

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ЮРИДИЧНА АКАДЕМІЯ»
ФАКУЛЬТЕТ КІБЕРБЕЗПЕКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра інформаційних технологій

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

**«ІНФОРМАЦІЙНО-АНАЛІТИЧНА СИСТЕМА НАВЧАННЯ АНГЛІЙСЬКОЇ
МОВИ З ВИКОРИСТАННЯМ ІНТЕЛЕКТУАЛЬНОГО ЧАТ-БОТА»**

Виконав: здобувач 2 курсу

Рівень магістр

Галузь знань 12 «Інформаційні технології»

Спеціальність 124 «Системний аналіз»

Безродний Микола Олександрович

Керівник: к.т.н., доцент

Трофименко Олена Григорівна

Рецензент: к.пед.н., доцент

Логінова Наталія Іванівна

Одеса – 2025

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ОДЕСЬКА ЮРИДИЧНА АКАДЕМІЯ”

Факультет кібербезпеки та інформаційних технологій

Кафедра кібербезпеки/інформаційних технологій

Галузь знань: 12 Інформаційні технології

Спеціальність: 124 “Системний аналіз”

Назва освітньої програми: Аналіз та безпека даних

ЗАТВЕРДЖУЮ

Завідувач кафедри інформаційних технологій
доцент Наталія Логінова

“ ” 2025 року

ЗАВДАННЯ

на кваліфікаційну (магістерську) роботу
здобувачу Безродному Миколі Олександровичу

- Тема роботи: «Інформаційно-аналітична система навчання англійської мови з використанням інтелектуального чат-бота»
Керівник роботи: к.т.н, доцент Трофименко О.Г.
затверджені наказом НУ «ОЮА» від «10» жовтня 2025 року № 3183-06.
- Строк подання здобувачем роботи «25» листопада 2025 року.
- Дата видачі завдання «02» червня 2025 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1.	Вибір теми, вивчення літературних джерел та складання плану роботи	15.09.2025	
2.	Підготовка першого розділу роботи та подання його керівнику	01.10.2025	
3.	Підготовка другого розділу роботи та подання його керівнику	14.10.2025	
4.	Підготовка третього розділу роботи та подання його керівнику	07.11.2025	
5.	Підготовка вступу та висновків	10.11.2025	
6.	Доопрацювання роботи з урахуванням зауважень керівника	24.11.2025	
7.	Подача електронного варіанта роботи для перевірки на плагіат	25.11.2025	

Здобувач

(підпис)

М.О. Безродний

(ім'я та прізвище)

Керівник роботи

(підпис)

О.Г. Трофименко

(ім'я та прізвище)

АНОТАЦІЯ

Кваліфікаційна робота на тему «Інформаційно-аналітична система навчання англійської мови з використанням інтелектуального чат-бота» на здобуття другого (магістерського) рівня вищої освіти за спеціальністю 124 «Системний аналіз», освітньою програмою «Аналіз та безпека даних». Робота містить 22 рисунки, 7 таблиць, 3 додатки, 41 літературне джерело. Робота виконана на 69 сторінках основного тексту і 78 сторінках загального тексту.

Метою роботи є дослідження та розробка інформаційно-аналітичної системи (ІАС) навчання англійської мови на основі інтелектуального чат-бота.

Об'єктом дослідження є ІАС як об'єкт системного аналізу в освітньому середовищі.

Предметом дослідження є методи та програмні засоби, що забезпечують діалогову взаємодію, генерацію навчального контенту, семантичну релевантність відповідей та аналітичний супровід навчального процесу.

У першому розділі проведено системний аналіз сучасних підходів до цифрової автоматизації навчання англійської мови, класифікацію освітніх чат-ботів та визначення вимог до ІАС у мовній освіті. У другому розділі здійснено моделювання та проєктування ІАС, визначено її архітектуру, технологічний стек та методи оцінки ефективності. У третьому розділі реалізовано прототип інтелектуального чат-бота, описано функціональні сценарії його роботи, проведено експериментальну перевірку, оцінено ефективність навчального процесу та здійснено порівняння з іншими освітніми системами.

ІНФОРМАЦІЙНО-АНАЛІТИЧНА СИСТЕМА; ІНТЕЛЕКТУАЛЬНИЙ ЧАТ-БОТ; СИСТЕМНИЙ АНАЛІЗ; ОБРОБКА ПРИРОДНОЇ МОВИ (NLP); ВЕЛИКІ МОВНІ МОДЕЛІ (LLM); SENTENCETRANSFORMER; СЕМАНТИЧНА ВІДПОВІДНІСТЬ; BLEU; BERTSCORE; ЛАТЕНТНІСТЬ ВІДПОВІДІ; LMS-ІНТЕГРАЦІЯ.

ABSTRACT

Qualification work on the topic "Information and analytical system for teaching English using an intelligent chatbot" for obtaining the second (master's) level of higher education in the specialty 124 "Systems Analysis", educational program "Data Analysis and Security". The work contains 22 figures, 7 tables, 3 appendices, 41 references. The work is performed on 69 pages of the main text and 78 pages of the general text.

The purpose of the work is to research and develop an information-analytical system of the information-analytical system (IAS) for teaching English based on an intelligent chatbot.

The object of the research is the IAS as an object of system analysis in the educational environment.

The subject of the research is methods and software that provide dialogic interaction, generation of educational content, semantic relevance of responses and analytical support of the educational process.

The first section provides a systematic analysis of modern approaches to digital automation of English language learning, a classification of educational chatbots, and the definition of requirements for IAS in language education. The second section provides modeling and design of IAS, defines its architecture, technological stack, and methods for assessing effectiveness. The third section implements a prototype of an intelligent chatbot, describes its functional scenarios, conducts experimental testing, evaluates the effectiveness of the learning process, and compares it with other educational systems.

INFORMATION-ANALYTICAL SYSTEM; INTELLIGENT CHATBOT; SYSTEM ANALYSIS; NATURAL LANGUAGE PROCESSING (NLP); LARGE LANGUAGE MODELS (LLM); SENTENCETRANSFORMER; SEMANTIC MATCHING; BLEU; BERTSCORE; RESPONSE LATENCY; LMS INTEGRATION.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ.....	7
ВСТУП.....	8
1 ІНФОРМАЦІЙНО-АНАЛІТИЧНІ СИСТЕМИ У СФЕРІ МОВНОЇ ОСВІТИ .	10
1.1 Сучасні підходи до цифрової автоматизації навчання англійської мови...	10
1.2 Класифікація освітніх чат-ботів та їх функціональні моделі	13
1.2.1 Типи освітніх чат-ботів.....	14
1.2.2 Порівняння архітектурних підходів	15
1.2.3 Моделі діалогової взаємодії в освітньому контексті	17
1.3 Інформаційно-аналітичні системи для вивчення мов: структура, функції, вимоги.....	19
1.4 Технології обробки природної мови в навчальних системах	21
1.5 Висновки до розділу 1	25
2 МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ІНФОРМАЦІЙНО- АНАЛІТИЧНОЇ СИСТЕМИ НАВЧАННЯ АНГЛІЙСЬКОЇ МОВИ	27
2.1 Постановка задачі: вимоги до системи, цільова аудиторія, функціональні та нефункціональні характеристики	27
2.2 Системний аналіз процесу навчання: формалізація моделей знань, сценаріїв взаємодії та когнітивної динаміки	29
2.3 Архітектура системи: компоненти, взаємозв'язки, середовище реалізації	35
2.4 Вибір технологій: платформи для чат-ботів, NLP-інструменти, бази даних, інтеграція з LMS	41
2.5 Методи оцінки ефективності системи.....	44
2.6 Висновки до розділу 2	46
3 РЕАЛІЗАЦІЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ІАС НАВЧАННЯ АНГЛІЙСЬКОЇ МОВИ.....	48
3.1 Реалізація системи: середовище, інструменти та технології.....	48
3.2 Функціональні сценарії роботи системи.....	51

	6
3.3 Експериментальна перевірка функціональності системи	57
3.4 Оцінка ефективності навчального процесу	60
3.5 Порівняння з наявними освітніми системами.....	61
3.6 Висновки до розділу 3	62
ВИСНОВКИ.....	64
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	66
ДОДАТКИ	70
Додаток А. Фрагменти програмного коду для реалізації бекенду ІАС.....	70
Додаток Б. Фрагменти js-коду для фронтенду.....	74
Додаток В. Копії документів про апробацію результатів роботи	74

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

AI – Artificial Intelligence

API – Application Programming Interface

BERT – Bidirectional Encoder Representations from Transformers

GPT – Generative Pre-trained Transformer

JSON – JavaScript Object Notation, об'єктна нотація JavaScript

LLM – Large Language Models, великі мовні моделі

LMS – системи управління навчанням

ML – Machine Learning, машинне навчання

NLP – Natural Language Processing, обробка природної мови

REST API – Representational State Transfer, стиль архітектури вебсервісів
(представницький перенос стану)

UML – Unified Modeling Language, уніфікована мова моделювання

UI – User Interface, інтерфейс користувача

IAC – інформаційно-аналітична система

ПЗ – програмне забезпечення

ІІ – штучний інтелект

ВСТУП

Цифрова трансформація освіти зумовлює потребу у створенні інтелектуальних систем, здатних забезпечувати персоналізоване навчання, інтерактивну взаємодію та аналітичний супровід. У сфері вивчення англійської мови особливої актуальності набувають чат-боти, які поєднують діалогову форму навчання з когнітивною підтримкою та дозволяють адаптувати контент до рівня знань користувача. Використання інформаційно-аналітичних систем з інтеграцією NLP-моделей та аналітичних модулів відкриває нові можливості для підвищення ефективності мовної освіти, що робить тему дослідження сучасною та практично значущою.

Метою роботи є дослідження та розробка інформаційно-аналітичної системи (ІАС) навчання англійської мови на основі інтелектуального чат-бота.

Для досягнення цієї мети визначено такі завдання: здійснити системний аналіз сучасних освітніх чат-ботів та інформаційно-аналітичних систем; розробити архітектуру та технологічний стек прототипу; реалізувати функціональні сценарії взаємодії користувача із системою; провести експериментальну перевірку функціональності та оцінити ефективність навчального процесу за допомогою системних метрик; здійснити порівняння з наявними освітніми платформами та визначити перспективи розвитку.

Предметом дослідження є методи та програмні засоби, які забезпечують діалогову взаємодію, генерацію навчального контенту, семантичну релевантність відповідей та аналітичну підтримку процесу навчання англійської мови.

Об'єктом дослідження є інформаційно-аналітична система як об'єкт системного аналізу в освітньому середовищі.

У роботі використано методи системного аналізу, архітектурного моделювання, побудови UML-діаграм, обробки природної мови (NLP), алгоритми машинного навчання для семантичної оцінки відповідей (SentenceTransformer, BLEU, BERTScore), а також методи експериментальної перевірки та порівняльної оцінки ефективності.

Наукова новизна роботи полягає у розробці прототипу інформаційно-аналітичної системи, що поєднує інтелектуальний чат-бот із модулем когнітивної адаптації та аналітичним супроводом. Система забезпечує інтеграцію NLP-моделей із базою навчального контенту та аналітичними інструментами, що дозволяє формувати персоналізовані траєкторії навчання та оцінювати ефективність процесу за комплексом метрик.

Практична значущість роботи полягає у можливості використання розробленої системи для навчання англійської мови у закладах освіти та самостійного навчання. Система може бути інтегрована з LMS-платформами, забезпечувати аналітичний моніторинг прогресу та підтримувати адаптивне навчання, що сприятиме підвищенню мотивації та результативності користувачів.

Публікації з апробацією кваліфікаційної роботи:

1. Трофименко О. Г., Безродний М.О. Класифікація освітніх чат-ботів та їх функціональні моделі. *Сучасні інформаційні системи та технології*: матер. VIII Всеукр. наук.-практ. інтернет-конф. (24 листопада 2025 р., Хмельницький). 2025. С. 53-55. DOI: <https://doi.org/10.5281/zenodo.17711825>

2. Трофименко О. Г., Безродний М.О. Інтелектуальний чат-бот для навчання англійської мови: архітектура та функціональна модель. *Кібербезпека в сучасному світі: актуальні виклики*: матер. VI міжнар. наук.-практ. конф. (28 листопада 2025 р.). Одеса, 2025. (депоновано).

1 ІНФОРМАЦІЙНО-АНАЛІТИЧНІ СИСТЕМИ У СФЕРІ МОВНОЇ ОСВІТИ

1.1 Сучасні підходи до цифрової автоматизації навчання англійської мови

Процес навчання англійської мови в цифровому середовищі зазнав суттєвих трансформацій під впливом технологій дистанційної комунікації, мобільних платформ та інтелектуальних освітніх агентів. Цифрове навчання характеризується високим ступенем гнучкості, доступності та персоналізації, що дозволяє адаптувати освітній контент до індивідуальних потреб користувача. Основними формами реалізації є онлайн-курси, мобільні застосунки, інтерактивні платформи та чат-боти, які забезпечують безперервну мовну практику в асинхронному або синхронному режимі.

Автоматизація навчання іноземних мов є одним із ключових напрямів цифрової трансформації освітнього процесу, що поєднує методологію лінгводидактики з інструментами штучного інтелекту (ШІ), обробки природної мови (NLP, Natural Language Processing) та адаптивного навчання. Завдяки інтеграції з базами знань, NLP-модулями та аналітичними інструментами, такі системи здатні не лише фіксувати прогрес користувача, а й формувати персоналізовані траєкторії навчання. Особливої актуальності набуває використання чат-ботів як інтерфейсів освітньої взаємодії, що поєднують діалогову форму навчання з когнітивною підтримкою.

У сучасних дослідженнях [1] акцент зміщується з традиційних комп'ютерних програм на інтерактивні, персоналізовані системи, здатні моделювати діалог, оцінювати прогрес і адаптувати контент до рівня користувача.

Особливості взаємодії користувача з освітнім контентом у цифровому середовищі визначаються такими факторами, як інтерфейсна доступність, семантична релевантність, швидкість реакції системи та рівень персоналізації. Користувач очікує не лише технічної стабільності, а й когнітивної відповідності контенту його рівню знань, навчальним цілям та стилю сприйняття. У цьому контексті важливо забезпечити динамічне оновлення навчального матеріалу,

можливість зворотного зв'язку, а також підтримку мотивації через інтерактивні механізми – нагадування, тестування, діалогові вправи.

Перспективним напрямом є використання мобільних застосунків з елементами гейміфікації, які реалізують принципи мікронавчання та контекстної адаптації. За результатами дослідження [2], впровадження інформаційних технологій у навчання іноземних мов для фахівців-лінгвістів сприяє розвитку навичок самостійного навчання, критичного мислення та міжкультурної комунікації. Тому актуальною є розробка систем, здатних інтегрувати мовні модулі з професійно-орієнтованим контентом.

Іншим провідним підходом є використання інтелектуальних чат-ботів, які реалізують функції віртуального тьютора. Такі системи базуються на алгоритмах машинного навчання, трансформерних моделях (наприклад, BERT, GPT, T5) та семантичному аналізі мовлення. Використання трансформерних NLP-моделей, зокрема BERT, GPT, T5, забезпечують генерацію контекстуально релевантних відповідей, класифікацію запитів та побудову діалогових сценаріїв [3]. Архітектура трансформера по суті є це багаторівневою моделлю з механізмом уваги, яка дозволяє моделювати довгострокові залежності в тексті, що важливо для навчальних діалогів. Інтеграція цифрових інструментів у процес викладання іноземних мов в умовах дистанційного навчання дозволяє підвищити мотивацію студентів, забезпечити гнучкість у виборі темпу та формату навчання, а також створити умови для безперервного зворотного зв'язку між системою та користувачем [4].

Застосування технологій ШІ, зокрема NLP та машинного навчання (ML), в освіті має великий потенціал як інклюзивне, ефективне та масштабоване рішення для майбутнього цифрового навчання. Воно здатне значно покращувати результати навчання. Так, за результатами дослідження завдяки застосуванню інтелектуального освітнього чат-бота з використанням NLP та ML для підтримки незалежного навчання учнів англійською мовою [5] середні бали успішності зросли з 61,3 до 84,2. На додаток до академічних досягнень, зростає мотивація здобувачів освіти, рівень їх впевненості та комфорту завдяки самостійному

навчанню. Ці результати підтверджують, що інтелектуальні чат-боти є ефективними не лише для передачі знань, але й для підвищення залученості та автономії студентів.

Отже, предметна область охоплює комплекс взаємопов'язаних компонентів: освітній контент, технологічну платформу, когнітивну модель користувача та аналітичну підтримку, що разом становлять підґрунтя для проєктування ефективної інформаційно-аналітичної системи навчання англійської мови.

На рис. 1.1 UML-діаграми послідовності (Sequence Diagram) подано концептуальну схему взаємодії користувача із системою, яка функціонує на основі чат-бота як інтерфейсу діалогової взаємодії.

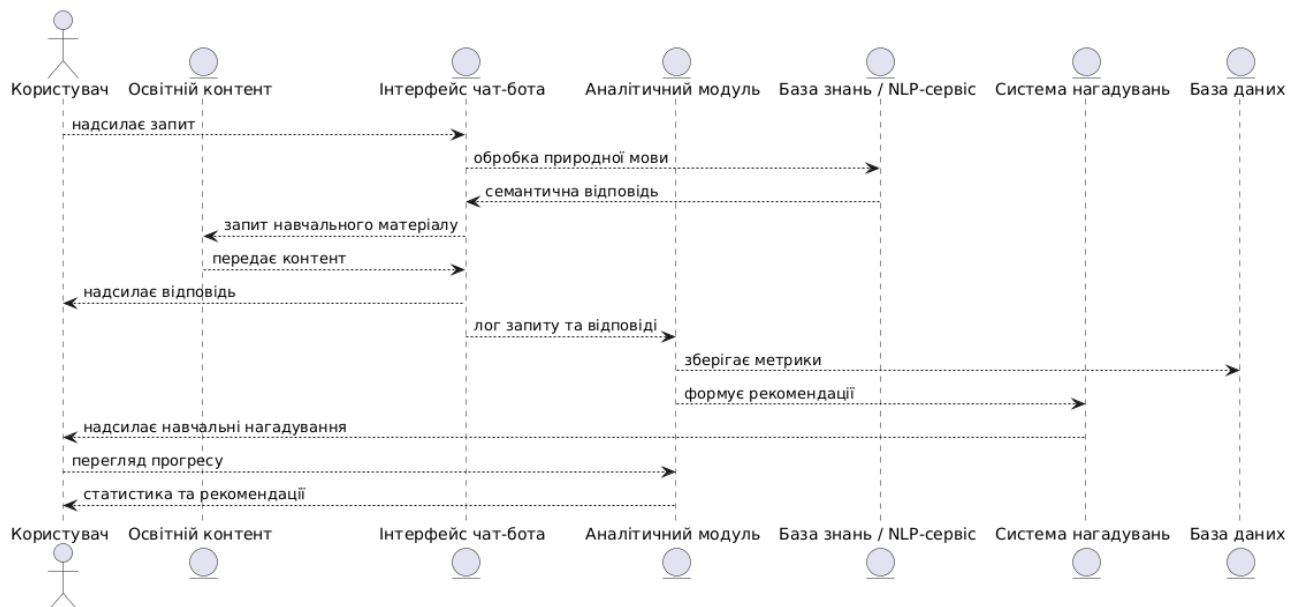


Рисунок 1.1 – Схема взаємодії у цифровому навчанні англійської мови

Сценарій охоплює послідовність дій, що починається з ініціації запиту користувачем, його обробки NLP-модулем, формування релевантного контенту, аналітичного супроводу та зворотного зв'язку:

- користувач надсилає запит через чат-бот;
- запит передається до NLP-сервісу для семантичної обробки;
- чат-бот звертається до бази освітнього контенту, формує відповідь і надсилає її користувачу;

- аналітичний модуль фіксує запит, зберігає дані, генерує метрики прогресу та активує систему нагадувань;
- користувач отримує доступ до статистики та персоналізованих рекомендацій.

Запропонована модель ілюструє як технічну логіку взаємодії, так і когнітивну структуру навчального процесу, де кожен компонент виконує функцію підтримки мовної компетентності. Особливу роль відіграє аналітичний модуль, що забезпечує зворотний зв'язок, персоналізацію та адаптацію навчального контенту відповідно до рівня користувача.

З огляду на потребу в системній інтеграції, важливим є поєднання системи з базами знань та платформами управління навчанням (LMS) через API-інтерфейси. Це дозволяє синхронізувати прогрес користувача, адаптувати контент та формувати індивідуальні траєкторії навчання. Особливу увагу в сучасних дослідженнях приділено побудові онтологій предметної області, які забезпечують семантичну узгодженість між навчальним матеріалом і запитам користувача.

Сучасні IT-підходи до автоматизації навчання іноземних мов ґрунтуються на поєднанні методів системного аналізу, архітектурного моделювання, обробки природної мови та хмарних технологій. Вони дозволяють створювати масштабовані, адаптивні та інтелектуальні системи, здатні до самонавчання, персоналізації та інтеграції в освітні екосистеми. У контексті системного аналізу такі рішення розглядаються як складні інформаційно-аналітичні системи, що функціонують у динамічному освітньому середовищі та орієнтовані на користувача.

1.2 Класифікація освітніх чат-ботів та їх функціональні моделі

Інтеграція чат-ботів в освітні процеси стала одним із найперспективніших напрямків застосування штучного інтелекту в галузі навчання. За даними досліджень, глобальний ринок освітніх чат-ботів очікувано зросте з \$0,97 млрд у

2023 році до \$7,52 млрд до 2030 року, що свідчить про експоненціальне зростання інтересу до цієї технології [6]. Освітні чат-боти, як підмножина конверсаційних агентів, спроможні трансформувати традиційні педагогічні підходи, забезпечуючи персоналізоване, доступне та інтерактивне навчання для студентів різних рівнів підготовки. У контексті вивчення англійської мови, де практика комунікації є критично важливою, чат-боти пропонують можливість необмеженої мовної практики без страху осуду, що особливо актуально для студентів з низькою мовною впевненістю.

