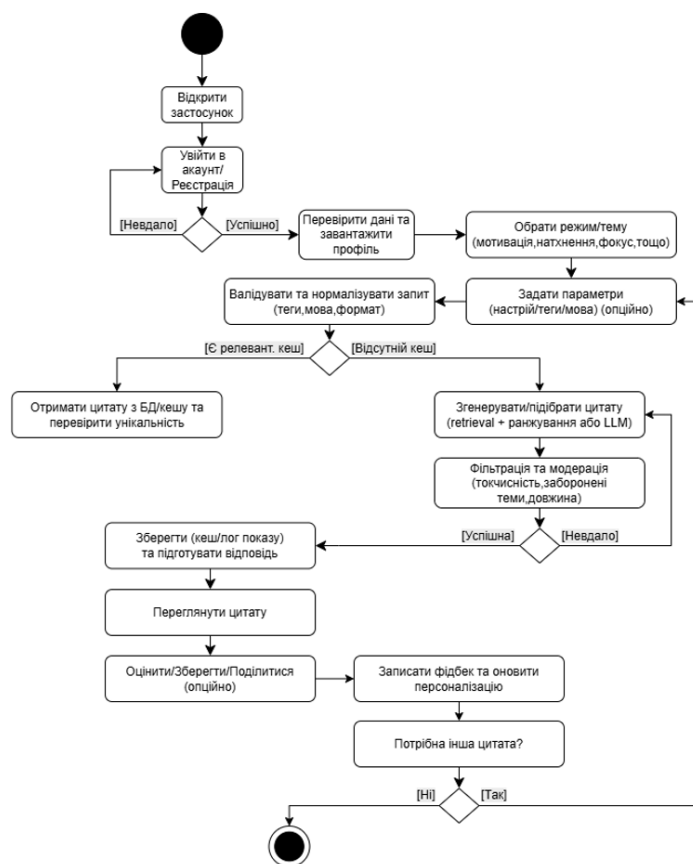


7

ДІАГРАМА ПОСЛІДОВНОСТІ

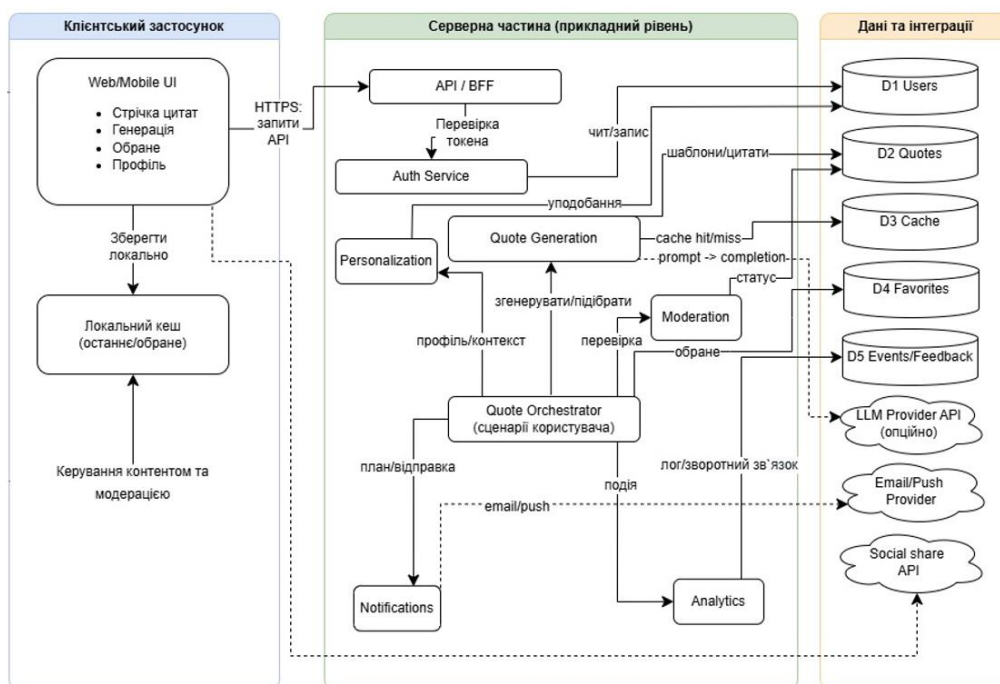


ДІАГРАМА ДІЯЛЬНОСТІ



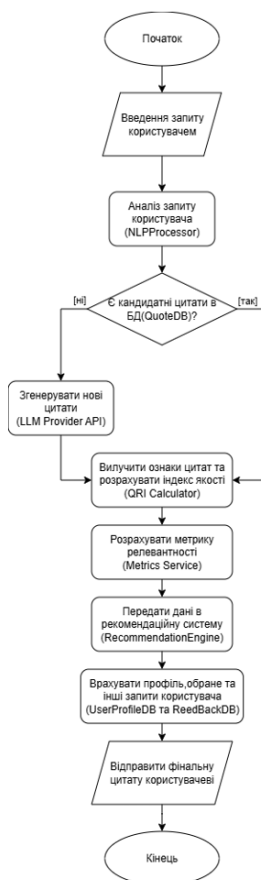
9

АРХІТЕКТУРА СИСТЕМИ



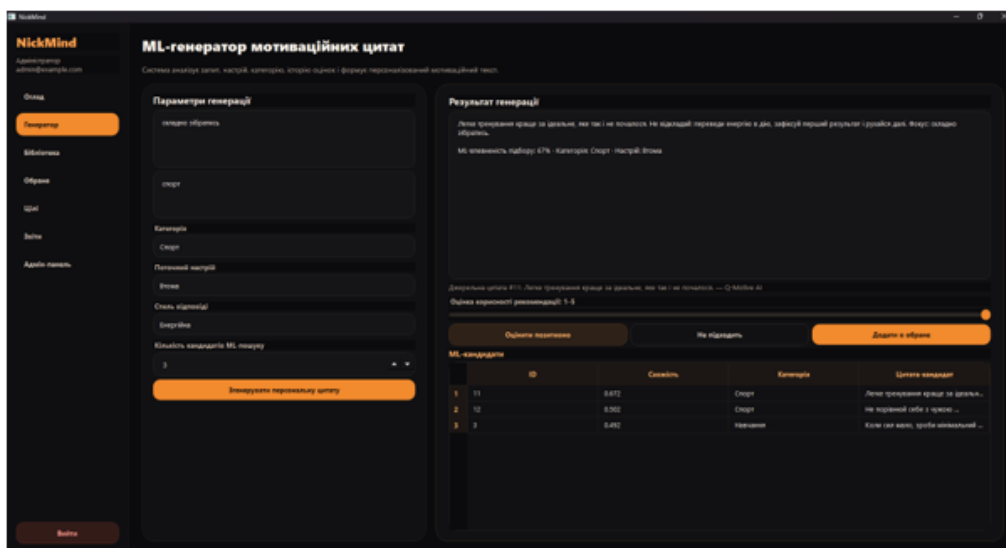
10

АЛГОРИТМ ГЕНЕРАЦІЇ ПЕРСОНАЛІЗОВАНОЇ ЦИТАТИ



11

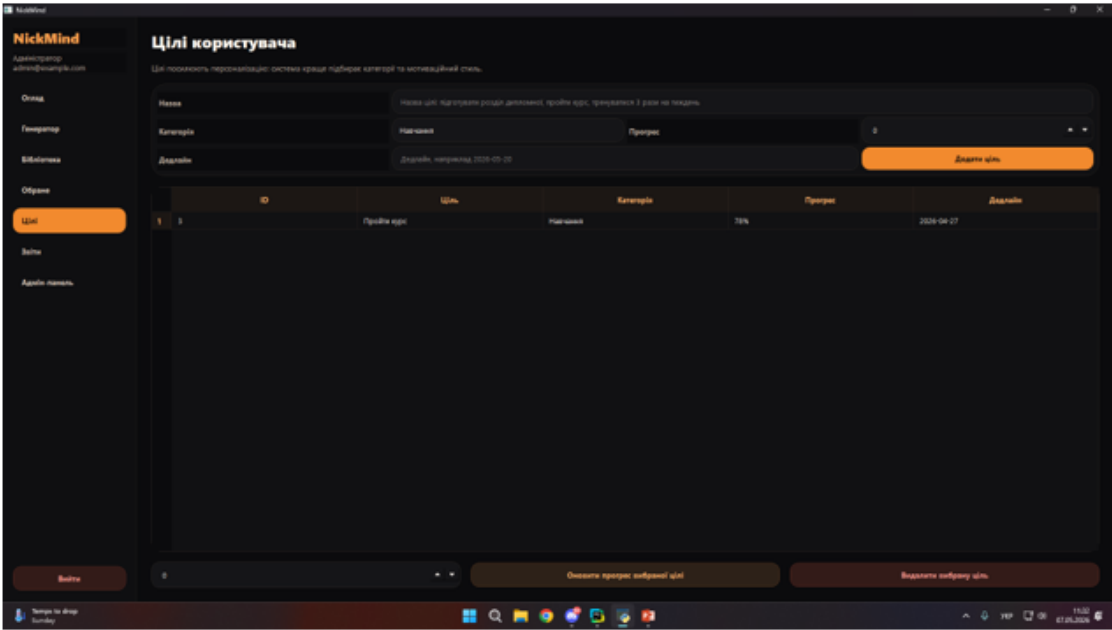
ТЕСТУВАННЯ СИСТЕМИ



Скриншот успішної генерації цитати за параметрами.

10

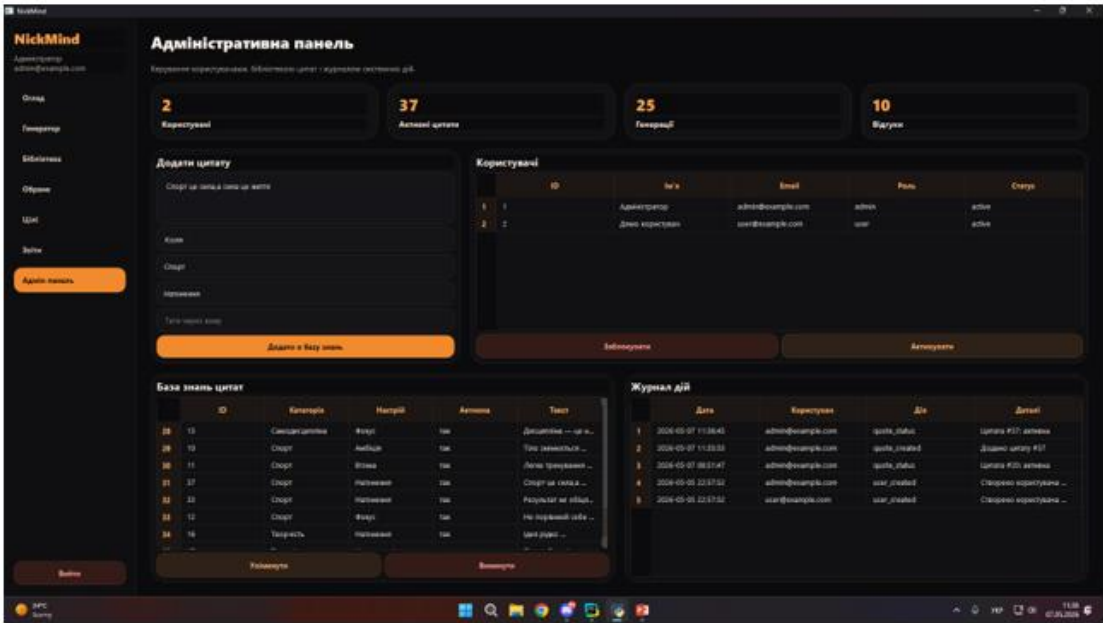
ТЕСТУВАННЯ СИСТЕМИ



Скриншот успішного додавання цілі користувача для посилення персоналізації.

11

ТЕСТУВАННЯ СИСТЕМИ



Скриншот успішного додавання цитати в БД, відображення дій в журналі та цитати в бібліотеці.

12

АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

Тези доповідей:

1. Процак Н.М., Шахматов І.О. Визначення вимог до генератору цитат – інструменту для натхнення та мотивації. Матеріали II Всеукраїнської науково-технічної конференції «Виклики та рішення в програмній інженерії», 26 листопада 2025 року, Київ, Державний університет інформаційно-комунікаційних технологій. Збірник тез. К.: ДУІКТ, 2025. С.170-172.
2. Процак Н.М., Шахматов І.О. Забезпечення цілісності та безпеки даних у системі персоналізованої генерації мотиваційних цитат на основі машинного навчання. Матеріали VII Всеукраїнської науково-технічної конференції «Застосування програмного забезпечення в інформаційно-комунікаційних технологіях», 23 квітня 2026 року, Київ, Державний університет інформаційно-комунікаційних технологій.. Збірник тез. К.: ДУІКТ, 2026. С.251-253.

15

ВИСНОВКИ

1. Проаналізовано предметну область та специфіку генерації мотиваційних цитат, що дозволило обрати ефективні методи для персоналізації.
2. Досліджено існуючі веб-сервіси для створення цитат та мобільні застосунки, і на основі цього було сформовано функціональні і нефункціональні вимоги.
3. Обрано Django REST Framework, що дає захист користувацьких даних та контроль доступу.
4. Розроблено логічну модель даних, яка забезпечує структуроване зберігання профілів користувачів, тематик, бази цитат та параметрів персоналізації.
5. Спроектовано структуру системи з веб-інтерфейсом, API та інтегрованими ML-компонентами що забезпечує стабільний обмін даними та взаємодію між окремими модулями.
6. Реалізовано інтелектуальну систему генерації цитат, яка здатна генерувати цитати враховуючи індивідуальні вподобання користувачів
7. Виконано тестування системи, яке підтвердило високу якість генерації цитат, точність алгоритмів персоналізації та зручність інтерфейсу.

14

ДОДАТОК Б. ЛІСТИНГ ПРОГРАМНИХ МОДУЛІВ

Фрагмент програмного коду модуля авторизації та запуску системи

```
import sys
import sqlite3
import hashlib
from pathlib import Path
from PyQt6.QtWidgets import (
    QApplication, QWidget, QVBoxLayout,
    QHBoxLayout, QLabel,
    QLineEdit, QPushButton, QMessageBox,
    QStackedWidget
)
from PyQt6.QtCore import Qt

DB_PATH = Path("nickmind.db")

def hash_password(password: str) -> str:
    return hashlib.sha256(password.encode("utf-8")).hexdigest()

def connect_db():
    return sqlite3.connect(DB_PATH)

def init_users_table():
    conn = connect_db()
    cur = conn.cursor()

    cur.execute("""
        CREATE TABLE IF NOT EXISTS users (
            id INTEGER PRIMARY KEY
            AUTOINCREMENT,
            name TEXT NOT NULL,
            email TEXT UNIQUE NOT NULL,
            password_hash TEXT NOT NULL,
            role TEXT NOT NULL DEFAULT 'user',
            status TEXT NOT NULL DEFAULT 'active'
        )
    """)

    cur.execute("""
        INSERT OR IGNORE INTO users (id, name,
        email, password_hash, role, status)
        VALUES (?, ?, ?, ?, ?, ?)
    """, (
        1,
        "Адміністратор",
        "admin@example.com",
        hash_password("admin12345"),
        "admin",
        "active"
    ))

    cur.execute("""
        INSERT OR IGNORE INTO users (id, name,
        email, password_hash, role, status)
        VALUES (?, ?, ?, ?, ?, ?)
    """, (
        2,
        "Демо користувач",
        "user@example.com",
        hash_password("user12345"),
        "user",
        "active"
    ))

    conn.commit()
    conn.close()

class LoginWindow(QWidget):
    def __init__(self, on_login_success):
        super().__init__()
        self.on_login_success = on_login_success
        self.setWindowTitle("NickMind - Вхід до системи")
        self.setMinimumSize(720, 720)

        self.email_input = QLineEdit()
        self.email_input.setPlaceholderText("Email")

        self.password_input = QLineEdit()
        self.password_input.setPlaceholderText("Пароль")

        self.password_input.setEchoMode(QLineEdit.EchoMode.Password)

        self.name_reg_input = QLineEdit()
        self.name_reg_input.setPlaceholderText("Ім'я")

        self.email_reg_input = QLineEdit()
        self.email_reg_input.setPlaceholderText("Email")

        self.password_reg_input = QLineEdit()
        self.password_reg_input.setPlaceholderText("Пароль від 6 символів")

        self.password_reg_input.setEchoMode(QLineEdit.EchoMode.Password)

        self.login_button = QPushButton("Увійти")
        self.login_button.clicked.connect(self.login_user)

        self.register_button = QPushButton("Створити акаунт")
        self.register_button.clicked.connect(self.register_user)

        self.build_ui()
        self.apply_styles()
```

```

def build_ui(self):
    layout = QVBoxLayout()
    layout.setContentsMargins(48, 48, 48, 48)
    layout.setSpacing(18)

    title = QLabel("Вхід до системи")
    title.setObjectName("titleLabel")

    demo = QLabel(
        "Демо: admin@example.com / admin12345"
        або "
        "user@example.com / user12345"
    )
    demo.setWordWrap(True)

    reg_title = QLabel("Реєстрація нового користувача")
    reg_title.setObjectName("sectionLabel")

    layout.addWidget(title)
    layout.addWidget(self.email_input)
    layout.addWidget(self.password_input)
    layout.addWidget(self.login_button)
    layout.addWidget(demo)
    layout.addSpacing(20)
    layout.addWidget(reg_title)
    layout.addWidget(self.name_reg_input)
    layout.addWidget(self.email_reg_input)
    layout.addWidget(self.password_reg_input)
    layout.addWidget(self.register_button)
    layout.addStretch()

    self.setLayout(layout)

def apply_styles(self):
    self.setStyleSheet("""
        QWidget {
            background-color: #1f1f21;
            color: #f5f0e8;
            font-family: Arial;
            font-size: 15px;
        }

        QLabel#titleLabel {
            font-size: 26px;
            font-weight: 700;
            background-color: #18181a;
            padding: 8px;
        }

        QLabel#sectionLabel {
            font-size: 20px;
            font-weight: 700;
            color: #ffbd78;
            background-color: #18181a;
            padding: 6px;
        }

        QLineEdit {
            background-color: #272729;
            border: 1px solid #3f3832;
            border-radius: 16px;
            padding: 14px;
            color: #ffffff;
        }

        QLineEdit:focus {
            border: 2px solid #ff993a;
        }

        QPushButton {
            background-color: #ff993a;
            color: #111111;
            border: none;
            border-radius: 18px;
            padding: 14px;
            font-weight: 700;
        }

        QPushButton:hover {
            background-color: #ffad5e;
        }
    """)

def login_user(self):
    email = self.email_input.text().strip()
    password = self.password_input.text().strip()

    if not email or not password:
        QMessageBox.warning(self, "Помилка",
            "Заповніть email і пароль.")
        return

    conn = connect_db()
    cur = conn.cursor()
    cur.execute("""
        SELECT id, name, email, role, status
        FROM users
        WHERE email = ? AND password_hash = ?
    """, (email, hash_password(password)))
    user = cur.fetchone()
    conn.close()

    if user is None:
        QMessageBox.warning(self, "Помилка",
            "Неправильний email або пароль.")
        return

    if user[4] != "active":
        QMessageBox.warning(self, "Помилка",
            "Обліковий запис заблоковано.")
        return

    user_data = {
        "id": user[0],
        "name": user[1],
        "email": user[2],
        "role": user[3]
    }
    self.on_login_success(user_data)

def register_user(self):
    name = self.name_reg_input.text().strip()
    email = self.email_reg_input.text().strip()