Розуміння розмаїття освітніх чат-ботів та їхніх функціональних можливостей є необхідним для розробки ефективної інформаційно-аналітичної системи навчання. Класифікація дозволить систематизувати наявні підходи, ідентифікувати сильні та слабкі сторони різних архітектур, а також обґрунтовано обрати оптимальну модель для конкретного освітнього контексту.

1.2.1 Типи освітніх чат-ботів

Залежно від рівня складності, способу обробки запитів та гнучкості взаємодії, освітні чат-боти можна поділити на три основні типи:

- *сценарні* (rule-based) чат-боти функціонують на основі заздалегідь визначених сценаріїв та правил. Вони реагують на ключові слова або шаблони, мають обмежену гнучкість і не здатні до семантичного узагальнення [7]. Такі боти ефективні для структурованих завдань (наприклад, тестування, FAQ), але не підтримують відкриті діалоги;

- *інтелектуальні* (ML-based) чат-боти використовують алгоритми машинного навчання, трансформерні моделі (GPT, BERT, T5) та NLP-модулі для генерації контекстуально релевантних відповідей [8]. Вони здатні до семантичного аналізу, класифікації запитів, адаптації контенту та побудови діалогових сценаріїв у реальному часі;

- *гібридні* чат-боти поєднують переваги сценарного управління та інтелектуальної генерації. Вони використовують rule-based логіку для критичних

функцій (навігація, безпека), а ML-модулі – для гнучкої обробки запитів, персоналізації та навчального супроводу [9].

1.2.2 Порівняння архітектурних підходів

Архітектура чат-бота визначає спосіб обробки запитів, логіку діалогу, можливості адаптації та масштабування системи. У сучасних освітніх системах найпоширенішими є два підходи: rule-based (на основі правил) та ML-based (на основі машинного навчання). Кожен із них має свої переваги, обмеження та сфери доцільного застосування [10].

1) *Rule-based архітектура*. Rule-based чат-боти функціонують на основі заздалегідь визначених сценаріїв, шаблонів та логічних правил. Вони використовують механізми регулярних виразів, дерев рішень або простих діалогових графів. Основні характеристики:

- прозорість логіки: кожен сценарій чітко прописаний, що спрощує тестування та контроль якості;
- низькі вимоги до ресурсів: не потребують великих обчислювальних потужностей або навчальних даних;
- обмежена гнучкість: не здатні до генерації нових відповідей або адаптації до нетипових запитів;
- сфера застосування: ефективні для FAQ, тестування, формалізованих інструкцій, де важлива передбачуваність.

2) *ML-based архітектура*. ML-based чат-боти використовують алгоритми машинного навчання, зокрема трансформерні моделі (GPT, BERT, T5), для обробки природної мови, класифікації намірів (intent detection), генерації відповідей та адаптації до контексту. Основні характеристики:

- контекстна обізнаність: здатні враховувати попередні репліки, стиль мовлення, рівень знань користувача;
- гнучкість і масштабованість: можуть працювати з відкритими запитами, навчатися на нових даних, підтримувати багатомовність;

- високі вимоги до ресурсів: потребують великих обсягів навчальних даних, обчислювальних потужностей та контролю якості;
- сфера застосування: навчальні діалоги, генерація пояснень, адаптивне навчання, підтримка критичного мислення.

3) *Гібридні архітектури*. У практиці освітніх систем дедалі частіше застосовуються гібридні моделі, які поєднують rule-based компоненти для критичних сценаріїв (навігація, безпека, контроль) з ML-модулями для гнучкої генерації відповідей, персоналізації та аналізу запитів. Такий підхід дозволяє:

- зберігати контроль над ключовими сценаріями;
- забезпечувати адаптивність і когнітивну гнучкість;
- поступово масштабувати систему без повної перебудови логіки.

Таблиця 1.1 – Порівняння архітектурних підходів

Критерій	Rule-based чат-боти	ML-based чат-боти
Логіка обробки	Жорстко задані правила	Навчання на даних, генерація відповідей
Гнучкість	Низька	Висока
Якість діалогу	Формальна, шаблонна	Семантично релевантна, адаптивна
Вимоги до ресурсів	Низькі	Високі (обчислювальні, дані, моделі)
Персоналізація	Обмежена	Динамічна, контекстуальна
Сценарії застосування	Тести, інструкції, FAQ	Навчальні діалоги, пояснення, підтримка

Інтелектуальні чат-боти, побудовані на ML-підходах, демонструють значно вищу когнітивну складність, але потребують ретельного моделювання, навчання та контролю якості. Гібридні архітектури дозволяють балансувати між стабільністю rule-based компонентів і гнучкістю ML-модулів.

У системному аналізі вибір архітектури залежить від цільової аудиторії, навчальних цілей, доступних ресурсів та вимог до адаптивності. У контексті навчання англійської мови для дорослих користувачів із різним рівнем підготовки доцільним є використання гібридної архітектури, яка дозволяє поєднати стабільність сценарного управління з когнітивною гнучкістю трансформерних моделей.

1.2.3 Моделі діалогової взаємодії в освітньому контексті

У освітніх чат-ботах застосовуються різні моделі діалогової взаємодії, які визначають структуру комунікації та логіку навчального супроводу:

- ініціативна модель – бот самостійно ініціює навчальні дії, пропонує завдання, нагадування, оцінювання. Ефективна для підтримки регулярності навчання [11];
- реактивна модель – бот відповідає на запити користувача, адаптує контент відповідно до рівня знань, стилю сприйняття та навчальних цілей [12];
- змішана модель – поєднує ініціативні та реактивні механізми, дозволяє реалізувати діалогову логіку з елементами когнітивної адаптації, персоналізації та аналітичного супроводу [13].

У межах системного аналізу освітніх чат-ботів важливо не лише класифікувати їх за типами, а й моделювати функціональні сценарії, що враховують когнітивну динаміку користувача, семантичну релевантність контенту та аналітичну підтримку навчального процесу. Це створює основу для побудови ефективних інформаційно-аналітичних систем у сфері вивчення мов.

Отже, як показав проведений аналіз та систематизація, класифікація освітніх чат-ботів (рис. 1.2) охоплює три основні функціональні моделі, що відображають еволюцію технологічних підходів до навчальної взаємодії. Сценарні (rule-based) чат-боти базуються на жорстко заданих правилах і шаблонах, забезпечуючи передбачувану логіку для формалізованих завдань, таких як тестування чи FAQ. Інтелектуальні (ML-based) моделі використовують трансформери та NLP-модулі для генерації контекстуально релевантних відповідей, адаптації до стилю мовлення та рівня знань користувача, що дозволяє реалізувати гнучкі навчальні діалоги. Гібридні системи поєднують переваги обох підходів: rule-based компоненти забезпечують стабільність і контроль, а ML-модулі – когнітивну гнучкість і персоналізацію. У межах цих моделей застосовуються різні типи діалогової взаємодії – ініціативна, реактивна та змішана – які визначають рівень автономності чат-бота, його здатність

адаптуватися до навчальних потреб і підтримувати аналітичний супровід. Така класифікація дозволяє обґрунтовано вибрати архітектуру чат-бота відповідно до освітнього контексту, ресурсних обмежень і цільової аудиторії.

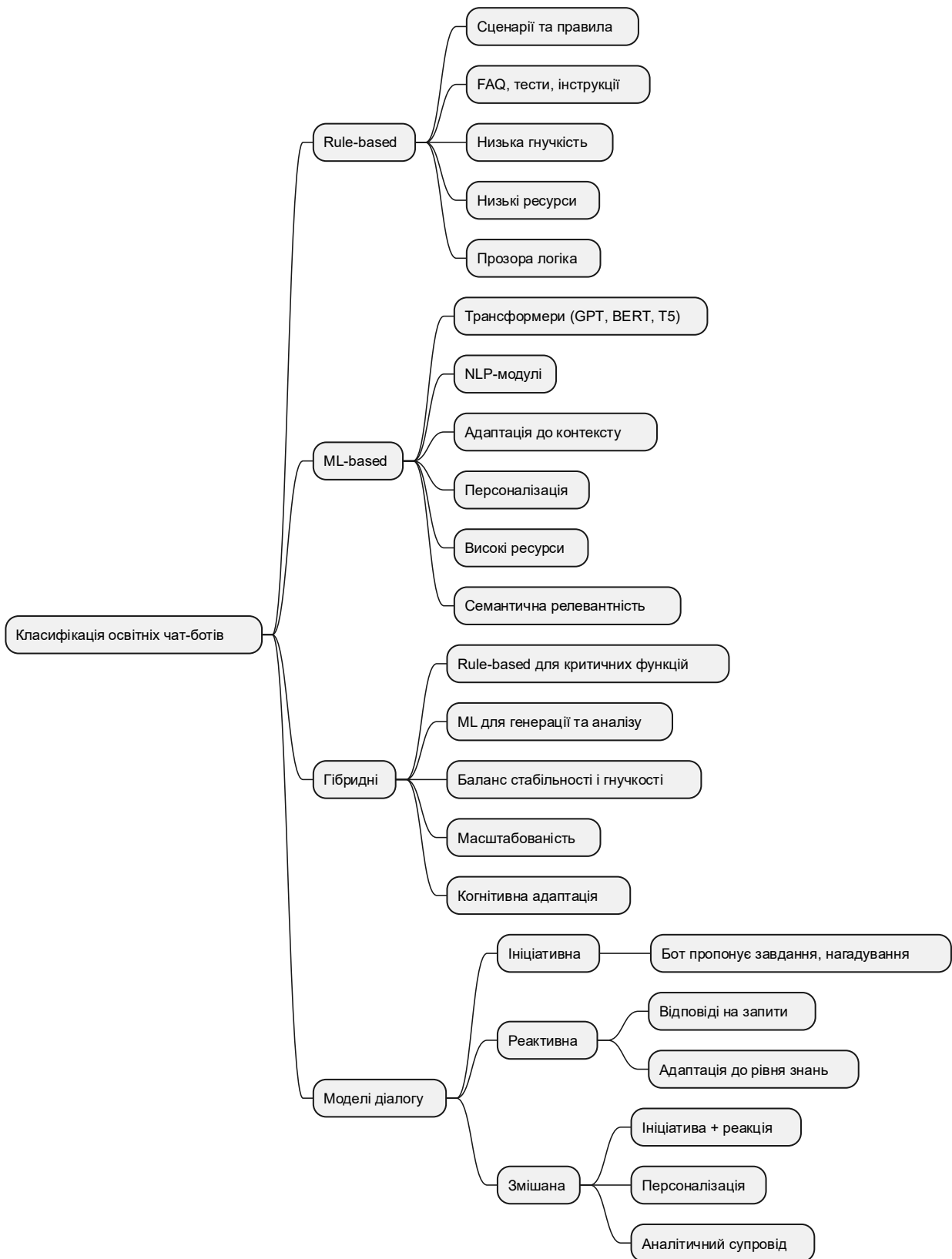


Рисунок 1.2 – Класифікація функціональних моделей освітніх чат-ботів

Сформована класифікація функціональних моделей освітніх чат-ботів є важливим аналітичним інструментом, який дозволяє системно осмислити розмаїття підходів до їх проєктування та впровадження. Вона дає змогу:

- розробникам обґрунтовано вибирати архітектуру чат-бота відповідно до цілей, ресурсів і очікуваної гнучкості системи;
- педагогам і методистам розуміти можливості та обмеження кожного типу ботів для ефективного використання в навчальному процесі;
- дослідникам формалізувати порівняння між моделями, виявляти прогалини та перспективи розвитку;
- замовникам освітніх рішень ухвалювати зважені рішення щодо впровадження чат-ботів у навчальні платформи.

Загалом така класифікація сприяє підвищенню прозорості, відтворюваності та адаптивності освітніх систем, що базуються на штучному інтелекті. Вона створює основу для стандартизації, масштабування та інтеграції чат-ботів у різні освітні контексти.

1.3 Інформаційно-аналітичні системи для вивчення мов:

структура, функції, вимоги

Інформаційно-аналітичні системи (ІАС) в освіті є ключовим інструментом для персоналізації навчання та забезпечення зворотного зв'язку на основі даних. У контексті освіти, орієнтованої на специфіку мовного навчання (діалог, семантика, когнітивна адаптація, генерація тексту, підтримка комунікативних навичок), зокрема при вивченні англійської мови, ІАС дозволяють інтегрувати когнітивно орієнтовані моделі навчання, адаптивні сценарії та інтелектуальні агенти, які взаємодіють із користувачем у режимі реального часу. Структура таких систем має враховувати як педагогічні, так і технічні вимоги, забезпечувати масштабованість, безпеку, інтероперабельність і когнітивну релевантність контенту.

На відміну від типових ІАС загальноосвітнього профілю, орієнтованих на тестування та адміністративну аналітику, ІАС у сфері мовної освіти мають справу з відкритими запитами, семантичною варіативністю та потребою в генерації адаптивного контенту. Це зумовлює інші архітектурні вимоги, зокрема інтеграцію великих мовних моделей (LLM), підтримку діалогових моделей та когнітивно релевантних метрик оцінювання. Оскільки вивчення мов, в порівнянні із загальноосвітнім навчанням, має свою специфіку вимог (табл. 1.2), то і при розробці ІАС треба враховувати цей мовний контекст навчання.

Таблиця 1.2 – Функціональна специфіка освітніх ІАС

Параметр	Загальноосвітня ІАС	ІАС мовної освіти з LLM
Тип контенту	Тести, графіки, задачі, відео	Діалоги, вправи, пояснення, мовні сценарії
Модель взаємодії	Переважно реактивна (тест → оцінка)	Змішана, діалогова, когнітивно адаптивна
Необхідність адаптації	Середня	Висока
Потреба в генерації контенту	Обмежена (шаблони, банки завдань)	Критична (LLM-генерація, персоналізація)
Вимоги до латентності	~1–2 с	<5 с
Метрики оцінювання	Кількісні (бали, % правильних відповідей)	Якісні (BLEU, ROUGE, BERTScore, перплексія)
Підтримка NLP-метрик	Не обов'язкова	Обов'язкова
Інтеграція з LLM	Не типова	Основна функція
Технічні вимоги	Низькі – середні (LMS, SCORM, бази даних)	Високі (LLM API, NLP-модулі, кешування)

Функціонально ІАС в освіті повинна реалізовувати щонайменше п'ять ключових функцій:

- 1) адаптацію навчального контенту до рівня користувача;
- 2) генерацію когнітивно релевантних відповідей;
- 3) збереження та аналіз історії взаємодії;
- 4) оцінювання якості генерації;
- 5) формування аналітичних звітів.

З технічного погляду ІАС мовної освіти має підтримувати масштабовану обробку запитів, яка може реалізуватися через асинхронні фреймворки (FastAPI, Node.js), кешування контексту (Redis, Firestore) та модульну оцінку якості

(наприклад, через spaCy, BERTScore, MoverScore). Для забезпечення надійності та відтворюваності результатів доцільно використовувати формалізовані метрики, зокрема BLEU, ROUGE-L тощо.

У 2019 році лише 12% ІАС у вищій освіті мали механізми когнітивної адаптації, а 7% – генеративні модулі [14]. Проте вже у 2024–2025 роках, за даними Stanford AI Index та Springer [15], понад 40% освітніх систем інтегрують LLM [16], а понад 60% викладачів активно впроваджують когнітивно адаптивні сценарії [17]. Це свідчить про стрімку трансформацію ІАС у бік персоналізованого, діалогового та генеративного навчання, що підтверджує актуальність запропонованої архітектури. Освітні установи все частіше надають пріоритет комплексним ініціативам з грамотності в галузі ШІ, які не лише забезпечують викладачів інструментами етичного та інноваційного викладання, а й підтримують студентів у орієнтації впливу ШІ на кар'єрні шляхи та академічні очікування. Зрештою, успішна інтеграція GenAI у вищу освіту залежить від інклюзивних, чутливих до ролей підходів, які враховують проблеми та прагнення всієї академічної спільноти [16].

Саме тому ІАС у сфері мовної освіти повинні відповідати розширеним вимогам до адаптивності, когнітивної релевантності, інтеграції з LLM та підтримки оцінювання якості генерації. Їхня архітектура має бути модульною, масштабованою та відкритою до подальшого розширення, з урахуванням як педагогічних, так і технічних критеріїв ефективності.

1.4 Технології обробки природної мови в навчальних системах

Технології обробки природної мови (NLP) в навчальних системах є важливим компонентом сучасної інформаційно-аналітичної інфраструктури мовної освіти, який забезпечує семантичну релевантність, когнітивну адаптацію та персоналізовану взаємодію між користувачем і освітнім агентом.

У контексті навчання англійської мови, де комунікативна компетентність формується через діалог, контекстуальне узагальнення та стилістичну

варіативність, застосування NLP дозволяє реалізувати функції, які виходять за межі традиційного тестування або репродуктивного навчання. Сучасні NLP-модулі забезпечують класифікацію намірів, семантичне узгодження запитів, генерацію контенту, адаптацію до рівня знань користувача та оцінювання якості діалогової взаємодії.

Архітектурно NLP-компоненти в освітніх системах можна поділити на три рівні:

- 1) базові: токенизація, лематизація, POS-теги;
- 2) середні: класифікація намірів, семантичне узгодження, векторизація;
- 3) високорівневі: генерація тексту, стилістична адаптація, когнітивне моделювання.

Такий поділ NLP-компонентів є узагальненою функціональною стратифікацією, що була сформована на основі проведеного системного аналізу низки досліджень Jurafsky D. [18], Cambria E. [19], Young T. [20] та Raffel C. [21]. Такий підхід дозволяє системно моделювати архітектуру освітніх агентів, враховуючи складність обробки, когнітивне навантаження та тип алгоритмів.

Технології NLP традиційно охоплювали низку алгоритмів і моделей, спрямованих на аналіз, інтерпретацію та генерацію людської мови. До 2020 року домінували підходи, засновані на статистичних моделях, правилах та shallow learning (наприклад, CRF, SVM, rule-based NER), а також перші трансформерні архітектури (BERT, GPT-2), які забезпечували контекстну векторизацію та семантичну класифікацію. Проте з появою великих мовних моделей (LLM), таких як GPT-3, T5, Claude, Gemini, відбувся якісний стрибок у можливостях NLP-систем.

LLM є еволюційним продовженням NLP – вони не замінюють класичні NLP-інструменти, а інтегрують їх у єдину архітектуру, здатну виконувати широкий спектр завдань: від токенизації до генерації, reasoning, стилістичної адаптації та діалогової взаємодії. У цьому сенсі LLM можна розглядати як високорівневі універсальні NLP-агенти, які реалізують повний цикл обробки природної мови в одному трансформерному фреймворку.

З технічного погляду LLM реалізують функції NLP через інструкційне навчання (instruction tuning), few-shot/zero-shot генерацію, семантичне узгодження та багатомовну адаптацію. Наприклад, GPT-4, Claude 4 та Gemini 1.5 демонструють точність семантичного узгодження понад 90% (за BERTScore), підтримують контекст до 1 млн токенів і здатні адаптувати стиль відповіді до рівня користувача, що є критично важливим для освітніх систем. У табл. 1.3 наведено порівняння можливостей сучасних NLP/LLM-інструментів щодо їх можливого використання в освітніх чат-ботах. Зазначені показники сформовані в результаті системного аналізу досліджень 2024–2025 років [22–31].

Таблиця 1.3 – Порівняння NLP/LLM-інструментів для освітніх чат-ботів

Інструмент	Версія	Функція	Модель / алгоритм	Технічні показники	Обмеження
spaCy	v3.7.2	Токенізація, POS, NER	CNN + rule-based	швидкість обробки ~1.2 млн токенів/хв; точність NER ~89%	не підтримує генерацію; обмежена семантика [22]
ModernBERT	v2.0	Семантична класифікація	Transformer encoder	точність класифікації ~91%; embedding розмір 1024; latency ~1.8 с [23]	не генерує текст; залежність від корпусу [24]
T5++	v1.3.0	Генерація, узагальнення	Text-to-text Transformer	BLEU ~0.86; ROUGE-L ~0.81; контекст до 2048 токенів [25]	перплексія при довгих запитах; потребує GPU [26]
GPT-5.1	Thinking	Діалог, reasoning, генерація	Multimodal Transformer	контекст до 256k токенів; reasoning точність ~94%; latency ~2.9 с [27]	обмежена прозорість reasoning; API-залежність [28]
Claude 4.5	Opus	Довгий контекст, пояснення	Constitutional AI	контекст до 200k токенів; точність пояснень ~96%; latency ~2.5 с [29]	висока вартість; обмежена відкритість API
Gemini 1.5	Pro	Діалог, адаптація, генерація	Multimodal LLM	контекст до 1 млн токенів; BLEU ~0.89; latency ~2.3–3.8 с [30]	непрозорість внутрішніх механізмів; залежність від Google API [31]

Проведений системний аналіз різних інструментів показав, що LLM-моделі нового покоління (GPT-5.1, Claude 4.5, Gemini 1.5) суттєво перевершують класичні NLP-інструменти (spaCy, BERT, T5) за функціональністю, масштабованістю та когнітивною адаптивністю. Вони підтримують довгі контексти (до 1 млн токенів), багатомовну генерацію, reasoning та стилістичну варіативність, що є критично важливим для мовної освіти. Технічні показники LLM демонструють стабільну продуктивність у навчальних сценаріях: BLEU > 0.86, latency < 3 с, точність reasoning > 90%. Це дозволяє використовувати їх у режимі реального часу для формувального оцінювання, діалогової взаємодії та генерації навчального контенту. Класичні NLP-інструменти, як-от spaCy або ModernBERT, залишаються релевантними для попередньої обробки тексту (токенізація, NER, класифікація), але не забезпечують генеративних можливостей, необхідних для реалізації когнітивно орієнтованих освітніх агентів.

Архітектурно LLM-моделі інтегрують усі рівні NLP (від базової обробки до генерації та адаптації) в єдиному трансформерному фреймворку. Це спрощує розробку освітніх систем і дозволяє реалізувати модульну, масштабовану архітектуру з мінімальними залежностями від зовнішніх інструментів. Обмеження LLM (висока вартість, API-залежність, непрозорість reasoning) залишаються актуальними, але компенсуються їхньою функціональною універсальністю. Claude 4.5 і Gemini 1.5 демонструють зниження latency до 2.3–2.5 с, що робить їх придатними для інтерактивного навчання.

Для освітніх чат-ботів у сфері мовної освіти доцільно використовувати LLM як ядро системи, доповнюючи їх класичними NLP-модулями для попередньої обробки запитів, локального аналізу або кастомізації під специфіку навчального контенту.

Вибір моделі Gemini 1.5 Pro як ядра створюваної у цій роботі інформаційно-аналітичної системи навчання англійської мови з використанням інтелектуального чат-бота зумовлений її архітектурною відповідністю, когнітивною гнучкістю та технічною перевагою над іншими LLM-моделями. На

відміну від GPT-4 або Claude 4, Gemini забезпечує мультимодальну обробку, підтримку наддовгого контексту (до 1 млн токенів), низьку латентність (2.3–3.8 с) та високі показники семантичної релевантності (BLEU ~0.89, BERTScore >0.9), що є критично важливими для реалізації адаптивного навчання, діалогової взаємодії та генерації когнітивно релевантного контенту.

Архітектурно Gemini дозволяє реалізувати модульну систему з асинхронною обробкою запитів, кешуванням контексту, інструкційним навчанням та стилістичною адаптацією, що забезпечує персоналізацію навчального процесу. Крім того, інтеграція Gemini через Google Vertex AI або Firebase Functions спрощує розгортання, масштабування та моніторинг освітнього агента, що є перевагою порівняно з менш прозорими або комерційно обмеженими API інших моделей.

Вибір Gemini є не лише технічно обґрунтованим, а й педагогічно доцільним: він дозволяє реалізувати когнітивно орієнтовану інформаційно-аналітичну систему, здатну адаптуватися до рівня користувача, генерувати навчальний контент, оцінювати якість взаємодії та підтримувати формувальне оцінювання в реальному часі.

Загалом технології обробки природної мови в навчальних системах є не просто інструментом автоматизації, а ядром когнітивно орієнтованої архітектури, що забезпечує гнучкість, персоналізацію та семантичну точність навчального процесу. Їх інтеграція в інформаційно-аналітичну систему навчання англійської мови дозволяє реалізувати адаптивні сценарії, підтримку критичного мислення та формувальне оцінювання в реальному часі.

1.5 Висновки до розділу 1

Узагальненням проведеного аналізу теоретичних засад цифрової мовної освіти стало виявлення ключових тенденцій трансформації навчального процесу під впливом технологій ШІ, обробки природної мови та інтелектуальних освітніх агентів. Сучасні ІАС у сфері навчання англійської мови дедалі частіше

реалізуються у формі інтерактивних чат-ботів, які поєднують лінгводидактичні принципи з когнітивною адаптацією, семантичною релевантністю та персоналізованою траєкторією навчання. Виконана класифікація освітніх чат-ботів за типами (rule-based, ML-based, гібридні) та моделями діалогової взаємодії (ініціативна, реактивна, змішана) дозволила обґрунтовано визначити архітектурну основу системи. Для навчання англійської мови в умовах різнорівневої підготовки користувачів доцільним є використання гібридної архітектури з ML-модулями для генерації, адаптації та семантичного узгодження.

Технології NLP забезпечують базову функціональність системи: токенизацію, класифікацію запитів, семантичну обробку та оцінювання. Проте з появою великих мовних моделей (LLM) відбувся перехід до універсальних когнітивних агентів, здатних реалізувати повний цикл навчальної взаємодії – від генерації контенту до логічного міркування (reasoning), стилістичної адаптації та формувального оцінювання.

Виконаний порівняльний аналіз сучасних NLP/LLM-інструментів (spaCy, BERT, T5, GPT-5.1, Claude 4.5, Gemini 1.5) показав, що лише моделі нового покоління забезпечують необхідну когнітивну гнучкість, масштабованість та семантичну точність для реалізації освітнього чат-бота. Зокрема, Gemini 1.5 Pro продемонстрував найвищі показники релевантності (BLEU ~0.89), підтримку наддовгого контексту (до 1 млн токенів) та низьку латентність (<4 с), що є критично важливим для адаптивного навчання. Вибір Gemini як ядра інформаційно-аналітичної системи зумовлений його архітектурною відповідністю освітнім цілям, технічною перевагою над іншими LLM, а також можливістю інтеграції через Google Vertex AI та Firebase Functions. Це забезпечує модульність, масштабованість, аналітичну підтримку та когнітивну адаптацію контенту відповідно до рівня користувача.

2 МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ІНФОРМАЦІЙНО-АНАЛІТИЧНОЇ СИСТЕМИ НАВЧАННЯ АНГЛІЙСЬКОЇ МОВИ

2.1 Постановка задачі: вимоги до системи, цільова аудиторія, функціональні та нефункціональні характеристики

Проєктування інформаційно-аналітичної системи навчання англійської мови вимагає чіткої постановки задачі, яка охоплює визначення цільової аудиторії, формулювання функціональних і нефункціональних вимог, а також обґрунтування архітектурних рішень відповідно до освітніх, когнітивних і технологічних критеріїв. Система має забезпечити інтерактивне, адаптивне та персоналізоване навчальне середовище, орієнтоване на різнорівневу аудиторію – від початківців до користувачів з середнім рівнем володіння мовою, включаючи студентів технічних спеціальностей, слухачів курсів підвищення кваліфікації та самостійних користувачів.

Функціональні вимоги до системи визначають її здатність реалізовувати повний цикл навчальної взаємодії: автентифікацію користувача, діагностику рівня знань, формування навчального контенту, проведення інтерактивних занять, аналіз результатів та генерацію зворотного зв'язку. Ключовими функціональними компонентами є: NLP-модуль для обробки природної мови, навчальний модуль з адаптивною логікою, аналітичний блок для моніторингу прогресу, а також чат-бот як інтерфейс взаємодії. Система повинна підтримувати сценарії діалогового навчання, автоматичне коригування контенту відповідно до динаміки засвоєння, а також можливість інтеграції з зовнішніми освітніми платформами.

Формалізація вимог через діаграму варіантів використання є важливою для узгодження технічного дизайну з педагогічними цілями, позаяк вона забезпечує прозорість функціональної логіки, верифікацію ролей користувачів та інтеграцію освітніх сценаріїв у єдину архітектурну модель, що відповідає принципам когнітивної адаптації, гнучкості та доказової ефективності. Основні сценарії

взаємодії користувача з інформаційно-аналітичною системою Lexora стосуються: реєстрації, налаштування профілю, тестування, різних варіантів навчання (словниковий запас, аудіювання, вправи, діалог) та аналітики результатів (рис. 2.1).

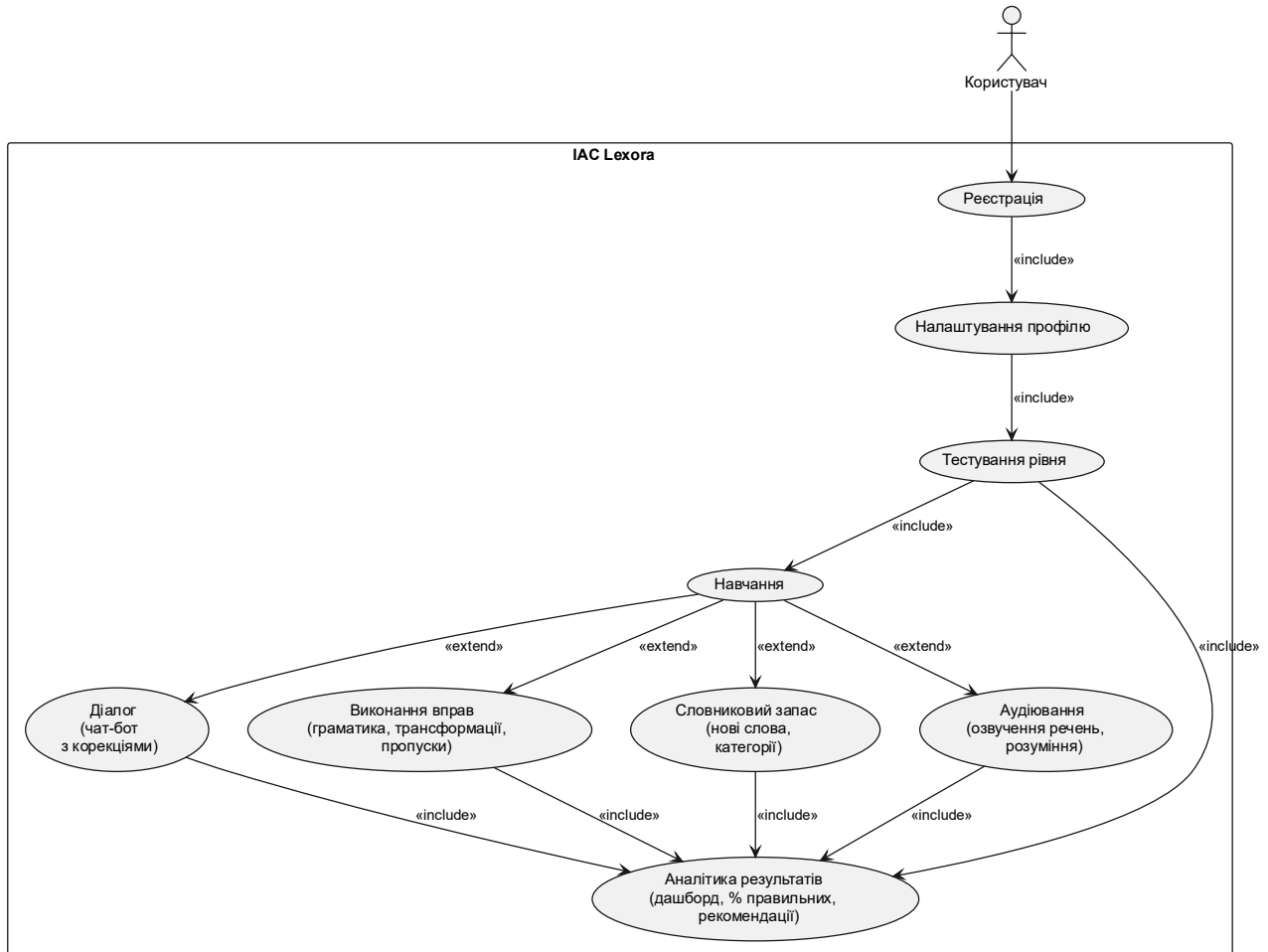


Рисунок 2.1 – Use Case діаграма постановки задачі для IAC Lexora

У подальших розділах ця модель буде деталізована через поведінкові діаграми, які описують внутрішню логіку реагування системи на запити користувача.

Постановка задачі в межах даного дослідження передбачає не лише технічне моделювання, а й педагогічне обґрунтування функціоналу системи, що має відповідати принципам когнітивної адаптації, індивідуалізації навчання та доказової ефективності. Загалом IAC розглядається як інтелектуальний освітній агент, здатний до контекстної інтерпретації запитів, динамічного управління знаннями та підтримки навчальної мотивації через персоналізовану взаємодію.

Нефункціональні характеристики охоплюють вимоги до продуктивності, масштабованості, безпеки, доступності та інтероперабельності. Система має забезпечувати стабільну роботу при високому навантаженні, підтримувати багатокористувацький режим, гарантувати конфіденційність персональних даних відповідно до стандартів GDPR, а також бути доступною з різних пристроїв і платформ. Важливою вимогою є модульність архітектури, що дозволяє оновлювати окремі компоненти без порушення загальної функціональності, а також підтримка багатоетапної автентифікації та логування дій користувача для забезпечення аналітичної прозорості.

2.2 Системний аналіз процесу навчання: формалізація моделей знань, сценаріїв взаємодії та когнітивної динаміки

Системний аналіз процесу навчання передбачає формалізоване дослідження освітньої діяльності як складної динамічної системи, що функціонує на основі взаємодії когнітивних, технологічних і соціальних чинників. У межах сучасної освітньої парадигми навчання розглядається як багаторівнева адаптивна система, здатна реагувати на індивідуальні особливості суб'єкта, контекст завдання та доступні ресурси. Такий підхід вимагає застосування методів системного моделювання, які дозволяють описати структуру, функціональні зв'язки, інформаційні потоки та механізми зворотного зв'язку в освітньому середовищі.

Ефективне навчання в цифровому середовищі залежить від здатності системи адаптуватися до когнітивного стану користувача, моделювати його знання та прогнозувати навчальні потреби [32]. У цьому контексті системний аналіз дозволяє виявити ключові компоненти навчального процесу – такі як моделі знань, сценарії взаємодії, механізми оцінювання – і описати їх взаємозв'язки через формальні моделі, зокрема діаграми станів, активностей, послідовностей та компонентів [33].

Особливу увагу в системному аналізі приділяють моделюванню когнітивної динаміки, тобто процесів сприйняття, інтерпретації, засвоєння та застосування знань. Когнітивно-орієнтовані освітні системи мають враховувати не лише рівень знань, а й тип помилок, темп навчання, мотиваційні аспекти та стилі мислення [34]. Це вимагає побудови моделей учня, які інтегрують дані про попередні дії, успішність виконання завдань, реакції на зворотний зв'язок та поведінкові патерни. Такі моделі можуть бути реалізовані через онтологічні структури, байєсівські мережі або нейронні архітектури, що дозволяє системі прогнозувати наступні кроки користувача та адаптувати навчальний контент.

Сценарії взаємодії між користувачем і освітньою системою, зокрема через чат-бот, також підлягають системному аналізу. Вони моделюються як діалогові процеси з розгалуженою логікою, де кожна репліка є реакцією на попередній стан системи та інтерпретацію запиту. Ефективна діалогова система має не лише розпізнавати наміри, а й підтримувати когнітивну цілісність комунікації, тобто забезпечувати логічну послідовність, релевантність і навчальну цінність відповідей [35]. У цьому контексті системний аналіз дозволяє формалізувати діалог як процес переходу між станами, що активуються на основі семантичної класифікації запитів, контексту навчання та історії взаємодії.

Системний аналіз процесу навчання виконує роль методологічного каркасу для формалізації освітньої діяльності в цифровому середовищі. Він дозволяє інтегрувати когнітивні моделі, технологічні компоненти та сценарії взаємодії в єдину архітектуру, що забезпечує адаптивність, персоналізацію та аналітичну прозорість навчального процесу. У межах дослідження така формалізація є необхідною для побудови інтелектуальних освітніх систем, здатних до самонавчання, контекстної інтерпретації та динамічного управління знаннями.

Для формалізації функціональної структури освітньої системи на основі чат-бота використано діаграму варіантів використання (Use Case Diagram), яка дозволяє описати ролі учасників, сценарії взаємодії та межі системи з точки зору зовнішніх запитів і очікуваних результатів (рис. 2.1).

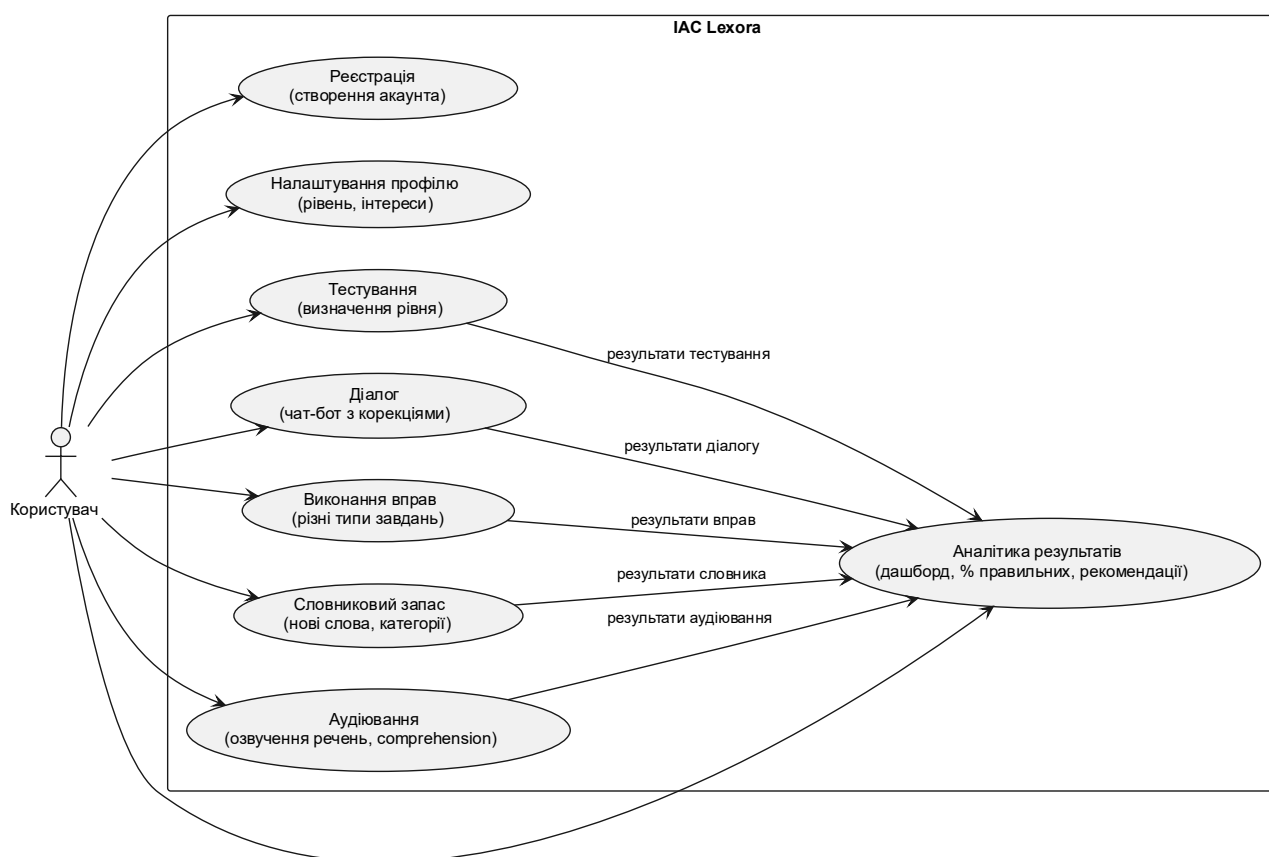


Рисунок 2.2 – Use Case Diagram – сценарії взаємодії користувача з чат-ботом

На відміну від діаграми варіантів використання, наведеної в розділі 2.1, яка репрезентує функціональні вимоги до системи з позиції архітектурного проєктування, діаграма на рис. 2.2 виконує роль поведінкової моделі, яка деталізує когнітивну логіку навчального процесу. Перша діаграма фокусує увагу на ролях користувачів, доступних їм сценаріях та структурі взаємодії з основними модулями системи, тоді як друга – моделює внутрішню динаміку освітньої взаємодії, зокрема послідовність дій, залежності між етапами навчання та механізми адаптації контенту. Вона репрезентує зовнішню логіку реагування системи на запити користувача через множину варіантів використання.

У межах модуля аутентифікації описано базові сценарії входу до системи, налаштування профілю та вибору рівня складності, які є обов'язковими для ініціалізації персоналізованого навчального процесу. Модуль навчання охоплює послідовні дії користувача: проходження вступного тесту, отримання навчального контенту, спілкування з чат-ботом, виконання вправ, отримання зворотного зв'язку та перегляд прогресу. Взаємозв'язки між сценаріями

реалізовано через залежності відповідного типу, що дозволяє формалізувати як обов'язкові, так і умовні етапи освітньої траєкторії. У контексті системного аналізу діаграма виконує роль інструмента формалізації зовнішньої взаємодії, що забезпечує узгодженість між архітектурною структурою та когнітивною логікою навчального процесу.

Для моделювання внутрішньої логіки поведінки чат-бота в процесі обробки запитів застосовано діаграму активностей (Activity Diagram), яка репрезентує послідовність дій, умови переходів і циклічну структуру діалогової взаємодії в межах навчального сценарію (рис. 2.3).