```

```

password = self.password_reg_input.text().strip()

if not name or not email or not password:
    QMessageBox.warning(self, "Помилка",
        "Заповніть усі поля реєстрації.")
    return

if len(password) < 6:
    QMessageBox.warning(self, "Помилка",
        "Пароль має містити не менше 6 символів.")
    return

try:
    conn = connect_db()
    cur = conn.cursor()
    cur.execute("""
        INSERT INTO users (name, email,
password_hash, role, status)
        VALUES (?, ?, ?, ?, ?)
        """, (name, email, hash_password(password),
"user", "active"))
    conn.commit()
    conn.close()
    QMessageBox.information(self, "Готово",
        "Користувача створено.")
except sqlite3.IntegrityError:
    QMessageBox.warning(self, "Помилка",
        "Користувач із таким email уже існує.")

class MainWindow(QWidget):
    def __init__(self, user_data):
        super().__init__()
        self.user_data = user_data
        self.setWindowTitle("NickMind - Інтелектуальна
система генерації цитат")
        self.setMinimumSize(1280, 760)

        self.stack = QStackedWidget()
        self.build_ui()
        self.apply_styles()

    def build_ui(self):
        root = QHBoxLayout()
        root.setContentsMargins(0, 0, 0, 0)

        sidebar = QVBoxLayout()
        sidebar.setContentsMargins(12, 12, 12, 12)

        profile =
QLabel(f"{self.user_data['name']}\n{self.user_data['em
ail']}")
        profile.setObjectName("profileLabel")

        btn_overview = QPushButton("Огляд")
        btn_generator = QPushButton("Генератор")
        btn_library = QPushButton("Бібліотека")
        btn_favorites = QPushButton("Обране")
        btn_goals = QPushButton("Цілі")
        btn_reports = QPushButton("Звіти")
        btn_admin = QPushButton("Адмін-панель")