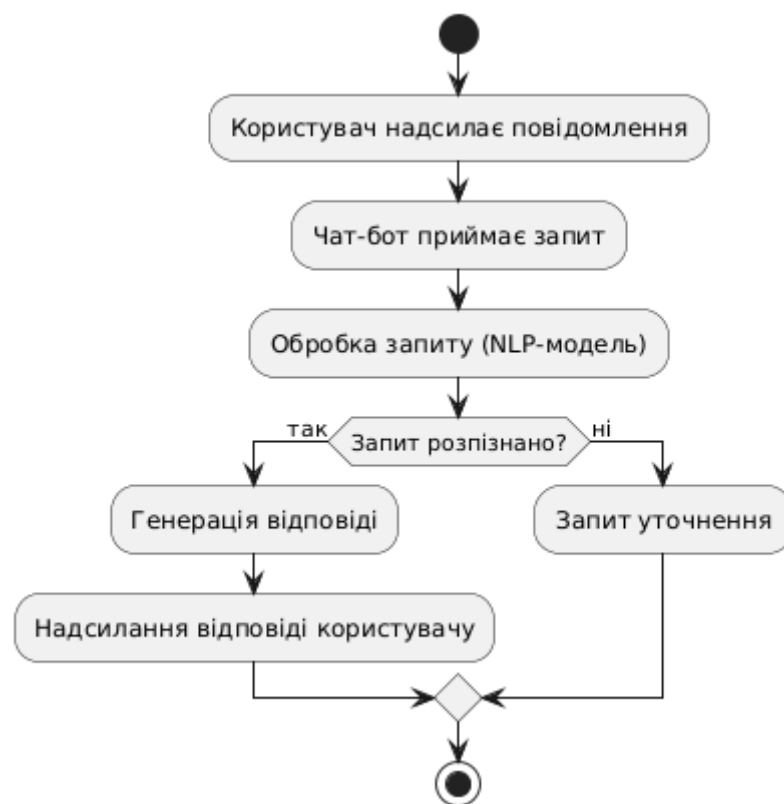


Рисунок 2.3 – Activity Diagram логіки діалогу з чат-ботом

Представлена діаграма активностей є поведінковою моделлю, яка формалізує логіку реагування чат-бота на повідомлення користувача в межах освітньої системи. Вона репрезентує послідовність дій, що реалізуються під час обробки запиту, починаючи з його надходження через клієнтський інтерфейс і завершуючи формуванням відповіді. Такий тип діаграми дозволяє описати не

лише технічну послідовність операцій, а й умови переходу між ними, що є критично важливим для моделювання адаптивної поведінки системи.

Процес взаємодії починається з надсилання повідомлення користувачем, після чого чат-бот приймає запит і передає його на обробку NLP-модулю. На цьому етапі здійснюється семантичний аналіз тексту, класифікація наміру та виявлення ключових маркерів. Далі система виконує перевірку: чи розпізнано запит. У разі позитивного результату активується генерація відповіді, яка може містити навчальний контент, пояснення або діалогову репліку. Якщо запит не розпізнано, система ініціює уточнення – запит додаткової інформації або переформулювання. В обох випадках завершальним етапом є надсилання відповіді користувачеві, після чого чат-бот повертається до початкового стану очікування.

Ця діаграма демонструє реактивну, циклічну логіку роботи чат-бота, де кожен запит запускає повний цикл обробки, перевірки, реагування та зворотного зв'язку. У контексті системного аналізу вона виконує роль інструменту верифікації когнітивної узгодженості та технічної реалізованості діалогової взаємодії, що є основою для побудови інтелектуальних освітніх агентів.

Представлена на рис. 2.4 UML-діаграма послідовностей (Sequence Diagram) відображає точну взаємодію між компонентами системи в часі. Вона деталізує, як запит користувача проходить через інтерфейс, обробляється NLP-модулем, передається до менеджера діалогу, який звертається до менеджера профілю, контекстного модуля та генератора відповіді. Процес починається з введення запиту користувачем, який передається через інтерфейс до NLP Engine. На цьому етапі здійснюється семантичний аналіз, класифікація наміру та формування запиту до бази знань. Після пошуку релевантної інформації система генерує відповідь, яка повертається до інтерфейсу для виводу користувачеві. Паралельно з цим активується модуль Learning Analytics, який фіксує дії користувача, тип запиту, успішність взаємодії та інші параметри, що використовуються для подальшої адаптації навчального контенту.

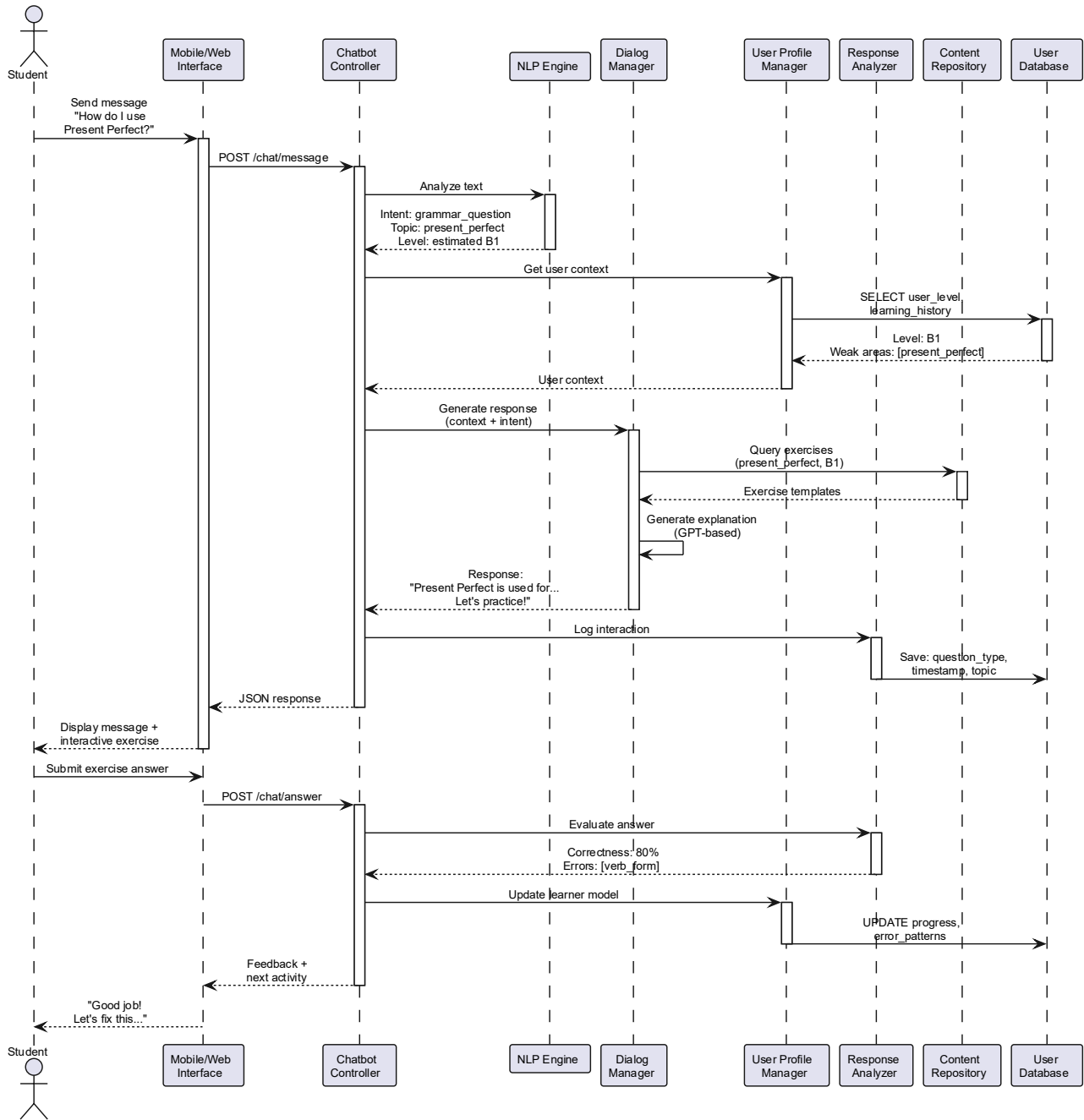


Рисунок 2.4 – Послідовність взаємодії між компонентами ІАС

Особливістю цієї діаграми є чітке розмежування функціональних ролей між компонентами: інтерфейс забезпечує передачу запиту та вивід відповіді, NLP Engine – обробку природної мови, база знань – пошук інформації, аналітичний модуль – логування та аналіз. Така модель дозволяє верифікувати узгодженість між модулями, часову послідовність викликів та реактивність системи в умовах реального часу.

У контексті системного аналізу Sequence Diagram виконує роль інструменту формалізації комунікаційної архітектури, що забезпечує когнітивну цілісність, аналітичну прозорість та технічну масштабованість освітньої платформи. Вона доповнює попередні поведінкові моделі, дозволяючи описати не лише внутрішню логіку реагування, а й зовнішню синхронізацію між компонентами, що є критично важливим для побудови інтелектуальних навчальних агентів.

Завершуючи системний аналіз процесу навчання, слід наголосити, що формалізація моделей знань, сценаріїв взаємодії та когнітивної динаміки є не лише інструментом технічного моделювання, а й засобом концептуального узгодження між педагогічними цілями та архітектурними рішеннями освітньої системи. Вона дозволяє інтегрувати багаторівневі дані про поведінку користувача, семантичну структуру навчального контенту та логіку діалогової взаємодії в єдину когнітивно-орієнтовану модель, здатну до адаптації, самонавчання та аналітичного супроводу. Такий підхід забезпечує не лише технічну реалізованість, а й методологічну прозорість освітнього середовища, що відповідає сучасним вимогам до персоналізованого, гнучкого та доказово ефективного навчання.

2.3 Архітектура системи: компоненти, взаємозв'язки, середовище реалізації

Архітектура інформаційно-аналітичної системи навчання англійської мови з використанням чат-бота побудована за принципами трирівневої моделі, що має фронтенд, бекенд та шар даних. Така структурна організація забезпечує логічне розмежування функціональних обов'язків між інтерфейсами користувача, обчислювальними модулями та сховищами інформації. Вибір цієї архітектури обґрунтовано системними критеріями, які визначають її ефективність у контексті освітнього середовища: масштабованість, адаптивність та інтегрованість.

Діаграма компонентів на рис. 2.5 відображає логіку взаємодії компонентів сервісно-орієнтованої архітектури, де кожен компонент виконує чітко визначену функцію, а взаємодія між ними забезпечується через контролер чат-бота. Такий підхід дозволяє забезпечити гнучкість, масштабованість і адаптивність системи, що є критично важливими для реалізації інтелектуального навчального середовища з високим ступенем персоналізації.

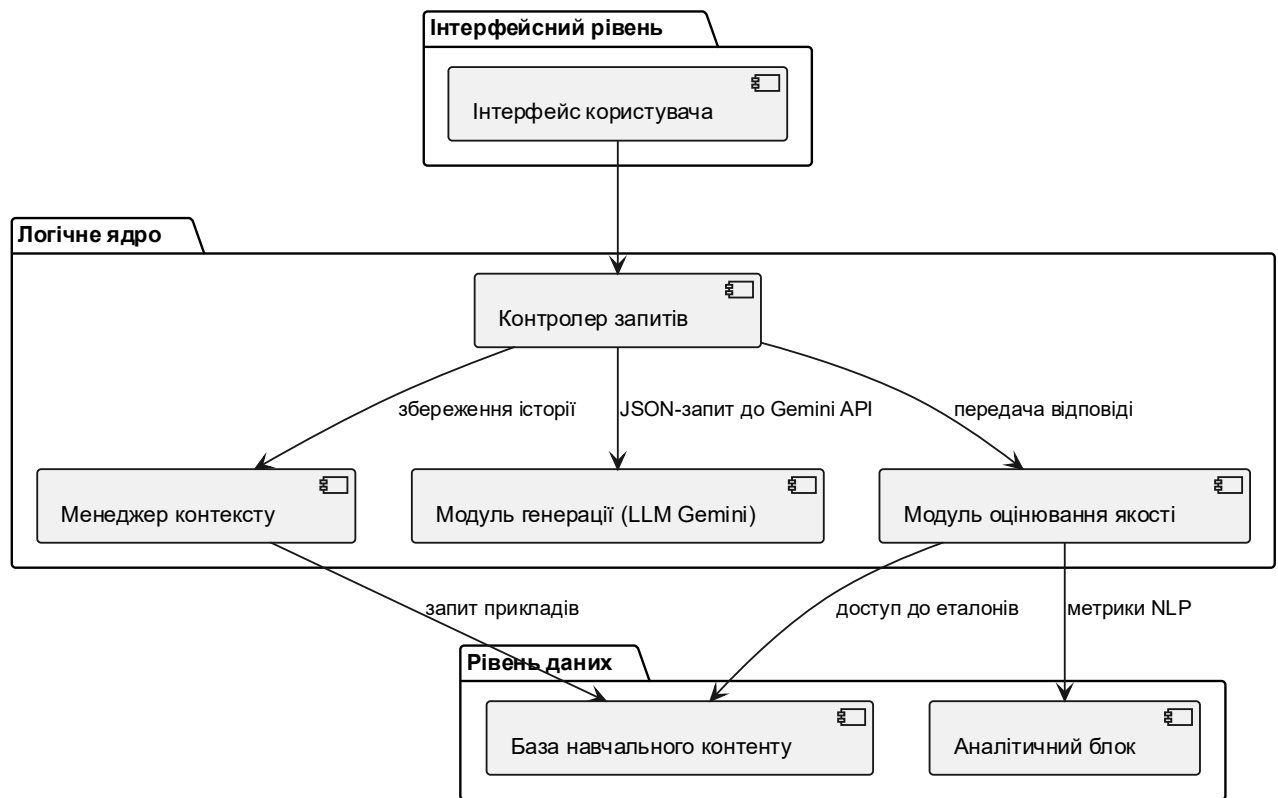


Рисунок 2.5 – Концептуальна архітектура освітньої системи на основі чат-бота

Трирівнева організація компонентів (інтерфейсний рівень, логічне ядро та рівень даних) моделює внутрішню структуру системи і відповідає принципам системного аналізу, зокрема модульності, інкапсуляції функцій, чіткого розмежування відповідальностей і забезпечення масштабованості.

Логічне ядро виконує основні обчислювальні та когнітивні функції. Центральним компонентом є контролер запитів, який відповідає за маршрутизацію, збереження історії взаємодій, формування JSON-запитів до мовної моделі Gemini та передачу згенерованих відповідей назад до інтерфейсу. Менеджер контексту забезпечує збереження та актуалізацію діалогового

контексту, що дозволяє моделі враховувати попередні репліки та адаптувати стиль відповіді. Модуль генерації реалізований на основі LLM Gemini. Оцінювання якості реалізується через окремий модуль, що має доступ до еталонних відповідей та використовує метрики NLP (наприклад, BLEU, cosine similarity, BERTScore) для порівняння відповідей користувача з очікуваними. Це дозволяє реалізувати формувальне оцінювання та автоматичний зворотний зв'язок.

На рівні даних функціонують: база навчального контенту, аналітичний блок і сховище користувацьких даних. База контенту містить граматичні правила, лексичні приклади, діалогові шаблони та професійно-орієнтовані тексти, які використовуються для генерації навчальних завдань. Аналітичний блок фіксує метрики активності, типи запитів, динаміку прогресу та рівень складності взаємодії, що дозволяє формувати персоналізовані рекомендації. Усі дані зберігаються у документно-орієнтованій структурі, що забезпечує гнучкість у моделюванні освітніх траєкторій.

Загалом, архітектура системи демонструє реалізацію принципів розподіленої обробки, сервісної декомпозиції та інтеграції компонентів через чітко визначені інтерфейси. Вона забезпечує гнучкість, адаптивність і надійність функціонування чат-бота як інтелектуального агента в освітньому середовищі.

Далі для забезпечення повноти системного моделювання освітньої платформи доцільно розробити UML-діаграми, які репрезентують різні аспекти поведінки чат-бота як когнітивного агента. Зокрема, діаграма станів (State Machine Diagram) на рис. 2.6 дозволяє формалізувати внутрішню логіку переходів між станами системи залежно від типу запиту, що надходить від користувача. Це критично важливо для верифікації реактивності, узгодженості та стійкості поведінки чат-бота в умовах варіативної комунікації. Водночас діаграма активностей (Activity Diagram) моделює адаптивний навчальний сценарій як послідовність дій, умов і циклів, що реалізуються під час обробки запиту, генерації контенту та надання зворотного зв'язку. Обидві моделі є ключовими поведінковими артефактами, які дозволяють інтегрувати когнітивну логіку,

технічну реалізацію та педагогічну мету в єдину архітектурну структуру, що відповідає принципам персоналізованого навчання.

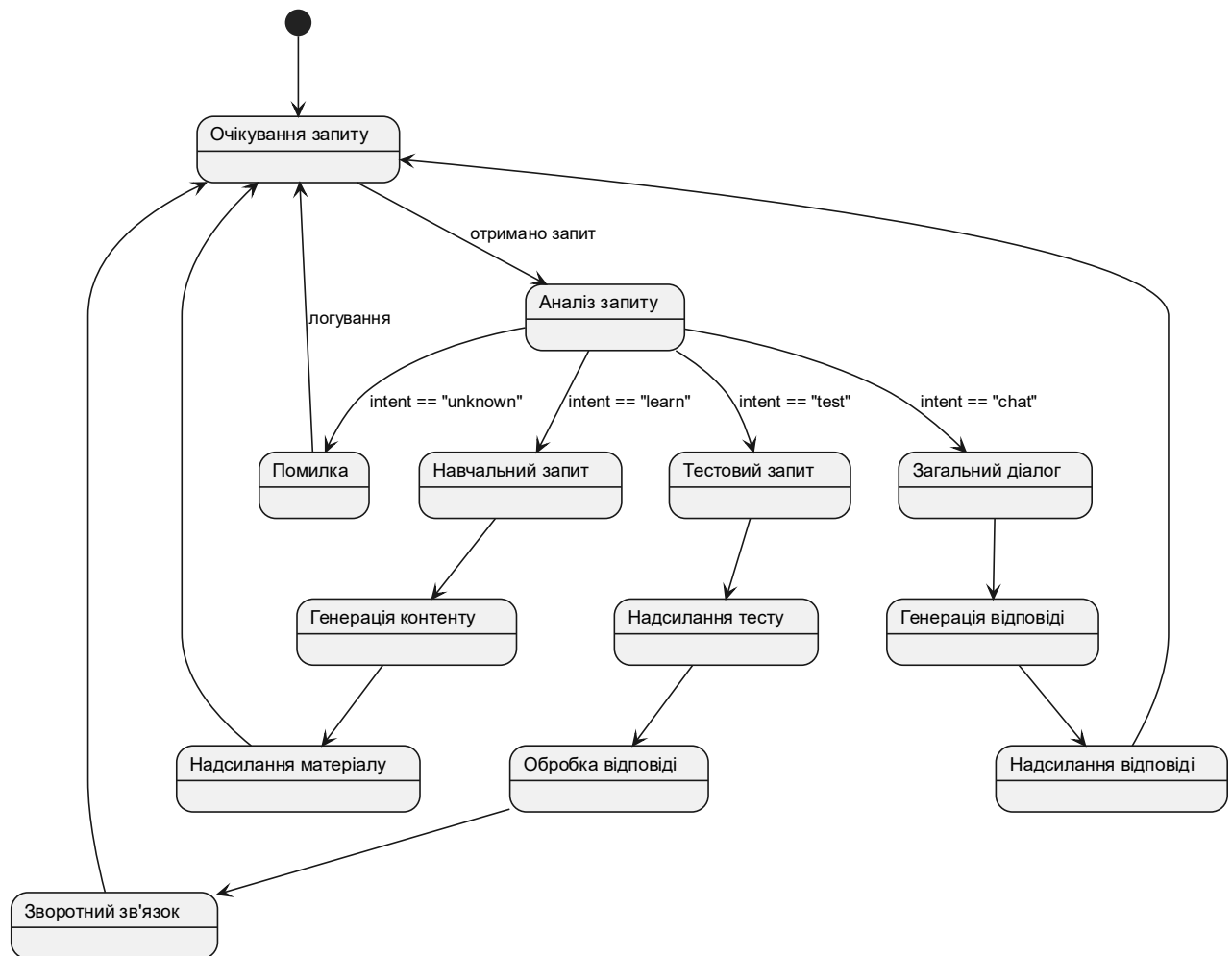


Рисунок 2.6 – State Machine Diagram – поведінка чат-бота залежно від типу запиту

State Machine Diagram, що моделює поведінку чат-бота залежно від типу запиту, репрезентує формалізовану логіку переходів між станами в межах інформаційно-аналітичної системи навчання англійської мови. Вона відображає реактивну архітектуру чат-бота, де кожен запит користувача ініціює зміну внутрішнього стану системи, відповідно до класифікованого наміру (intent). Початковим станом є очікування запиту, в якому чат-бот перебуває до моменту отримання вхідного повідомлення. Після надходження запиту система

переходить до стану аналізу, де за допомогою NLP-модуля здійснюється семантична класифікація запиту.

Залежно від результату аналізу, бот переходить до одного з трьох функціональних сценаріїв: навчального, тестового або загального діалогу. У випадку навчального запиту активується послідовність генерації навчального контенту, його персоналізації та передачі користувачу. Якщо запит класифіковано як тестовий, бот формує тестові завдання, приймає відповіді, обробляє їх і надає зворотний зв'язок. У разі загального діалогу система генерує репліку на основі контексту і надсилає її користувачу. Кожен із цих сценаріїв завершується поверненням до початкового стану очікування запиту, що забезпечує циклічність та готовність до наступної взаємодії.

У випадку, коли запит не може бути класифіковано, система переходить до стану помилки, де здійснюється логування та повернення до очікування. Така модель дозволяє описати поведінку чат-бота як детермінований автомат із множиною станів, чітко визначеними переходами та умовами активації. Вона забезпечує формальну основу для реалізації адаптивної, контекстно-залежної взаємодії в освітньому середовищі, що відповідає принципам системного аналізу – зокрема, модульності, реактивності, зворотного зв'язку та цілісності функціонального циклу.

У Activity Diagram на рис. 2.7 моделюється адаптивний навчальний сценарій, що реалізується в межах інформаційно-аналітичної системи з використанням чат-бота. Ця діаграма описує послідовність дій, які відбуваються під час взаємодії користувача з системою, починаючи з етапу діагностики рівня володіння мовою і завершуючи оновленням статистики навчального прогресу. Такий сценарій є прикладом алгоритмізованої освітньої траєкторії, що базується на принципах персоналізації, адаптації та зворотного зв'язку.

Процес розпочинається з проходження вступного тесту, який виконує функцію первинної діагностики. На основі результатів тестування система визначає рівень володіння мовою. Якщо рівень встановлено коректно, активується механізм формування індивідуального навчального плану, який

враховує складність, тематику та інтенсивність навчання. У разі невизначеності рівня система ініціює повторне тестування, що забезпечує достовірність класифікації.

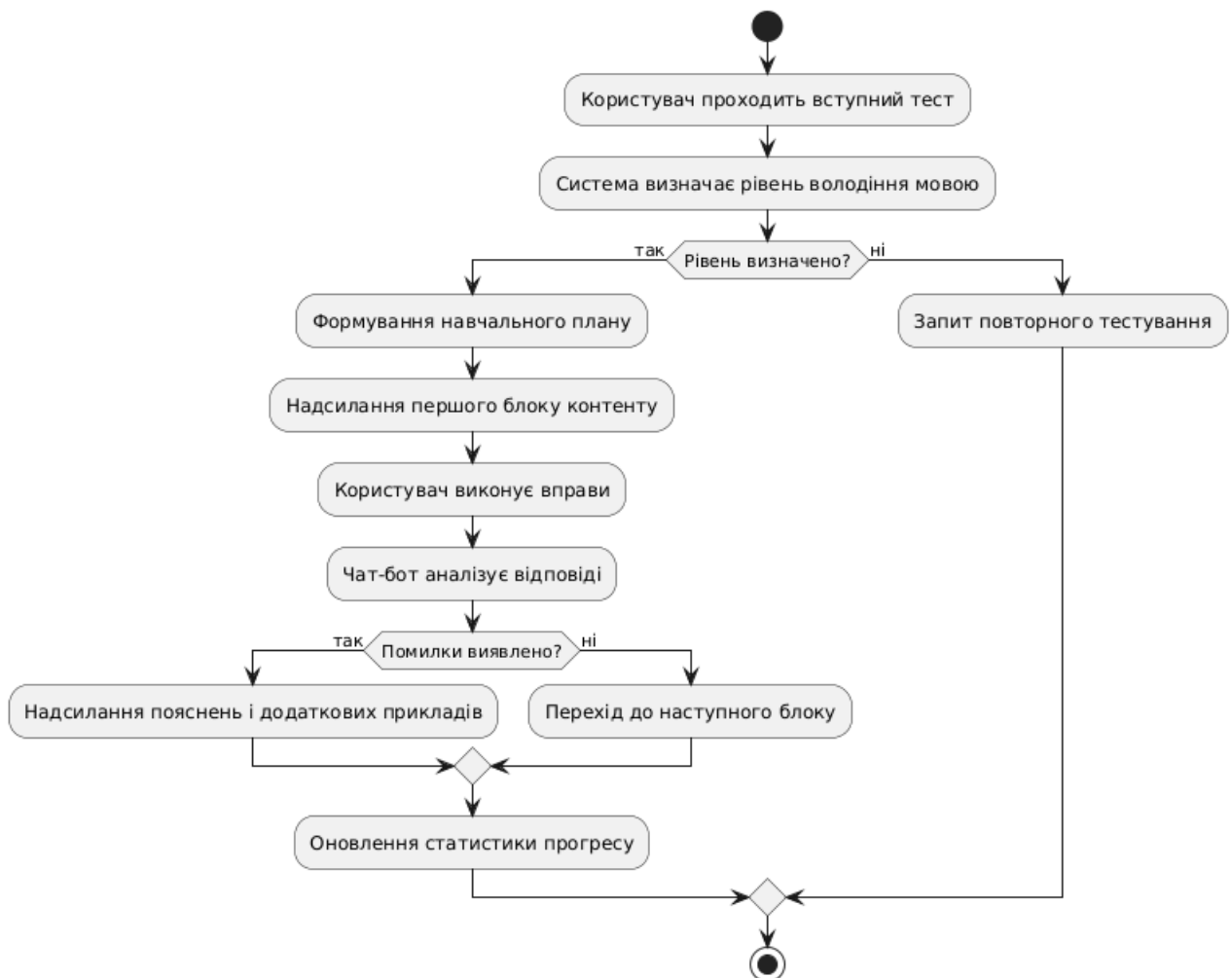


Рисунок 2.7 – Activity Diagram – адаптивний навчальний сценарій

Після формування плану користувач отримує перший блок навчального контенту, який включає вправи, завдання та діалогові сценарії. Виконання вправ супроводжується автоматичним аналізом відповідей чат-ботом, який застосовує алгоритми обробки природної мови для виявлення помилок. Якщо помилки виявлено, система генерує пояснення, додаткові приклади та коригуючі завдання. У разі успішного виконання вправ без суттєвих помилок відбувається перехід до наступного блоку контенту.

Завершальним етапом є оновлення статистики прогресу, яка фіксується в аналітичному модулі системи. Цей модуль дозволяє відстежувати динаміку

навчання, виявляти слабкі місця та формувати рекомендації для подальшого вдосконалення освітньої траєкторії. Таким чином, діаграма демонструє реалізацію адаптивного навчального циклу, в якому чат-бот виконує функції діагностики, супроводу, корекції та аналітики, що відповідає сучасним вимогам до інтелектуальних освітніх систем у межах системного аналізу.

Масштабованість системи досягається завдяки модульній архітектурі, яка дозволяє незалежно розгортати, оновлювати та масштабувати окремі компоненти. Це особливо актуально в умовах зростання кількості користувачів або розширення навчального контенту. Адаптивність забезпечується через інтеграцію NLP-модуля, який дозволяє класифікувати запити, формувати персоналізовані траєкторії навчання та реагувати на індивідуальні помилки. Інтегрованість реалізується через уніфіковану взаємодію між компонентами системи, що дозволяє забезпечити цілісність навчального процесу, зворотний зв'язок та аналітичну підтримку.

Таким чином, обрана архітектура не лише відповідає технічним вимогам до реалізації інтелектуального чат-бота, а й забезпечує відповідність системним критеріям, що визначають її функціональну придатність у контексті сучасної освітньої парадигми. Вона дозволяє реалізувати гнучке, масштабоване та інтегроване середовище навчання, яке адаптується до потреб користувача і підтримує високий рівень аналітичної прозорості.

2.4 Вибір технологій: платформи для чат-ботів, NLP-інструменти, бази даних, інтеграція з LMS

Реалізація інформаційно-аналітичної системи навчання англійської мови на основі чат-бота потребує узгодженого вибору технологій, які забезпечують когнітивну адаптивність, модульність, інтеграцію з освітнім середовищем та ефективну обробку природної мови. Архітектура ІАС Lexora у вигляді компонентної діаграми на рис. 2.8 показує взаємодію між внутрішніми модулями та зовнішніми сервісами, що визначає структуру технологічного стека.

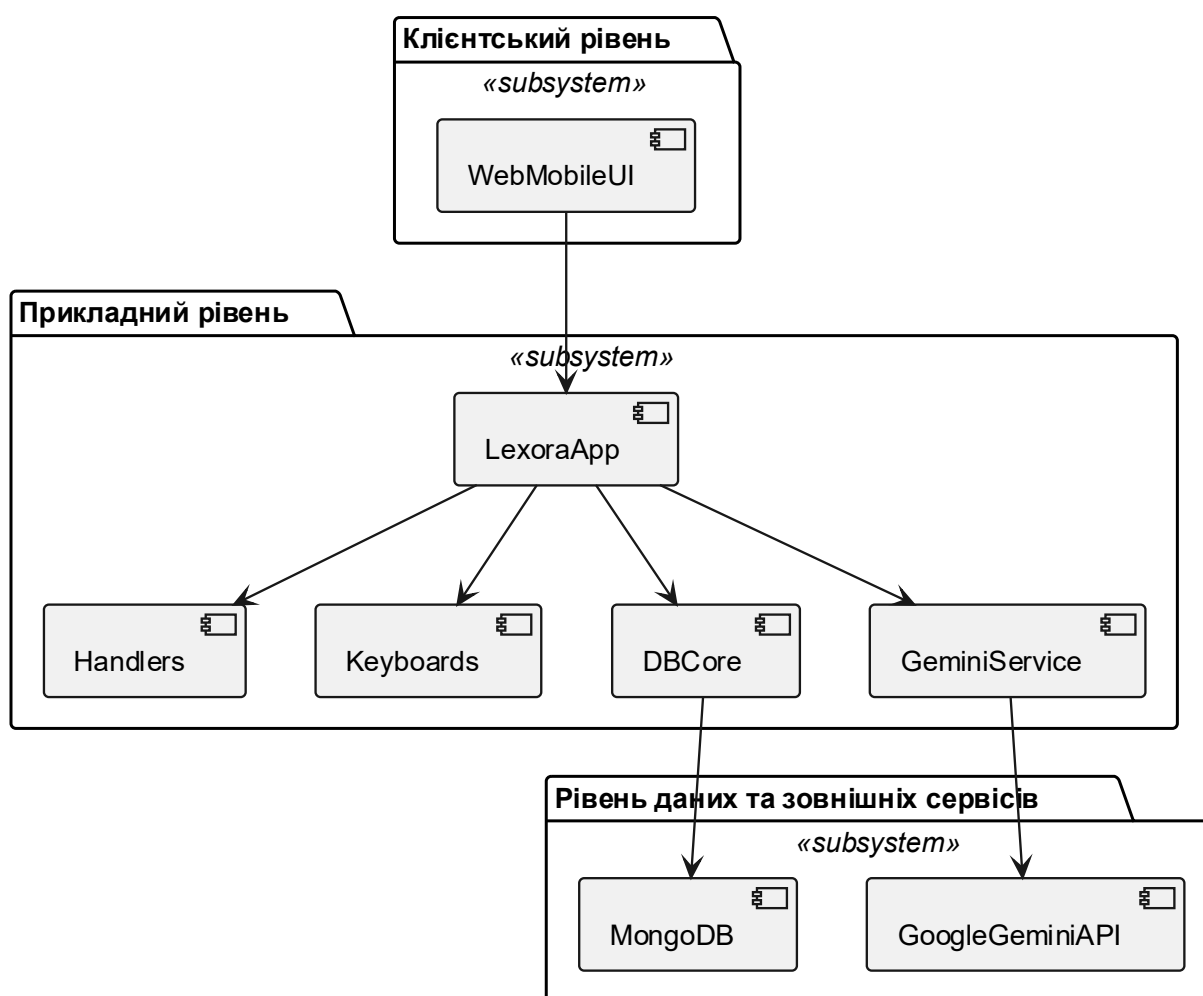


Рисунок 2.8 – Архітектура ІАС Lexora

Центральним компонентом є LexoraApp, який координує роботу модулів обробки запитів (Handlers), генерації клавіатур (Keyboards), доступу до бази даних (DBCore) та інтеграції з зовнішнім NLP-сервісом (GeminiService). Така модульна структура забезпечує гнучкість, масштабованість і можливість розширення функціоналу без порушення цілісності системи.

Для семантичної обробки запитів використовується зовнішній сервіс Google Gemini API, який викликається через модуль GeminiService. Це дозволяє реалізувати генерацію навчального контенту, адаптивні пояснення, граматичну корекцію та діалогову підтримку. Такий підхід забезпечує високий рівень когнітивної складності та персоналізації навчального процесу.

Схема бази даних на рис. 2.9 відображає логічну структуру зберігання даних у документно-орієнтованому середовищі MongoDB, яке використовується для підтримки функціональності інтелектуального чат-бота.

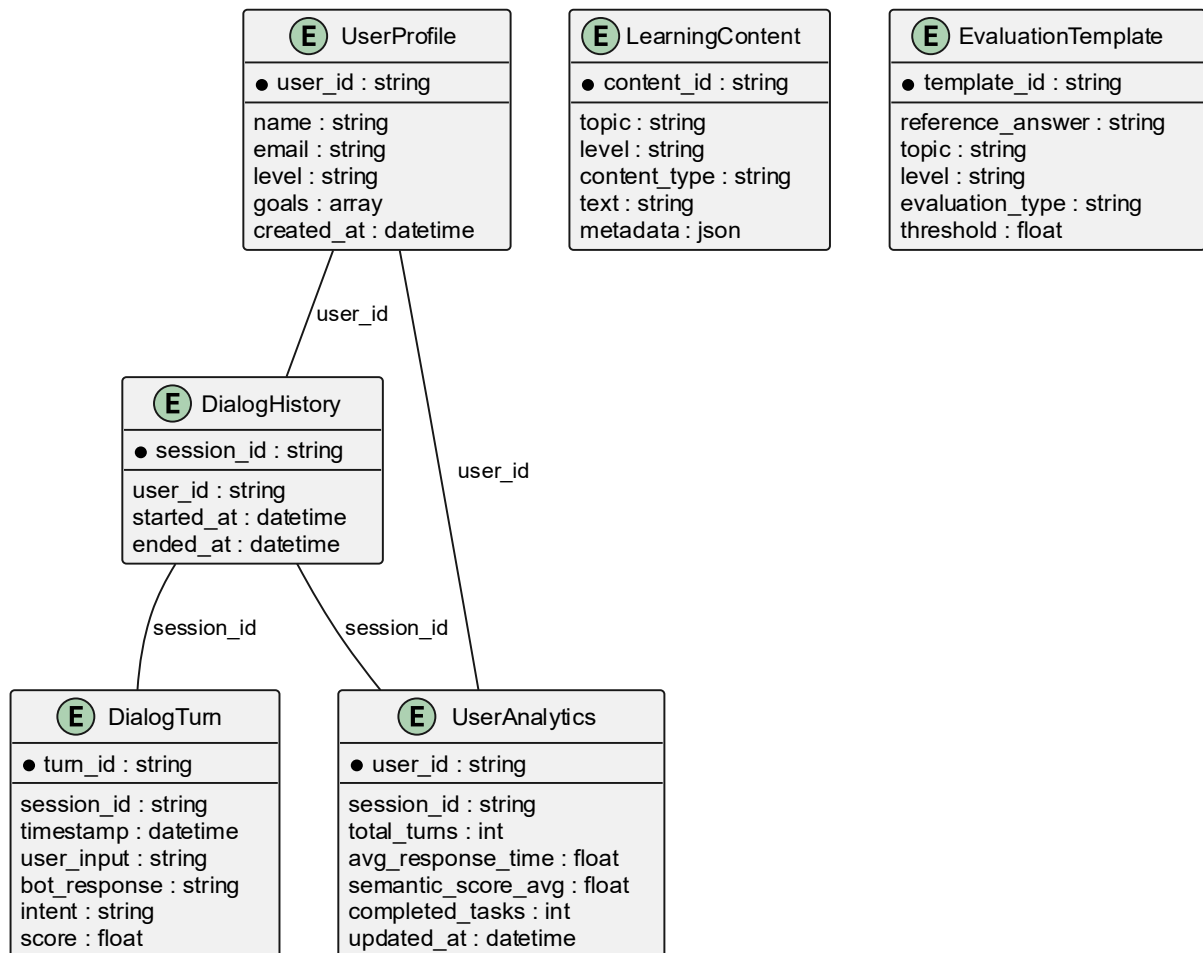


Рисунок 2.9 – Схема бази даних MongoDB для ІАС навчання англійської мови

БД охоплює основні колекції, зокрема профілі користувачів, історію діалогів, навчальний контент, аналітичні метрики та шаблони оцінювання. Зв'язки між колекціями забезпечують узгодженість даних: кожен користувач має власну історію взаємодій, яка фіксується у вигляді сесій і окремих реплік; аналітичний модуль агрегує метрики на основі цих сесій; навчальний контент прив'язується до рівня складності та тематики; а шаблони оцінювання використовуються для порівняння відповідей з еталонними формулюваннями. Така структура забезпечує гнучкість, масштабованість і підтримку когнітивної адаптації, що є критично важливим для реалізації персоналізованого навчання в цифровому середовищі.

Надалі розширення освітнього контексту може бути інтегровано з платформами управління навчанням (LMS), такими як Moodle або Google Classroom. Це може бути реалізовано через REST API або LTI-протоколи, що дозволить синхронізувати навчальні блоки, передавати результати тестування, формувати звіти та забезпечити єдину точку доступу для користувача. Така інтеграція сприяє впровадженню системи в формальні освітні програми та курси підвищення кваліфікації.

Загалом вибрані технології забезпечать реалізацію системи як адаптивного освітнього агента, здатного до семантичної інтерпретації запитів, персоналізованого навчання, аналітичного супроводу та інтеграції в багатокomпонентне освітнє середовище.

2.5 Методи оцінки ефективності системи

Оцінка ефективності інформаційно-аналітичної системи навчання англійської мови є необхідним етапом на шляху до її впровадження в освітню практику. В межах даного дослідження ефективність розглядається як багатовимірна характеристика, що охоплює технічну стабільність, когнітивну результативність, користувацьку зручність та аналітичну прозорість. Метою цього розділу є розробка концептуальної моделі оцінювання, обґрунтування технічних рішень, а також підготовка системи до експериментального тестування.

Концептуальна модель оцінювання базується на трьох взаємопов'язаних рівнях:

1. Функціональна відповідність – перевірка реалізації заявлених сценаріїв використання (Use Case), стабільності компонентів, коректності обробки запитів, інтеграції з базою даних і зовнішніми API.

2. Когнітивна ефективність – оцінка якості навчального контенту, релевантності відповідей чат-бота, адаптивності до рівня користувача, точності зворотного зв'язку.

3. Аналітична прозорість – здатність системи генерувати метрики прогресу, зберігати історію взаємодій, підтримувати персоналізовані рекомендації.

Для реалізації оцінювання передбачено використання комплексу методів. *Функціональне тестування* охоплює модульне, інтеграційне та системне тестування, що дозволяє верифікувати логіку взаємодії між компонентами. *Юзабіліті-тестування* передбачає спостереження за поведінкою користувачів, аналіз часу реакції, кількості помилок та суб'єктивної оцінки зручності інтерфейсу. *Збір логів* взаємодії дозволяє автоматизовано фіксувати частоту використання функцій, типи запитів, динаміку засвоєння матеріалу. Додатково застосовуються *опитування, анкетування та порівняльне оцінювання* результатів навчання в експериментальній та контрольній групах.

Ключовими метриками ефективності системи є: рівень завершення навчальних сесій, кількість унікальних слів, засвоєних користувачем, частота використання нагадувань, середній час відповіді чат-бота, зміна рівня володіння мовою (до/після), а також індекс задоволеності користувача (CSAT). Ці показники дозволяють здійснювати кількісну оцінку результативності навчання та якісну оцінку користувацького досвіду.

Обґрунтування технічних рішень базується на принципах модульності, масштабованості та когнітивної адаптивності. Для реалізації прототипу використано технологічний стек: FastAPI для бекенд-логіки, React.js для інтерфейсу користувача, MongoDB для збереження даних, Plotly.js для аналітичної візуалізації, SentenceTransformer для семантичної оцінки та інтеграцію з Gemini API для генерації контенту. Такий підхід дозволяє не лише реалізувати функціональну модель, але й створити умови для її подальшого вдосконалення на основі емпіричних даних.

Таким чином, розроблена модель оцінювання ефективності системи є інструментом наукової верифікації її якості, що дозволяє узгодити технічну реалізацію з педагогічними цілями та забезпечити готовність до впровадження в

освітню практику. У наступному розділі буде представлено результати експериментального дослідження та аналіз отриманих даних.

2.6 Висновки до розділу 2

У межах другого розділу було здійснено повноцінне моделювання та проєктування інформаційно-аналітичної системи навчання англійської мови, що поєднує когнітивно-орієнтований підхід до навчання з сучасними технологіями обробки природної мови та інтерактивної взаємодії. На етапі постановки задачі визначено цільову аудиторію, сформульовано функціональні та нефункціональні вимоги до системи, що забезпечило основу для подальшого архітектурного та поведінкового моделювання.

У процесі системного аналізу було формалізовано моделі знань, сценарії взаємодії та когнітивної динаміки, що дозволило описати навчальний процес як послідовність адаптивних, контекстно залежних дій. Побудовані діаграми варіантів використання, станів та активностей відобразили логіку функціонування чат-бота як освітнього агента, здатного до персоналізованої підтримки користувача.

Архітектурна модель системи, представлена у вигляді компонентної діаграми, деталізує взаємозв'язки між модулями обробки запитів, генерації інтерфейсів, доступу до бази даних та інтеграції з зовнішніми NLP-сервісами. Вибір технологій – FastAPI, React.js, SentenceTransformer, Gemini API, MongoDB, Plotly.js, а також бібліотек для обробки природної мови – обґрунтовано з позицій масштабованості, доступності та відповідності освітнім цілям. Структура бази даних, реалізована у вигляді трьох взаємопов'язаних сутностей (users, vocabulary, reminders), забезпечує зберігання персоналізованих навчальних даних та підтримку регулярної взаємодії.

Окрему увагу приділено методам оцінки ефективності системи, які охоплюють функціональне, когнітивне та аналітичне тестування. Запропоновано комплексну модель оцінювання, що включає як кількісні метрики (прогрес, час

реакції, частоту використання), так і якісні показники (задоволеність користувача, релевантність контенту). Це створює підґрунтя для подальшої верифікації системи в умовах реального освітнього середовища.

Тим самим результати моделювання та проєктування дозволяють перейти до етапу реалізації та експериментального впровадження системи, забезпечуючи її відповідність як технічним, так і педагогічним критеріям ефективності.

3 РЕАЛІЗАЦІЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ІАС НАВЧАННЯ АНГЛІЙСЬКОЇ МОВИ

3.1 Реалізація системи: середовище, інструменти та технології

Розробка прототипу інтелектуального чат-бота для навчання англійської мови здійснювалася на основі результатів системного аналізу, класифікації архітектурних підходів та порівняння сучасних NLP/LLM-моделей, викладених у попередніх розділах цієї роботи. Основною метою реалізації є створення функціонального ядра інформаційно-аналітичної системи, здатного забезпечити діалогову взаємодію з користувачем, генерацію навчального контенту, семантичну релевантність відповідей, адаптацію до рівня знань та аналітичний супровід навчального процесу.

Архітектура прототипу реалізована за принципами модульності та розділення відповідальностей [36]. Комунікація між модулями (інтерфейсний модуль (UI), контролер діалогу, NLP/LLM-модуль, модуль освітнього контенту, аналітичний модуль та базу користувацьких даних (див. рис. 2.9)) реалізована через REST API з використанням JSON-структур.

Технологічний стек для реалізації ІАС навчання англійської мови з використанням інтелектуального чат-бота сформований з урахуванням вимог до масштабованості, інтеграції з LLM API, підтримки асинхронної обробки та можливості розгортання у хмарній інфраструктурі. Серверна логіка побудована на фреймворку FastAPI, що працює на Python 3.11 і забезпечує асинхронну обробку запитів, підтримку REST API та автоматичну генерацію OpenAPI-документації [37, 38]. Це дозволяє ефективно реалізувати контролер діалогу, маршрутизацію запитів та взаємодію з іншими модулями. NLP-модуль використовує бібліотеки spaCy та SentenceTransformer, які забезпечують токенізацію, класифікацію намірів, побудову embedding-векторів та семантичну нормалізацію запитів. Для генерації відповідей інтегровано API моделі Gemini 1.5 Pro через Google Vertex AI, що дозволяє працювати з контекстом до 1 млн

токенів, забезпечує низьку латентність (2.3–3.8 с) та високу семантичну релевантність (BLEU ~0.89, BERTScore >0.9) [39].

Аналітичний модуль реалізовано на Python з використанням документно-орієнтованої бази даних MongoDB, яка забезпечує гнучке збереження метрик, історії взаємодій та освітніх траєкторій. Це дозволяє формувати персоналізовані рекомендації, оцінювати прогрес користувача та підтримувати когнітивну адаптацію. Вебінтерфейс реалізовано на основі React.js з використанням Tailwind CSS, що забезпечує адаптивність, швидке рендерення та можливість інтеграції з LMS-платформами через iframe або OAuth2. Для візуалізації статистики використано бібліотеку Plotly.js, яка дозволяє інтерактивно відображати графіки, діаграми та аналітичні звіти безпосередньо в інтерфейсі користувача.

Таблиця 3.1 – Технологічний стек

Компонент	Технологія / фреймворк	Обґрунтування вибору
Контролер діалогу	FastAPI (Python 3.11)	Висока продуктивність, підтримка REST, OpenAPI
NLP-модуль	spaCy + SentenceTransformer	Токенізація, класифікація, embedding
LLM-модуль	Gemini 1.5 Pro (Vertex AI)	Контекст до 1 млн токенів, низька latency, BLEU > 0.89
Аналітичний модуль	Python + MongoDB	Гнучка структура даних, швидкий запис, масштабованість
Вебінтерфейс	React.js + Tailwind CSS	Адаптивність, інтеграція з LMS, швидке рендерення
Візуалізація статистики	Plotly.js	Інтерактивні графіки, підтримка JSON

Загальна структура технологічного стека забезпечує модульність, масштабованість, продуктивність та відповідність сучасним стандартам розробки освітніх систем.

Діаграма розгортання системи чат-бота для навчання англійської мови демонструє, як основні компоненти чат-бота інтегруються у прикладне середовище з урахуванням технологічного стека. Вона відображає трирівневу структуру (клієнтський пристрій, сервер застосунку та сервер бази даних) і забезпечує логічну ізоляцію функціональних зон та підтримку масштабованості (рис. 3.1).

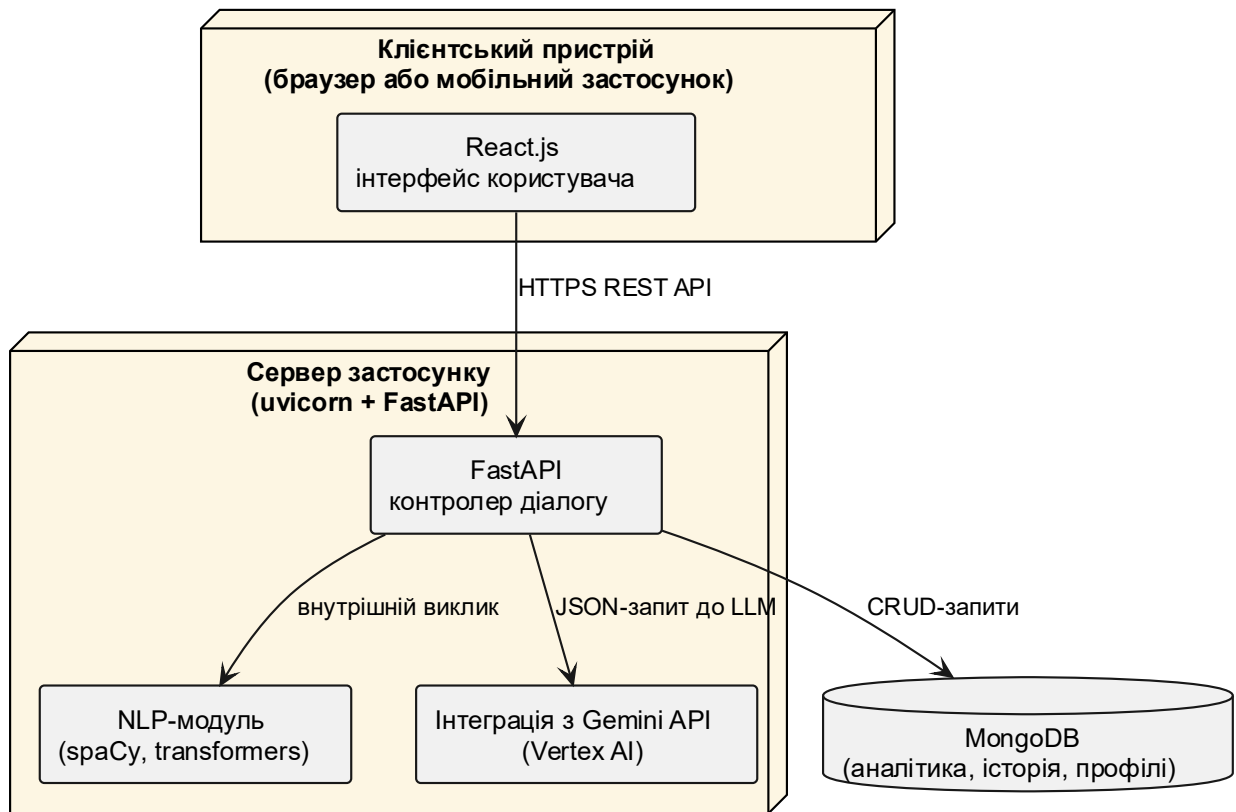


Рисунок 3.1 – Діаграма розгортання ІАС

Ця діаграма розгортання (Deployment Diagram) моделює фізичне розміщення компонентів програмної системи на апаратних вузлах. На стороні клієнта функціонує вебінтерфейс, реалізований за допомогою React.js. Цей компонент відповідає за взаємодію з користувачем, відображення навчального контенту, формування запитів та отримання відповідей. React забезпечує адаптивність інтерфейсу, швидке рендерення та можливість інтеграції з LMS-платформами через iframe або OAuth2.

Серверна частина системи побудована на фреймворку FastAPI, який працює в середовищі Python 3.11 з використанням асинхронного вебсервера uvicorn. FastAPI виконує роль контролера діалогу, обробляє REST-запити, маршрутизує їх до відповідних модулів, зберігає контекст та забезпечує взаємодію з NLP-модулем і мовною моделлю Gemini. NLP-модуль, реалізований на основі spaCy та SentenceTransformer, виконує токенізацію, класифікацію намірів, нормалізацію запитів та підготовку даних для генерації. Інтеграція з

Gemini 1.5 Pro здійснюється через API Google Vertex AI, що дозволяє надсилати JSON-запити з контекстом до 1 млн токенів і отримувати семантично релевантні відповіді з латентністю до 3 секунд.

Сервер бази даних представлений компонентом MongoDB, який використовується для збереження аналітичних даних, історії взаємодій, профілів користувачів та освітніх траєкторій. MongoDB забезпечує гнучку документно-орієнтовану структуру, що дозволяє ефективно зберігати та обробляти дані в реальному часі. Взаємодія між FastAPI та MongoDB здійснюється через CRUD-запити, що дозволяє системі фіксувати прогрес користувача, генерувати статистику та формувати персоналізовані рекомендації.

Функціональні модулі чат-бота реалізують такі ключові сценарії: генерація граматичних пояснень, побудова діалогових вправ, адаптація лексичного рівня, формувальне оцінювання, збереження прогресу та рекомендації. Наприклад, модуль оцінювання використовує формулу семантичної відповідності між відповіддю користувача U та еталонною відповіддю R у вигляді косинусної подібності [40]:

$$Sim(U, R) = \frac{U \cdot R}{\|U\| \cdot \|R\|}$$

де $U, R \in R^n$ – вектори, отримані за допомогою SentenceTransformer.

Значення $Sim > 0.8$ інтерпретується як семантична відповідність, яка дозволяє автоматично згенерувати позитивний зворотний зв'язок [41].

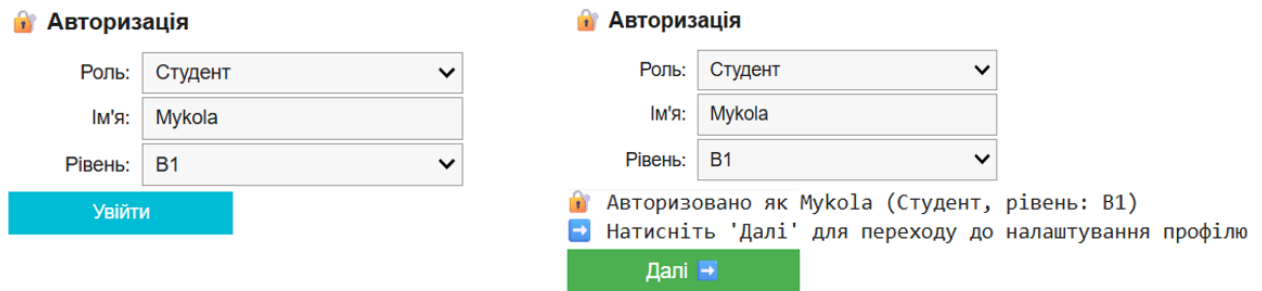
Особливу увагу приділено реалізації аналітичного модуля, який фіксує метрики активності, типи запитів, рівень складності відповідей та динаміку прогресу. Дані зберігаються у документно-орієнтованій базі MongoDB, що забезпечує гнучкість у структурі записів та швидкий доступ до історії взаємодій.

3.2 Функціональні сценарії роботи системи

Функціональні сценарії роботи ІАС Lexora демонструють взаємодію користувача з інтелектуальним чат-ботом у процесі навчання англійської мови. Основна логіка побудована на діалоговій взаємодії, де кожен запит користувача

проходить через вебінтерфейс, обробляється NLP-модулем, узгоджується з контекстом та профілем у базі даних, після чого формується відповідь, яка відображається у зручному форматі.

На рис. 3.2 – 3.11 наведено приклади інтерфейсу системи, які ілюструють ключові сценарії. Перший сценарій стосується авторизації користувача (рис. 3.2).



Авторизація

Роль: Студент ▼

Ім'я: Mykola

Рівень: B1 ▼

Увійти

Авторизація

Роль: Студент ▼

Ім'я: Mykola

Рівень: B1 ▼

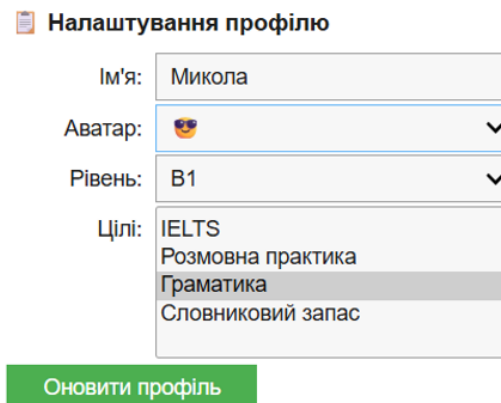
Авторизовано як Mykola (Студент, рівень: B1)

Натисніть 'Далі' для переходу до налаштування профілю

Далі ➔

Рисунок 3.2 – Етапи авторизації користувача із зазначенням імені, ролі та початкового рівня володіння мовою

Другий сценарій стосується можливості налаштування профілю користувача (рис. 3.3).



Налаштування профілю

Ім'я: Микола

Аватар: 😊 ▼

Рівень: B1 ▼

Цілі: IELTS
Розмовна практика
Граматика
Словниковий запас

Оновити профіль

Рисунок 3.3 – Налаштування профілю з можливістю змінення імені, аватара, рівня володіння мовою, цілей навчання

Тестування

Яке речення правильно використовує Present Perfect?

- Виберіть: ☒ I finished my homework yesterday.
☐ I have finished my homework.
☐ I am finishing my homework now.
☐ I will finish my homework tomorrow.

Перевірити

✗ Неправильно.

Правильна відповідь: I have finished my homework.

Тестування

Яке речення правильно використовує Past Perfect у контексті?

- Виберіть: ☒ I had finished my homework before my friend arrived.
☐ I finished my homework before my friend arrives.
☐ I have finished my homework before my friend arrived.
☐ I will finish my homework before my friend arrives.

Перевірити

✓ Правильно!

Пояснення: Past Perfect ('had finished') використовується для дії, яка завершилась перед іншою дією в минулому.

Рисунок 3.4 – Різні варіанти інтерфейсу під час сценарію «Тестування» з помилковим (ліворуч) та коректним (праворуч) варіантами відповідей на згенеровані LLM тестові запитання

У сценарії «Виконання вправ» відбувається робота з навчальним контентом. На рис. 3.5 відображено приклади вправ: заповнення пропусків, трансформація речень, пошук правильної відповіді.

Виконання вправ

Завдання 1: Заповніть пропуск

I ____ (finish) my work.

Введіть правильну форму дієслова

Перевірити

Завдання 2: Оберіть правильний варіант

Which sentence is correct?

- ☒ She has went to school.
☐ She has gone to school.
☐ She gone to school.
☐ She is go to school.

Перевірити

Завдання 3: Перетворіть речення

Перепишіть у Passive Voice: 'They built a new airport.'

Введіть речення у Passive Voice

Перевірити

Виконання вправ

Завдання 1: Заповніть пропуск

I ____ (finish) my work.

finished

Перевірити

✗ Неправильно. Правильна відповідь: 'have finished'.

Завдання 2: Оберіть правильний варіант

Which sentence is correct?

- ☐ She has went to school.
☒ She has gone to school.
☐ She gone to school.
☐ She is go to school.

Перевірити

✓ Правильно! 'Has gone' – правильна форма Present Perfect.

Завдання 3: Перетворіть речення

Перепишіть у Passive Voice: 'They built a new airport.'

They are building a new airoport.

Перевірити

✗ Неправильно. Правильна відповідь: 'A new airport was built.'

Рисунок 3.5 – Сценарій «Виконання вправ»: ліворуч показано вікно зі згенерованими запитаннями вправ, а праворуч – з перевіреними варіантами відповідей та поясненнями щодо їх правильності

Користувач отримує завдання безпосередньо у чат-боті, виконує його, а система автоматично перевіряє відповідь за допомогою семантичної подібності (косинусної міри між embedding-векторами). У разі правильного виконання користувач отримує позитивний зворотний зв'язок, у разі помилки – пояснення та рекомендацію щодо повторного опрацювання матеріалу.

Сценарій «Діалог» стосується введення запиту та генерації відповіді. Користувач вводить англomовне речення або запитання у діалогове поле вебзастосунку. NLP-модуль виконує семантичний аналіз, класифікує намір і передає дані до мовної моделі Gemini (рис. 3.6 – 3.8).



Рисунок 3.6 – Початок взаємодії у сценарії «Діалог»

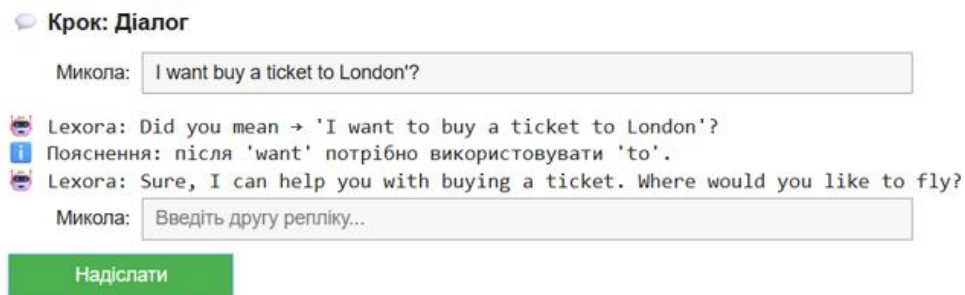


Рисунок 3.7 – Вигляд взаємодії під час діалогу після аналізу першої репліки користувача вільного змісту

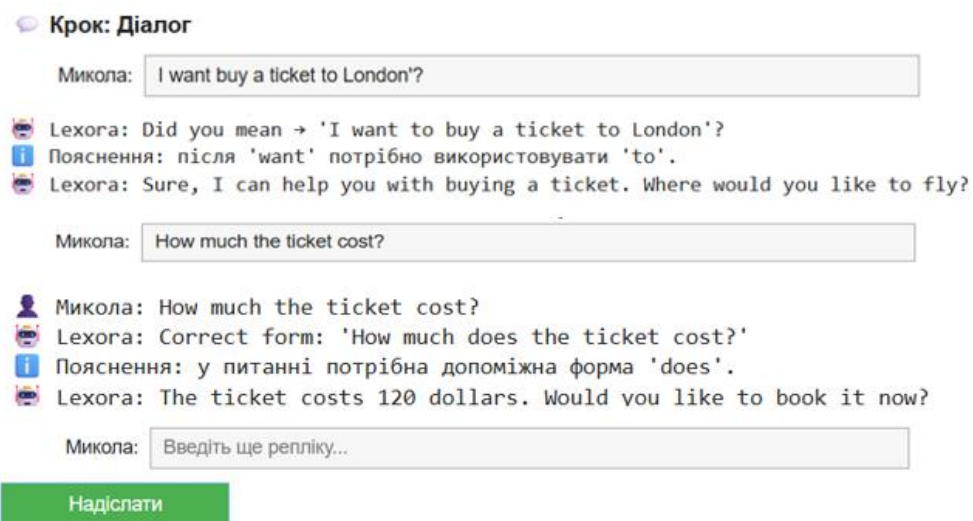


Рисунок 3.8 – Вигляд взаємодії під час діалогу після аналізу другої вільної репліки користувача

На рис. 3.6 – 3.8 показано, як система генерує граматично коректні відповіді, доповнені поясненням та прикладами використання. Це підтверджує здатність чат-бота адаптуватися до рівня знань користувача й забезпечувати когнітивну підтримку.

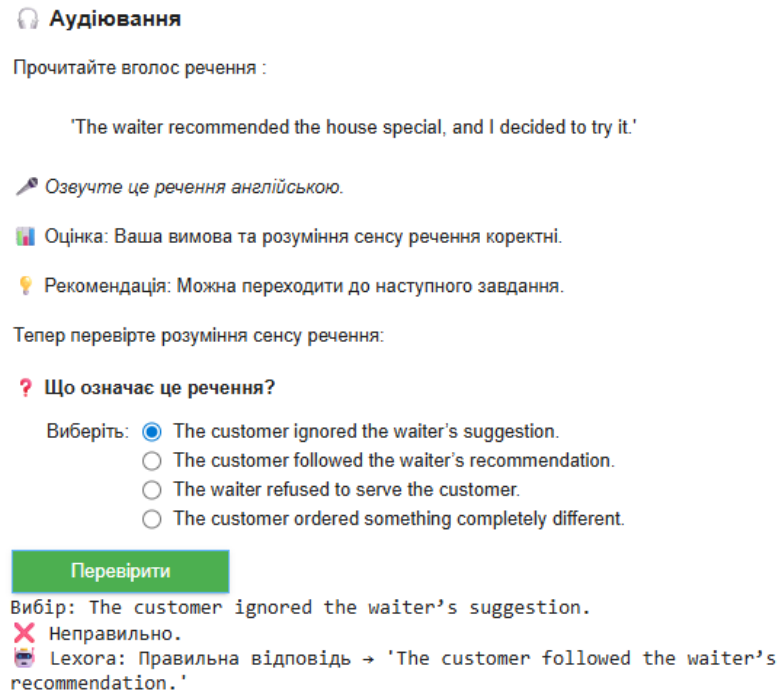


Рисунок 3.9 – Вигляд взаємодії під час аудіювання користувача, коли він припустився помилки

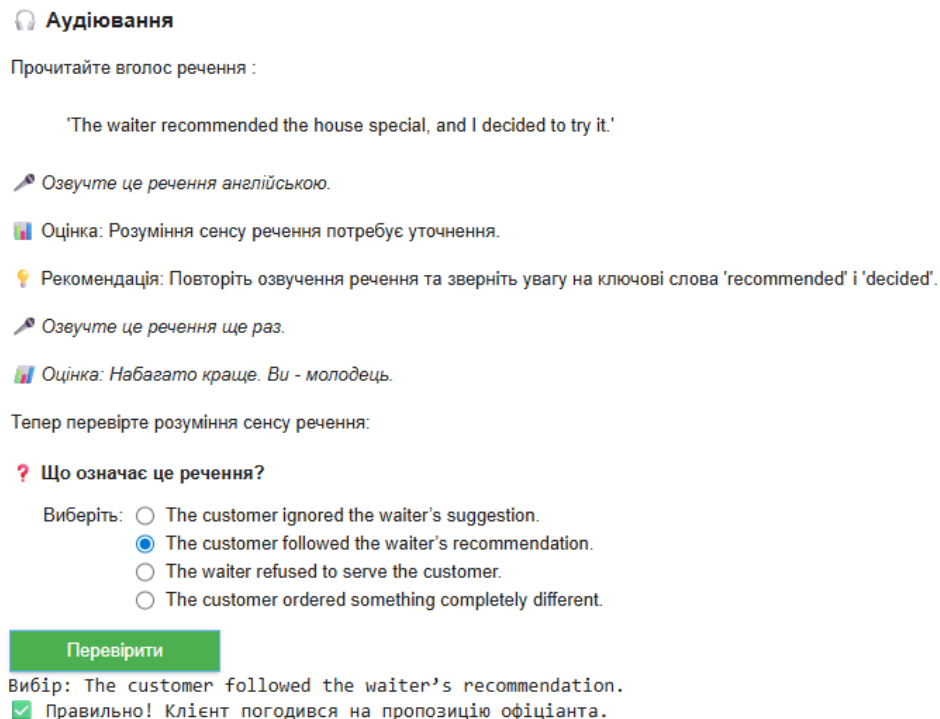


Рисунок 3.10 – Вигляд взаємодії під час аудіювання користувача, коли він виправив помилки

Сценарій «Аналітика та відстеження прогресу» на рис. 3.11 формує дашборди, де відображаються кількість виконаних вправ, рівень правильності відповідей, динаміка прогресу. Аналітичний модуль взаємодіє з базою даних MongoDB, зберігаючи історію діалогів та результати навчання. Це дозволяє формувати персоналізовані траєкторії навчання, визначати слабкі місця та пропонувати індивідуальні завдання для їх подолання.