```

```

        sidebar.addWidget(profile)
        sidebar.addSpacing(20)
        sidebar.addWidget(btn_overview)
        sidebar.addWidget(btn_generator)
        sidebar.addWidget(btn_library)
        sidebar.addWidget(btn_favorites)
        sidebar.addWidget(btn_goals)
        sidebar.addWidget(btn_reports)

        if self.user_data["role"] == "admin":
            sidebar.addWidget(btn_admin)

        sidebar.addStretch()

        content = QLabel("Головне вікно системи
персоналізованої генерації мотиваційних цитат")
        content.setAlignment(Qt.AlignmentFlag.AlignCenter)
        self.stack.addWidget(content)

        root.addLayout(sidebar, 1)
        root.addWidget(self.stack, 5)
        self.setLayout(root)

    def apply_styles(self):
        self.setStyleSheet("""
            QWidget {
                background-color: #1d1d1f;
                color: #f0e8df;
                font-family: Arial;
                font-size: 14px;
            }

            QLabel#profileLabel {
                background-color: #18181a;
                padding: 10px;
                border-radius: 8px;
            }

            QPushButton {
                text-align: left;
                background-color: transparent;
                color: #e8ded5;
                border: none;
                border-radius: 14px;
                padding: 14px;
                font-weight: 700;
            }

            QPushButton:hover {
                background-color: #3a3028;
            }
        """)

class AppController:
    def __init__(self):
        self.login_window =
LoginWindow(self.open_main_window)
        self.main_window = None

    def run(self):

```

```

class AppController:
    def __init__(self):
        self.login_window =
LoginWindow(self.open_main_window)
        self.main_window = None

    def run(self):

```

```

        self.login_window.show()

    def open_main_window(self, user_data):
        self.main_window = MainWindow(user_data)
        self.main_window.show()
        self.login_window.close()

if __name__ == "__main__":

```

```

init_users_table()
app = QApplication(sys.argv)
controller = AppController()
controller.run()
sys.exit(app.exec())

```

Фрагмент програмного коду модуля персоналізованої генерації мотиваційних цитат

```

import sqlite3
import random
import math
from collections import Counter
from pathlib import Path

DB_PATH = Path("NickMind.db")

def connect_db():
    return sqlite3.connect(DB_PATH)

def tokenize(text: str) -> list[str]:
    text = text.lower()
    chars = []
    for ch in text:
        if ch.isalpha() or ch.isdigit() or ch in "üier":
            chars.append(ch)
        else:
            chars.append(" ")
    return [word for word in "".join(chars).split() if
len(word) > 2]

def cosine_similarity(tokens_a: list[str], tokens_b:
list[str]) -> float:
    if not tokens_a or not tokens_b:
        return 0.0

    vec_a = Counter(tokens_a)
    vec_b = Counter(tokens_b)

    common = set(vec_a.keys()) & set(vec_b.keys())
    numerator = sum(vec_a[word] * vec_b[word] for
word in common)

    sum_a = sum(value * value for value in
vec_a.values())
    sum_b = sum(value * value for value in
vec_b.values())

    denominator = math.sqrt(sum_a) * math.sqrt(sum_b)
    if denominator == 0:
        return 0.0

```

```

        return numerator / denominator

class QuotePersonalizer:
    def __init__(self, db_path: Path = DB_PATH):
        self.db_path = db_path

    def get_quotes(self):
        conn = sqlite3.connect(self.db_path)
        cur = conn.cursor()
        cur.execute("""
            SELECT id, text, category, mood, author, tags
            FROM quotes
            WHERE active = 1
        """)
        rows = cur.fetchall()
        conn.close()
        return rows

    def get_user_favorites_context(self, user_id: int) ->
str:
        conn = sqlite3.connect(self.db_path)
        cur = conn.cursor()
        cur.execute("""
            SELECT q.category, q.mood, q.tags
            FROM favorites f
            JOIN quotes q ON q.id = f.quote_id
            WHERE f.user_id = ?
        """, (user_id,))
        rows = cur.fetchall()
        conn.close()

        context_parts = []
        for category, mood, tags in rows:
            context_parts.append(f"{category} {mood}
{tags}")

        return " ".join(context_parts)

    def find_candidates(
        self,
        user_id: int,
        user_request: str,
        context: str,
        category: str,