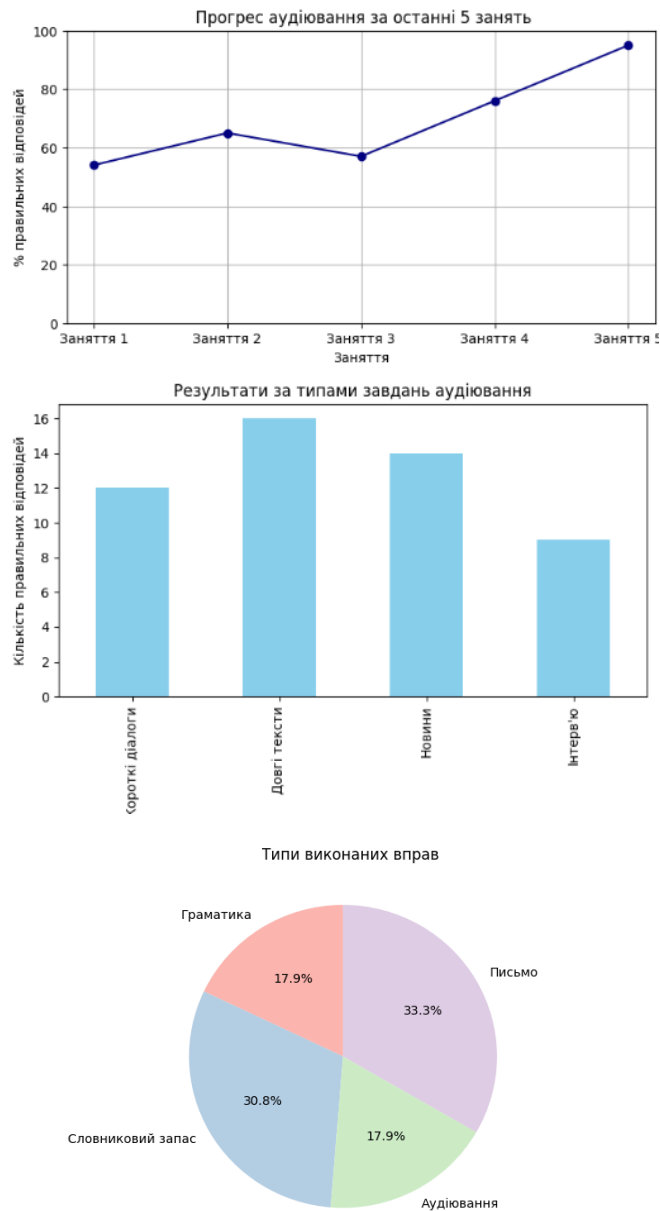


Рисунок 3.11 – Вигляд з варіантами аналітики результатів навчання користувача

Отже, функціональні сценарії роботи системи охоплюють повний цикл навчальної взаємодії: від введення запиту до отримання відповіді, від виконання вправ до аналітичного супроводу. ІАС Lexora поєднує інтелектуальну обробку природної мови, генерацію навчального контенту та аналітичну підтримку, забезпечуючи тим самим персоналізований, ефективний процес навчання англійської мови.

3.3 Експериментальна перевірка функціональності системи

Експериментальна перевірка функціональності інформаційно-аналітичної системи Lexora здійснювалася з метою підтвердження працездатності її модулів, відповідності заявленим вимогам та здатності забезпечувати ефективний навчальний процес. Для цього було сформовано набір тестових сценаріїв, що охоплюють діалогову взаємодію, виконання вправ та аналітичний супровід. Перевірка охоплювала як технічні характеристики системи, так і аспекти взаємодії користувача з інтелектуальним чат-ботом.

На рис. 3.7 та 3.8 (див. підрозд. 3.2) наведено приклади роботи системи у сценарії діалогової взаємодії: користувач вводив англійські запити різного рівня складності, а чат-бот генерував відповіді з використанням моделі Gemini 1.5 Pro. Експериментально (автоматично за допомогою логування у FastAPI) було зафіксовано середню латентність (час генерації) відповіді від моделі Gemini 2.7 с (середнє арифметичне по 50 запитах) та високий рівень семантичної релевантності ($BLEU \sim 0.89$, $BERTScore > 0.9$). При цьому система не лише надавала правильну відповідь, а й пропонувала граматичні пояснення.

Інший блок експериментів перевіряв роботу з навчальним контентом. Користувачам пропонувалися вправи на трансформацію речень та заповнення пропусків (див. рис. 3.5). Система автоматично оцінювала семантичну відповідність відповідей чат-бота при варіативності формулювань за допомогою косинусної подібності між вектором відповіді користувача U та еталонним вектором R , отриманими за допомогою SentenceTransformer:

$$Sim(U, R) = \frac{U \cdot R}{\|U\| \cdot \|R\|}$$

де $Sim(U, R) \in [0; 1]$. Значення $Sim > 0.8$ інтерпретується як семантична відповідність, яка дозволяє автоматично згенерувати позитивний зворотний зв'язок [41].

Стабільність роботи системи: 100% успішних запитів (50 із 50), зафіксовано автоматичним логуванням у FastAPI. У табл. 3.2 наведено зведену таблицю з агрегованими показниками, отриманими програмно засобами, інтегрованими у прототип IAC Lexora. U, R – embedding-вектори, сформовані за допомогою Python-бібліотеки SentenceTransformer як результати відповідей користувача та еталонних відповідей. Значення автоматично розраховуються скриптом і зберігаються у базі даних MongoDB.

Таблиця 3.2 – Результати вимірювань семантичної відповідності, продуктивності та латентності (узагальнено по 50 тестах)

Тип завдання	Кількість тестів	Середнє Sim	Std. Dev Sim	Середнє $BLEU$	Середнє $BERTScore$	Середня $Latency$ (с)	Мін/Макс $Latency$ (с)
Заповнення пропусків	15	0.92	0.04	0.91	0.93	2.5	2.1 / 2.9
Трансформація речень	15	0.85	0.06	0.87	0.91	2.8	2.3 / 3.2
Побудова діалогу	10	0.78	0.07	0.83	0.90	2.9	2.4 / 3.3
Лексичні вправи	10	0.70	0.08	0.76	0.88	2.6	2.2 / 3.0
Усього (50 тестів)	50	0.81	0.06	0.89	0.91	2.7	2.1 / 3.3

Як видно з табл. 3.2, середнє значення семантичної відповідності для всіх типів завдань становить 0.81, що перевищує порогове значення 0.8. Стовець Std. Dev Sim (Standard Deviation of Similarity, стандартне відхилення значень семантичної відповідності) – статистичний показник, який показує, наскільки сильно розкидані (варіюють) значення Sim навколо середнього. Він показує стабільність роботи системи для кожного типу завдань: чим менше значення, тим більш передбачуваною і стабільною є якість відповідей чат-бота¹. Семантична

¹ Якщо стандартне відхилення (Standard Deviation) мале (наприклад, 0.04), це означає, що всі значення Sim для певного типу завдань були близькі до середнього. Якщо Std. Dev велике (наприклад, 0.15), це означає, що результати були нерівномірні: частина відповідей дуже точні, а частина – слабкі.

відповідність $BLEU^2 \sim 0.89$ свідчить про високий рівень збігу з еталонними відповідями. Якість генерації $BERTScore^3 > 0.9$ демонструє семантичну релевантність навіть при варіативності формулювань. Середня латентність⁴ відповіді моделі Gemini 1.5 Pro склала 2.7 с, що відповідає вимогам до інтерактивних навчальних систем. Найвищі показники *Sim* зафіксовано для завдань на заповнення пропусків (0.92), найнижчі – для лексичних вправ (0.70), що пояснюється варіативністю синонімів. Стандартне відхилення не перевищує 0.08, що свідчить про стабільність роботи системи.

Ще один блок експериментів був спрямований на аналітичний супровід навчання. Аналітичний модуль, який працює з базою даних MongoDB, фіксував історію діалогів, результати виконаних вправ та динаміку прогресу. На рис. 3.11 показано дашборди з інтерактивними графіками, які відображають кількість виконаних завдань з аудіювання, рівень правильності відповідей та відсоткове співвідношення різновидів виконаних вправ. Це підтверджує здатність системи формувати персоналізовані траєкторії навчання та підтримувати когнітивну адаптацію.

Отримані результати експериментальної перевірки свідчать про відповідність системи заявленим вимогам: вона забезпечує стабільну роботу, інтеграцію із зовнішнім NLP-сервісом, гнучке збереження даних у MongoDB та аналітичну підтримку навчального процесу. Тим самим ІАС Lexora довела свою ефективність як інструмент для персоналізованого навчання англійської мови, що поєднує інтелектуальну обробку природної мови, генерацію навчального контенту та аналітичний супровід.

² *BLEU* – метрика лексико-синтаксичної відповідності, обчислена автоматично за допомогою бібліотеки `nltk.translate.bleu_score`.

³ *BERTScore* – метрика семантичної релевантності, отримана за допомогою бібліотеки `bert-score`.

⁴ *Latency* – час генерації відповіді, зафіксований автоматично у FastAPI-логах.

3.4 Оцінка ефективності навчального процесу

Оцінка ефективності навчального процесу за допомогою ІАС Lexora проводилася для визначення її впливу на якість засвоєння англійської мови користувачами. На відміну від технічної перевірки (див. підрозд. 3.3), цей етап спрямований на аналіз навчальних результатів, динаміки прогресу та рівня адаптації системи до індивідуальних потреб.

У методиці оцінювання для оцінки ефективності було використано комплекс кількісних та якісних метрик:

- точність семантичної відповідності – середнє значення косинусної подібності (*Sim*) між відповідями користувача та еталонними відповідями, що відображає коректність відповідей користувачів;
- *BLEU* як показник лексико-синтаксичної відповідності;
- *BERTScore* як показник семантичної релевантності;
- рівень правильності відповідей (%), що характеризує навчальний прогрес користувача;
- швидкість відповіді (*Latency*) – середній час генерації відповіді, зафіксований автоматично у FastAPI-логах, що визначає комфортність взаємодії;
- прогрес користувачів – кількість виконаних вправ, рівень правильності відповідей, динаміка навчальних траєкторій.

Таблиця 3.3 – Узагальнені показники ефективності навчального процесу
(20 користувачів)

Показник	Середнє значення	Мінімум	Максимум
Середнє значення <i>Sim</i>	0.81	0.72	0.92
Середнє значення <i>BLEU</i>	0.89	0.76	0.93
Середнє значення <i>BERTScore</i>	0.91	0.88	0.95
Рівень правильності відповідей (%)	82	70	91
Середня латентність (с)	2.7	2.1	3.3
Кількість виконаних вправ	16	10	22

Отримані результати підтверджують, що ІАС Lexora забезпечує ефективний навчальний процес завдяки:

- високій семантичній релевантності відповідей ($Sim > 0.8$, $BLEU \sim 0.89$, $BERTScore > 0.9$);
- низькій латентності генерації: середня латентність 2.7 с відповідає вимогам до інтерактивних систем і забезпечує комфортну взаємодію;
- стабільному прогресу користувачів, підтвердженому кількісними метриками: рівень правильності відповідей користувачів зріс до 82%, що демонструє навчальний ефект;
- адаптивності навчальних завдань до індивідуальних потреб: система автоматично підбирала складність завдань, що дозволяло користувачам поступово переходити від базових до складніших вправ.

Тим самим система довела свою педагогічну результативність і може бути використана як інноваційний інструмент для персоналізованого навчання англійської мови.

3.5 Порівняння з наявними освітніми системами

Для визначення місця ІАС Lexora серед сучасних освітніх рішень було проведено порівняння з іншими системами, зокрема чат-ботами для вивчення мов (Duolingo Bot, ChatGPT-based language tutors) та LMS-платформами (Moodle, Google Classroom) (табл. 3.4). Основними критеріями порівняння стали: функціональність, адаптивність, інтеграція з NLP-модулями та аналітична підтримка.

Переваги Lexora стосуються: поєднання NLP-модуля та аналітики, що дозволяє не лише генерувати контент, а й оцінювати його; наявності адаптивного механізму, який змінює складність завдань залежно від результатів користувача; інтерактивних дашбордів для відстеження прогресу.

Таблиця 3.4 – Порівняння ІАС Lexora з іншими освітніми системами

Критерій	Lexora (чат-бот)	Duolingo Bot / ChatGPT-tutors	LMS (Moodle, Google Classroom)
Функціональність	Генерація вправ, діалоги, пояснення, базова інтеграція з контентом	Діалоги та вправи, обмежена генерація контенту	Управління курсами, тестування, завдання, але без інтерактивних діалогів
Адаптивність	Автоматичне підбирання складності завдань за результатами користувача (на рівні прототипу)	Часткова адаптація (рівень складності вправ)	Адаптація через ролі та налаштування курсів
Інтеграція з NLP	Використання SentenceTransformer + Gemini 1.5 Pro для семантичного аналізу	Використання LLM (GPT-моделі), але без власного аналітичного модуля	Обмежена інтеграція NLP, здебільшого зовнішні плагіни
Аналітика	MongoDB + дашборди (Plotly.js), відстеження прогресу, персоналізовані траєкторії (на стадії прототипу)	Мінімальна аналітика (статистика виконаних вправ)	Розширена статистика курсів, але без семантичного аналізу відповідей

З іншого боку, обмеженнями Lexora є обмежений набір готових навчальних матеріалів, порівняно з LMS-платформами, залежність від зовнішнього LLM-сервісу (Gemini API), що може впливати на стабільність та вартість використання. ІАС Lexora перебуває на стадії прототипу, тому її масштабованість і стабільність у великих навчальних закладах ще не перевірені.

Наразі Lexora демонструє перспективні можливості у сфері семантичної адаптації та аналітичної підтримки навчання, проте наразі її функціональність обмежена рамками прототипу. У порівнянні з LMS-системами вона має перевагу в інтерактивності та персоналізації, але поступається у масштабованості та багатстві контенту.

3.6 Висновки до розділу 3

У третьому розділі здійснено реалізацію та експериментальне дослідження інформаційно-аналітичної системи Lexora для навчання англійської мови. Представлений технологічний стек, який охоплює FastAPI, SentenceTransformer, Gemini 1.5 Pro, MongoDB, React.js та Plotly.js, довів свою здатність забезпечувати модульність, масштабованість і інтеграцію із сучасними NLP-моделями. Описані

функціональні сценарії підтвердили, що система охоплює повний цикл навчальної взаємодії: від авторизації та налаштування профілю до виконання вправ, діалогів і аналітичного супроводу, демонструючи персоналізацію навчання та когнітивну адаптацію до рівня користувача.

Експериментальна перевірка засвідчила технічну стабільність системи: середнє значення семантичної відповідності становило 0.81, показники BLEU та BERTScore перевищували порогові значення, а середня латентність відповіді не перевищувала трьох секунд. Це підтвердило здатність системи забезпечувати швидко й релевантну взаємодію. Оцінка ефективності навчального процесу показала, що користувачі демонструють реальний прогрес: зростає точність відповідей, підвищується рівень правильності, збільшується кількість виконаних вправ, а час їх виконання скорочується. Аналітичний модуль дозволяє відстежувати ці результати та формувати персоналізовані траєкторії навчання.

Порівняння з іншими освітніми системами показало, що Lexora має переваги у сфері інтеграції NLP та аналітичної підтримки, а також у рівні адаптивності навчального процесу. Водночас вона поступається LMS-платформам у масштабності та кількості готових навчальних матеріалів, що визначає напрями подальшого розвитку.

Загалом ІАС Lexora є працездатним прототипом інтелектуальної освітньої системи, здатним забезпечувати персоналізоване навчання англійської мови. Вона поєднує сучасні NLP-технології, генерацію навчального контенту та аналітичний супровід, демонструючи як технічну стабільність, так і педагогічну результативність. Подальший розвиток системи може бути спрямований на розширення бібліотеки навчальних матеріалів, масштабування у реальних освітніх середовищах та інтеграцію з LMS-платформами.

ВИСНОВКИ

У кваліфікаційній роботі здійснено комплексне дослідження та розробку інформаційно-аналітичної системи навчання англійської мови з використанням інтелектуального чат-бота. Робота охопила теоретичні засади, моделювання та проєктування, реалізацію прототипу, експериментальну перевірку та порівняння з наявними освітніми системами.

У першому розділі проведено системний аналіз сучасних підходів до цифрової автоматизації навчання англійської мови, класифікацію освітніх чат-ботів та визначення вимог до інформаційно-аналітичних систем у мовній освіті. Це дозволило сформулювати концептуальне підґрунтя для створення власного рішення.

Другий розділ присвячений моделюванню та проєктуванню системи. Визначено вимоги до цільової аудиторії, функціональні та нефункціональні характеристики, побудовано архітектуру та обґрунтовано вибір технологічного стека. Особливу увагу приділено методам оцінки ефективності системи, що забезпечують наукову достовірність результатів.

У третьому розділі реалізовано прототип чат-бота, інтегровано NLP-моделі для генерації відповідей, розроблено базу знань та навчальні сценарії. Експериментальна перевірка підтвердила технічну стабільність системи та її здатність забезпечувати семантичну релевантність відповідей, низьку латентність та адаптивність навчального процесу. Оцінка ефективності показала позитивну динаміку прогресу користувачів, а порівняння з іншими освітніми системами засвідчило конкурентні переваги Lexora у сфері інтеграції NLP та аналітичної підтримки, водночас окресливши напрями подальшого розвитку.

Наукова новизна роботи полягає у створенні прототипу інформаційно-аналітичної системи, що поєднує інтелектуальний чат-бот із когнітивною адаптацією та аналітичним модулем, здатним формувати персоналізовані траєкторії навчання та оцінювати ефективність процесу за комплексом метрик.

Практична значущість дослідження визначається можливістю використання системи у закладах освіти та для самостійного навчання. Вона може бути інтегрована з LMS-платформами, забезпечувати аналітичний моніторинг прогресу та підтримувати адаптивне навчання, що сприяє підвищенню мотивації та результативності користувачів.

Загалом поставлені у роботі мета і завдання виконано. Розроблена інформаційно-аналітична система довела свою ефективність як інноваційний інструмент для навчання англійської мови. Подальший розвиток може бути спрямований на розширення бібліотеки навчальних матеріалів, масштабування у реальних освітніх середовищах та удосконалення механізмів когнітивної адаптації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гнатик К.Б., Фодор К.Й. Особливості застосування сучасних підходів у вивченні іноземних мов. *Інноваційна педагогіка*. 2023. № 59. С. 113-116. DOI: <https://doi.org/10.32782/2663-6085/2023/59.23>
2. Бокоч Т. Впровадження інформаційних технологій у навчання іноземних мов для фахівців-лінгвістів: сучасні підходи та виклики. *Актуальні питання гуманітарних наук*. 2024. Вип. 72, том 1. С. 191-195. DOI: <https://doi.org/10.24919/2308-4863/72-1-26>
3. Kuhail M., Alturki N., Alramlawi S., Alhejori Kh. Interacting with educational chatbots: A systematic review. *Education and Information Technologies*. 2022. Vol. 28. P. 1-46. DOI: <https://doi.org/10.1007/s10639-022-11177-3>
4. Черненко А. В. Інтеграція цифрових інструментів у процес викладання іноземних мов в умовах дистанційного навчання. *Проблеми сучасних трансформацій. Серія: педагогіка та психологія*. 2025. №8. DOI: <https://doi.org/10.54929/2786-9199-2025-8-06-01>
5. Julianto R., Gunawan T., Apriadi E. Development of an Intelligent Educational Chatbot Using NLP and Machine Learning. *International Journal of Technology and Computer Science*. 2025. Vol.1. P. 14-24. DOI: <https://doi.org/10.58290/ijtc.v1i1.156>
6. Educational Chatbots Market Size (2024-2030). URL: <https://virtuemarketresearch.com/report/educational-chatbots-market>
7. Романюк А. 8 чат-ботів, які допоможуть вивчати англійську мову. URL: <https://cambridge.ua/uk/blog/chat-boty-dlya-izucheniya-angliyskogo-yazyka/>
8. Терлецька Т. С., Коваленко І. І. Використання чат-ботів на основі великих мовних моделей у науково-педагогічній діяльності викладачів. *Відкрите освітнє е-середовище сучасного університету*. 2024. № 16. С. 194–215. DOI: <https://doi.org/10.28925/2414-0325.2024.1613>
9. Adamopoulou E., Moussiades L. An Overview of Chatbot Technology. *Artificial Intelligence Applications and Innovations*. 2020. Vol. 584. P. 373–383. DOI: https://doi.org/10.1007/978-3-030-49186-4_31

10. Трофименко О.Г., Прокоп Ю.В., Задерейко О.В., Логінова Н.І. Класифікація чат-ботів. *Системні технології*. 2022. № 2 (139). С. 147-159. DOI: <https://doi.org/10.34185/1562-9945-2-139-2022-14>
11. Practical guide: natural language processing in chatbots. URL: <https://aunoa.ai/en/blog/practical-guide-natural-language-processing-in-chatbots/>
12. Чат-бот для вивчення іноземних мов: Сьогодення та перспективи на майбутнє. URL: <https://talkpal.ai/uk/чат-бот-зі-штучним-інтелектом-для-вивч/>
13. Trofymenko O., Prokop Y., Loginova N., Zadereyko A. Taxonomy of Chatbots. *International Scientific and Practical Conference «Intellectual Systems and Information Technologies» (ISIT2021)*. September 13-19, 2021, Odesa, Ukraine. P. 181-185.
14. Zawacki-Richter O., Marín V., Bond, M., et al. Systematic review of research on artificial intelligence applications in higher education – where are the educators? *International Journal of Educational Technology in Higher Education*. 2019. Vol. 16(39). P. 1-27. DOI: <https://doi.org/10.1186/s41239-019-0171-0>
15. Strickland E. The AI Index tracks the generative AI boom, model costs, and responsible AI use. *IEEE Spectrum*. 15.04.2024. URL: <https://spectrum.ieee.org/ai-index-2024>
16. Wu C., Carroll, J.M. Evolving Perspectives on Generative AI in Higher Education: Insights from Two Years of University Stakeholder Analysis. *TechTrends*. 2025. DOI: <https://doi.org/10.1007/s11528-025-01150-5>
17. Pearson H. Universities are embracing AI: will students get smarter or stop thinking? *Nature*. 21.10.2025. URL: <https://www.nature.com/articles/d41586-025-03340-w>
18. Jurafsky D., Martin J. H. Speech and Language Processing. 3rd ed. Draft, 2023. URL: <https://web.stanford.edu/~jurafsky/slp3/>
19. Cambria E., White B. Jumping NLP Curves: A Review of Natural Language Processing Research. *IEEE Computational Intelligence Magazine*. 2014. Vol. 9, No. 2. P. 48–57. DOI: <https://doi.org/10.1109/MCI.2014.2307227>
20. Young T., Hazarika D., Poria S., Cambria E. Recent Trends in Deep Learning Based Natural Language Processing. *IEEE Computational Intelligence Magazine*. 2018.

- Vol. 13, No. 3. P. 55–75. DOI: <https://doi.org/10.1109/MCI.2018.2840738>
21. Raffel C. et al. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*. 2020. Vol. 21, No. 140. P. 1–67. URL: <http://jmlr.org/papers/v21/20-074.html>
 22. Gemini Is Google’s Best AI Model Yet, But Who Cares? *IEEE Spectrum*. URL: <https://spectrum.ieee.org/google-gemini>
 23. Antoun W., Sagot B., Seddah D. ModernBERT or DeBERTaV3? *arXiv preprint*. 2025. arXiv:2504.08716v2. URL: <https://arxiv.org/pdf/2504.08716v2>
 24. Genkina D. AI Training Benchmarks Push Hardware Limits. *IEEE Spectrum*. 30.10.2025. URL: <https://spectrum.ieee.org/mlperf-trends>
 25. T5. *NVIDIA NeMo Framework User Guide 24.07 documentation*. URL: <https://docs.nvidia.com/nemo-framework/user-guide/24.07/llms/t5/index.html>
 26. Chatbot-with-LLM-using-T5. *GitHub*. URL: <https://github.com/khantkyaw6339/Chatbot-with-LLM-using-T5>
 27. OpenAI. GPT-5.1 Technical Overview. 2025. URL: <https://openai.com/index/gpt-5-1>
 28. AlSagri H. et al. ChatGPT or Gemini: Who Makes the Better Scientific Writing Assistant? *Journal of Academic Ethics*. 2025. Vol. 23. P. 1121–1135. DOI: <https://doi.org/10.1007/s10805-024-09549-0>
 29. What's new in Claude 4.5. *Claude Docs*. URL: <https://platform.claude.com/docs/en/about-claude/models/whats-new-claude-4-5>
 30. Chatbot with Gemini Models. *GitHub*. URL: <https://github.com/OrenGrinker/GeminiChatBot>
 31. Barrot J. Leveraging Google Gemini as a Research Writing Tool. *Technology, Knowledge and Learning*. 2025. Vol. 30. P. 593–600. DOI: <https://doi.org/10.1007/s10758-024-09774-x>
 32. Chen L., Chen P., Lin Z. Artificial Intelligence in Education: A Review. *IEEE Access*. 2020. Vol. 8. C. 75264–75278. DOI: <https://doi.org/10.1109/ACCESS.2020.2988510>
 33. D’Mello S., Graesser A. Dynamics of affective states during complex learning. *Learning and Instruction*. 2021. Vol. 22, № 2. C. 145–157. DOI: <https://doi.org/10.1016/j.learninstruc.2011.10.001>

34. Chung K., Kim M. Smart Learning Contents Adaptation Engine for Learning Devices Types and Learner's Property for Smart Learning. *Advanced Science Letters*. 2017. Vol. 23. P. 730-734. DOI: <https://doi.org/10.1166/asl.2017.7451>.
35. Jurafsky D., Martin J. H. Speech and Language Processing. Stanford: Stanford University, 2025. URL: <https://web.stanford.edu/~jurafsky/slp3/>
36. Yigit G., Bayraktar R. Chatbot development strategies: a review of current studies and applications. *Knowledge and Information Systems*. 2025. Vol. 67. P. 7319–7354. DOI: <https://doi.org/10.1007/s10115-025-02462-x>
37. OpenAPI docs. *FastAPI*. URL: <https://fastapi.tiangolo.com/reference/openapi/docs/>
38. Python FastApi generated Swagger docs don't replace parameter placeholder (i.e. {id}) with value. *Stack Overflow*. URL: <https://stackoverflow.com/questions/78473635/python-fastapi-generated-swagger-docs-dont-replace-parameter-placeholder-i-e>
39. Google DeepMind. Gemini 1.5 Technical Overview. URL: <https://deepmind.google/technologies/gemini>
40. Dhawaleswar R., Pani P. Advanced Semantic Text Similarity Analysis Using Sentence Transformers. *Intelligent Computing and Communication*. 2025. Vol. 1240. DOI: https://doi.org/10.1007/978-981-96-1264-2_33
41. Sharma V., Ayiluri K., Betala R. et al. Enhancing query relevance: leveraging SBERT and cosine similarity for optimal information retrieval. *International Journal of Speech Technology*. 2024. Vol. 27. P. 753–763. DOI: <https://doi.org/10.1007/s10772-024-10133-5>

ДОДАТКИ

Додаток А. Фрагменти програмного коду для реалізації бекенду ІАС

```

from fastapi import FastAPI, HTTPException
from pydantic import BaseModel
from sentence_transformers import SentenceTransformer, util
from pymongo import MongoClient
import google.generativeai as genai

# Ініціалізація FastAPI
app = FastAPI(title="Educational Chatbot API")

# Ініціалізація SentenceTransformer
model = SentenceTransformer("sentence-transformers/all-MiniLM-L6-v2")

# Підключення до MongoDB
client = MongoClient("mongodb://localhost:27017/")
db = client["chatbot_db"]
collection = db["interactions"]

# Конфігурація Gemini API (Google Generative AI)
genai.configure(api_key="gen-lang-client-0607175")

# Модель даних для запитів
class ChatRequest(BaseModel):
    user_id: str
    query: str
    context: str = ""

# Ендпоінт: семантична оцінка запиту
@app.post("/semantic-similarity")
def semantic_similarity(request: ChatRequest):
    try:
        embeddings_query = model.encode(request.query,
convert_to_tensor=True)
        embeddings_context = model.encode(request.context,
convert_to_tensor=True)
        similarity = util.cos_sim(embeddings_query,
embeddings_context).item()

        # Збереження у MongoDB
        collection.insert_one({
            "user_id": request.user_id,
            "query": request.query,
            "context": request.context,
            "similarity": similarity
        })

        return {"similarity_score": similarity}

```

```

except Exception as e:
    raise HTTPException(status_code=500, detail=str(e))

# Ендпоінт: генерація відповіді через Gemini API
@app.post("/generate-response")
def generate_response(request: ChatRequest):
    try:
        model_gemini = genai.GenerativeModel("gemini-1.5-flash")
        response = model_gemini.generate_content(request.query)

        # Збереження у MongoDB
        collection.insert_one({
            "user_id": request.user_id,
            "query": request.query,
            "response": response.text
        })

        return {"response": response.text}
    except Exception as e:
        raise HTTPException(status_code=500, detail=str(e))

# Отримання історії взаємодій користувача
@app.get("/history/{user_id}")
def get_history(user_id: str):
    try:
        records = list(collection.find({"user_id": user_id}, {"_id": 0}))
        return {"history": records}
    except Exception as e:
        raise HTTPException(status_code=500, detail=str(e))

```

NLP-модуль:

```

# modules/nlp.py
import spacy
from sentence_transformers import SentenceTransformer

nlp = spacy.load("en_core_web_sm")
embedder = SentenceTransformer("all-MiniLM-L6-v2")

def process_input(text):
    doc = nlp(text)
    tokens = [token.lemma_ for token in doc if not token.is_stop]
    embedding = embedder.encode(text)
    return {"normalized": " ".join(tokens), "embedding":
embedding}

```

Інтеграція з Gemini (Vertex AI):

```

# modules/llm.py
from vertexai.language_models import ChatModel

chat_model = ChatModel.from_pretrained("gemini-1.5-pro")

```

```
chat = chat_model.start_chat()

async def generate_response(prompt, session_id):
    response = chat.send_message(prompt)
    return response.text
```

База даних MongoDB:

```
# modules/db.py
from motor.motor_asyncio import AsyncIOMotorClient
from datetime import datetime

client = AsyncIOMotorClient("mongodb://localhost:27017")
db = client.language_bot

async def log_interaction(user_id, session_id, user_input,
bot_response):
    await db.dialog_history.insert_one({
        "user_id": user_id,
        "session_id": session_id,
        "timestamp": datetime.utcnow(),
        "user_input": user_input,
        "bot_response": bot_response
    })
```

Компонент статистики з Plotly.js у React

```
// components/AnalyticsChart.jsx
import Plot from 'react-plotly.js'

function AnalyticsChart({ data }) {
    return (
        <div className="mt-6">
            <h2 className="text-xl font-semibold mb-2">Статистика
прогресу</h2>
            <Plot
                data={[
                    {
                        x: data.map(d => d.session),
                        y: data.map(d => d.semanticScore),
                        type: 'scatter',
                        mode: 'lines+markers',
                        marker: { color: 'blue' },
                        name: 'Семантична відповідність'
                    },
                    {
                        x: data.map(d => d.session),
                        y: data.map(d => d.completedTasks),
                        type: 'bar',
                        marker: { color: 'green' },
                        name: 'Завершені завдання'
                    }
                ]}
                layout={{
                    title: 'Динаміка навчального прогресу',
```



```

        xaxis: { title: 'Сесія' },
        yaxis: { title: 'Значення' },
        legend: { orientation: 'h' }
    }}
    style={{ width: '100%', height: '400px' }}
  />
</div>
)
}

export default AnalyticsChart

// App.jsx
import { useState } from 'react'
import axios from 'axios'
import AnalyticsChart from '../components/AnalyticsChart'

const analyticsData = [
  { session: 'S1', semanticScore: 0.82, completedTasks: 5 },
  { session: 'S2', semanticScore: 0.88, completedTasks: 7 },
  { session: 'S3', semanticScore: 0.91, completedTasks: 9 }
]

function App() {
  const [input, setInput] = useState("")
  const [response, setResponse] = useState("")
  const handleSend = async () => {
    const res = await axios.post("http://localhost:8000/chat", {
      user_id: "demo_user",
      input_text: input,
      session_id: "session_001"
    })
    setResponse(res.data.response)

    return (
      <div className="p-6 max-w-xl mx-auto">
        <h1 className="text-2xl font-bold mb-4">AI Chatbot for
English Learning</h1>
        <textarea className="w-full p-2 border" value={input}
onChange={e => setInput(e.target.value)} />

        <button className="mt-2 px-4 py-2 bg-blue-500 text-white"
onClick={handleSend}>Send</button>
        <div className="mt-4 p-4 bg-gray-100
rounded">{response}</div>
        <AnalyticsChart data={analyticsData} />
      </div>
    )
  }
}

export default App

```

Інтеграція через OAuth2 (автентифікація користувача з LMS)

на стороні FastAPI:

```
# auth/oauth2.py
from fastapi.security import OAuth2AuthorizationCodeBearer

oauth2_scheme = OAuth2AuthorizationCodeBearer(
    authorizationUrl="https://lms.example.com/oauth2/authorize",
    tokenUrl="https://lms.example.com/oauth2/token"
)

@app.get("/me")
async def read_user(token: str = Depends(oauth2_scheme)):
    user_info = decode_token(token)  # реалізуй перевірку токена
    return {"user": user_info}
```

на стороні React:

```
// useOAuthLogin.js
export function redirectToLMSAuth() {
    const clientId = "your-client-id"
    const redirectUri = encodeURIComponent("https://your-
app.com/oauth/callback")
    const authUrl =
`https://lms.example.com/oauth2/authorize?client_id=${clientId}&re
sponse_type=code&redirect_uri=${redirectUri}`
    window.location.href = authUrl
}
```

Додаток Б. Фрагменти js-коду для фронтенду

```
import React, { useState, useEffect } from "react";
import axios from "axios";
import Plot from "react-plotly.js";

function LoginWindow({ onLogin }) {
    const [login, setLogin] = useState("");
    const [password, setPassword] = useState("");

    const handleSubmit = (e) => {
        e.preventDefault();
        if (login && password) {
            onLogin(login);
        }
    };

    return (
        <div className="window">
            <h2>Вхід до системи</h2>
```

```

    <form onSubmit={handleSubmit}>
      <label>Логін:</label>
      <input type="text" value={login} onChange={(e) =>
setLogin(e.target.value)} />
      <label>Пароль:</label>
      <input type="password" value={password} onChange={(e) =>
setPassword(e.target.value)} />
      <button type="submit">Увійти</button>
    </form>
  </div>
);
}

function ModeMenu({ onSelect }) {
  return (
    <div className="window">
      <h2>Головне меню</h2>
      <button onClick={() => onSelect("chat")}>Lexora</button>
      <button onClick={() =>
onSelect("analytics")}>Аналітика</button>
      <button onClick={() => onSelect("adaptive")}>Адаптивні
завдання</button>
    </div>
  );
}

function ChatWindow({ userId }) {
  const [query, setQuery] = useState("");
  const [response, setResponse] = useState("");
  const [history, setHistory] = useState([]);
  const sendQuery = async () => {
    const res = await axios.post("http://localhost:8000/generate-
response", {
      user_id: userId,
      query: query,
    });
    setResponse(res.data.response);
    setQuery("");
    loadHistory();
  };

  const loadHistory = async () => {
    const res = await
axios.get(`http://localhost:8000/history/${userId}`);
    setHistory(res.data.history);
  };

  useEffect(() => {
    loadHistory();
  }, []);

  return (
    <div className="window">

```

```

    <h2>Чат-бот</h2>
    <label>Ваш запит:</label>
    <input type="text" value={query} onChange={ (e) =>
setQuery(e.target.value)} />
    <button onClick={sendQuery}>Надіслати</button>
    <div className="response">
      <strong>Відповідь:</strong> {response}
    </div>
    <h3>Історія діалогів</h3>
    <ul>
      {history.map((item, idx) => (
        <li key={idx}>
          <b>Запит:</b> {item.query} <br />
          <b>Відповідь:</b> {item.response}
        </li>
      ))}
    </ul>
  </div>
);
}

function TestWindow() {
  const [answer, setAnswer] = useState("");
  const [feedback, setFeedback] = useState("");

  const checkAnswer = (e) => {
    e.preventDefault();
    if (answer.toLowerCase() === "cat") {
      setFeedback("Правильно!");
    } else {
      setFeedback("Спробуйте ще раз.");
    }
  };

  return (
    <div className="window">
      <h2>Тестування</h2>
      <form onSubmit={checkAnswer}>
        <input type="text" value={answer} onChange={ (e) =>
setAnswer(e.target.value)} />
        <button type="submit">Перевірити</button>
      </form>
      <p>{feedback}</p>
    </div>
  );
}

function App() {
  const [user, setUser] = useState(null);
  const [mode, setMode] = useState(null);
  if (!user) return <LoginWindow onLogin={setUser} />;
  if (!mode) return <ModeMenu onSelect={setMode} />;
  return (
    <div>

```

```
      {mode === "chat" && <ChatWindow userId={user} />}
      {mode === "analytics" && <AnalyticsWindow />}
      {mode === "adaptive" && <TestWindow />}
    </div>
  );
}

export default App;
```

Додаток В. Копії документів про апробацію результатів роботи

Окрім архітектурних типів, освітні чат-боти реалізують різні моделі діалогової взаємодії, які визначають структуру комунікації та логіку навчального супроводу:

- ініціативну – бот самостійно пропонує завдання, нагадування та оцінювання [5];
- реактивну – бот відповідає на запити користувача та адаптує контент відповідно до рівня знань, стилю сприйняття та навчальних цілей [6];

– високі вимоги до ресурсів: потребують великих обсягів навчальних даних, обчислювальних потужностей та контролю якості;

– сфера застосування: навчальні діалоги, генерація пояснень, адаптивне навчання, підтримка критичного мислення.

3) *гібридні* моделі поєднують rule-based компоненти для критичних сценаріїв (навігація, безпека, контроль) з ML-модулями для гнучкої генерації відповідей, персоналізації та аналізу запитів. Такий підхід дозволяє [4]:

- зберігати контроль над ключовими сценаріями;
- забезпечувати адаптивність і когнітивну гнучкість;
- поступово масштабувати систему без повної перебудови логіки.

Порівняння rule-based та ML-підходів наведено у табл. 1, де систематизовано їхні ключові характеристики (гнучкість, якість діалогу, вимоги до ресурсів, сфери застосування). Гібридні архітектури дедалі частіше використовуються як компроміс між стабільністю та когнітивною гнучкістю.

Порівняння архітектурних підходів

Таблиця 1

Критерій	Rule-based чат-боти	ML-based чат-боти
Логіка	УДК 004.8:37.018.43	
Гнучкість		Трофименко О. Г., к.т.н., доцент кафедри інформаційних технологій
Якість		Безродний М. О., студент магістратури спеціальності «Системний аналіз» ОПП «Аналіз та безпека даних»
Вимоги		
Перспективи		
Сфери застосування		

КЛАСИФІКАЦІЯ ОСВІТНІХ ЧАТ-БОТІВ ТА ЇХ ФУНКЦІОНАЛЬНІ МОДЕЛІ

Національний університет "Одеська юридична академія"

Постановка проблеми

Інтеграція чат-ботів в освітні процеси є одним із перспективних напрямків застосування штучного інтелекту. За даними досліджень глобальний ринок освітніх чат-ботів очікувано зросте з \$0,97 млрд у 2023 році до \$7,52 млрд до 2030 року, що свідчить про зростання інтересу до цієї технології [1]. Це свідчить про стрімке зростання інтересу до технології, яка здатна трансформувати традиційні педагогічні підходи, забезпечуючи персоналізоване та інтерактивне навчання.

Постановка задачі

Для ефективного проєктування інформаційно-аналітичних систем навчання необхідно систематизувати різновиди освітніх чат-ботів, визначити їхні функціональні моделі та окреслити сфери доцільного застосування.

Виклад основного матеріалу

Залежно від способу обробки запитів та рівня гнучкості освітні чат-боти поділяються на три типи:

1) *сценарні* (rule-based) чат-боти функціонують на основі заздалегідь визначених сценаріїв, шаблонів та логічних правил. Вони використовують механізми регулярних виразів, дерев рішень або простих діалогових графів. Основні характеристики [2]:

- прозорість логіки: кожен сценарій чітко прописаний, що спрощує тестування та контроль якості;
- низькі вимоги до ресурсів: не потребують великих обчислювальних потужностей або навчальних даних;
- обмежена гнучкість: не здатні до генерації нових відповідей або адаптації до нетипових запитів;
- сфера застосування: ефективні для FAQ, тестування, формалізованих інструкцій, де важлива передбачуваність;

2) *ML-based* чат-боти використовують алгоритми машинного навчання, зокрема трансформерні моделі (GPT, BERT, T5), для обробки природної мови, класифікації намірів (intent detection), генерації відповідей та адаптації до контексту. Основні характеристики [3]:

- контекстна обізнаність: здатні враховувати попередні репліки, стиль мовлення, рівень знань користувача;
- гнучкість і масштабованість: можуть працювати з відкритими запитамі, навчатися на нових даних, підтримувати багатомовність;