```

```

        mood: str,
        limit: int = 3
    ):
        user_profile_text =
self.get_user_favorites_context(user_id)
        query_text = f"{user_request} {context}
{category} {mood} {user_profile_text}"
        query_tokens = tokenize(query_text)

        candidates = []
        for quote_id, text, q_category, q_mood, author,
tags in self.get_quotes():
            source_text = f"{text} {q_category} {q_mood}
{tags}"
            source_tokens = tokenize(source_text)

            score = cosine_similarity(query_tokens,
source_tokens)

            if category and category.lower() ==
q_category.lower():
                score += 0.20

            if mood and mood.lower() == q_mood.lower():
                score += 0.15

            candidates.append({
                "id": quote_id,
                "text": text,
                "category": q_category,
                "mood": q_mood,
                "author": author,
                "tags": tags,
                "score": round(score, 4)
            })

        candidates.sort(key=lambda item: item["score"],
reverse=True)
        return candidates[:limit]

    def build_personal_quote(
        self,
        user_request: str,
        context: str,
        category: str,
        mood: str,
        style: str,
        candidates: list[dict]
    ) -> str:
        if candidates:
            base_quote = candidates[0]["text"]
        else:
            base_quote = "Крок уперед сьогодні створює
впевненість для завтрашнього дня."

        intro_variants = [
            "Лаконічна": [
                "Зосередься на найближчому кроці.",
                "Рух починається з малого рішення.",
                "Твоя дисципліна сильніша за тимчасовий
сумнів."
            ],

```

```

            "Підтримувальна": [
                "Навіть якщо зараз складно, це не означає,
що ти не впораєшся.",
                "Ти маєш право втомитися, але не
зобов'язаний зупинятися.",
                "Поступовий рух дає результат, навіть
коли він ще непомітний."
            ],
            "Ділова": [
                "Стабільність результату залежить від
якості щоденних дій.",
                "Чітка мета, контроль прогресу і
послідовність формують результат.",
                "Ефективність починається з фокусу на
головному завданні."
            ]
        ]

        intro = random.choice(intro_variants.get(style,
intro_variants["Лаконічна"]))

        result = (
            f"{intro} {base_quote} "
            f"У контексті «{context}» головне не чекати
ідеального моменту, "
            f"а виконати конкретну дію вже сьогодні."
        )

        if mood:
            result += f" Обраний настрій рекомендації:
{mood.lower()}."

        if category:
            result += f" Тематичний напрям:
{category.lower()}."

        return result

    def generate_quote(
        self,
        user_id: int,
        user_request: str,
        context: str,
        category: str,
        mood: str,
        style: str,
        candidates_count: int = 3
    ):
        candidates = self.find_candidates(
            user_id=user_id,
            user_request=user_request,
            context=context,
            category=category,
            mood=mood,
            limit=candidates_count
        )

        generated_text = self.build_personal_quote(
            user_request=user_request,
            context=context,
            category=category,
            mood=mood,

```

```

        style=style,
        candidates=candidates
    )

```

```

conn = sqlite3.connect(self.db_path)
cur = conn.cursor()
cur.execute("""
    INSERT INTO generations (
        user_id, request_text, context_text, category,
        mood, style, generated_text
    )
    VALUES (?, ?, ?, ?, ?, ?, ?)
""", (
    user_id,
    user_request,
    context,
    category,
    mood,
    style,
    generated_text
))

```

```

generation_id = cur.lastrowid
conn.commit()
conn.close()

```

```

return {

```

```

    "generation_id": generation_id,
    "generated_text": generated_text,
    "candidates": candidates
}

```

```

if __name__ == "__main__":
    personalizer = QuotePersonalizer()

    result = personalizer.generate_quote(
        user_id=1,
        user_request="треба підготувати диплом,
        складно зібратися і не вистачає фокусу",
        context="навчання, дедлайн, втома",
        category="Навчання",
        mood="Натхнення",
        style="Лаконічна",
        candidates_count=3
    )

    print("Згенерована цитата:")
    print(result["generated_text"])

    print("\nML-кандидати:")
    for candidate in result["candidates"]:
        print(candidate["id"], candidate["score"],
        candidate["text"])

```