

ПОЛТАВСЬКИЙ ДЕРЖАВНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
Навчально-науковий інститут економіки, управління, права та
інформаційних технологій
Кафедра інформаційних систем та технологій

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти бакалавр

на тему: **«Розроблення чат-боту для сайту закладу освіти»**

Виконав: здобувач вищої освіти
за освітньою програмою
Інформаційні управляючі системи
спеціальності 126 Інформаційні системи
та технології
ступеня вищої освіти бакалавр
групи 126ІСТбд_31
Юдінцов Данило Миколайович
Керівник: Уткін Юрій Вікторович
Рецензент: Яхін Сергій Валерійович

Полтава – 2026 року

ПОЛТАВСЬКИЙ ДЕРЖАВНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
Навчально-науковий інститут економіки, управління, права та
інформаційних технологій
Кафедра інформаційних систем та технологій

Освітня програма Інформаційні управляючі системи
Спеціальність 126 Інформаційні системи та технології
Рівень вищої освіти перший (бакалаврський)

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ **Юрій УТКІН**
«10» червня 2025 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Юдінцова Данила Миколайовича

1. Тема кваліфікаційної роботи:
«Розробка чат-боту для сайту закладу освіти»,
Керівник роботи: к. т. н., доцент, доцент кафедри інформаційних систем та технологій Уткін Юрій Вікторович.
Затверджено наказом закладу вищої освіти від «10» квітня 2026 року № 384-ст
2. Строк подання здобувачем вищої освіти роботи «08» червня 2026 року.
3. Вихідні дані до роботи: офіційна документація Telegram Bot API <https://core.telegram.org/bots/api>, платформа-конструктор UChat <https://uchat.au/>, вебсайт Гімназії №35 Полтавської міської ради на базі Google Sites, <https://sites.google.com/view/pl35>, статистичні дані щодо функціонування вебсайту гімназії.
4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):
Розділ 1. Аналіз предметної області закладу середньої освіти
Розділ 2. Створення інформаційної бази та розробка чат-бота
Розділ 3. Впровадження, тестування та оцінка ефективності чат-бота
5. Перелік графічного матеріалу: схеми, рисунки, діаграми за темою та об'єктом дослідження
6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання отримав
Оцінювання економічної ефективності результатів дослідження	Дивнич О. Д., к. е. н., доцент, доцент кафедри економіки та публічного управління	01.04.2026 р.	29.05.2026 р.

7. Дата видачі завдання «10» червня 2025 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Вибір і затвердження теми роботи	02.06.2025 р. – 04.06.2025	
2	Складання і затвердження розгорнутого плану та завдання на кваліфікаційну роботу	05.06.2025 р. – 10.06.2025 р.	
3	Опрацювання джерел інформації	11.06.2025 р. – 15.06.2025 р.	
4	Збір, вивчення і обробка інформації, необхідної для виконання роботи	16.06.2025 р. – 20.06.2025 р.	
5	Виконання теоретико-методологічного розділу роботи	08.09.2025 р. – 01.11.2025 р.	
6	Виконання дослідницько-аналітичного розділу роботи	19.01.2026 р. – 14.02.2026 р.	
7	Виконання проєктно-рекомендаційного розділу роботи	16.02.2026 р. – 31.03.2026 р.	
8	Оцінювання економічної ефективності результатів дослідження	01.04.2026 р. – 29.05.2026 р.	
9	Оформлення тексту роботи	01.06.2026 р. – 06.06.2026р.	
10	Попередній захист роботи на кафедрі	08.06.2026 р.	
11	Доопрацювання роботи з урахуванням зауважень і пропозицій	09.06.2026 р. – 10.06.2026 р.	
12	Нормоконтроль	11.06.2026 р. – 15.06.2026 р.	
13	Захист кваліфікаційної роботи	16.06.2026 р. – 18.06.2026 р.	

Здобувач вищої освіти

Данило ЮДІНЦОВ

Керівник роботи

Юрій УТКІН

ЗМІСТ

ВСТУП	6
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ЗАКЛАДУ СЕРЕДНЬОЇ ОСВІТИ	8
1.1 Загальна характеристика закладу та мета дослідження.....	8
1.2 Аналіз цільової аудиторії та її інформаційних потреб.....	10
1.3 Програмно-технічне забезпечення закладу	12
1.4 Огляд наукових публікацій та обґрунтування технологічного рішення для розробки чат-бота.....	13
РОЗДІЛ 2 СТВОРЕННЯ ІНФОРМАЦІЙНОЇ БАЗИ ТА РОЗРОБКА ЧАТ-БОТА	17
2.1 Обґрунтування вибору платформи та технологій розробки.....	17
2.2 Дослідження підходів до побудови архітектури чат-ботів.....	23
2.3 Порівняльний аналіз інструментальних засобів розробки та вибір платформи	30
2.4 Формування інформаційної бази та проєктування логіки діалогів	38
2.5 Практична реалізація функціоналу чат-бота.....	40
РОЗДІЛ 3 ВПРОВАДЖЕННЯ, ТЕСТУВАННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ ЧАТ-БОТА	47
3.1 Обґрунтування вибору платформи та технологій розробки.....	47
3.2 Розробка методики та проведення тестування програмного продукту	51
3.3 Розрахунок економічних показників та оцінка ефективності впровадження	56
ВИСНОВКИ ЗА РЕЗУЛЬТАТАМИ ПРОХОДЖЕННЯ ПРАКТИКИ	62
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	63
ДОДАТКИ.....	

ВСТУП

Стрімкий розвиток цифрових технологій та курс на цифровізацію освіти в Україні, закріплений Концепцією «Нова українська школа» (Розпорядження КМУ № 988-р від 14.12.2016), формують нові вимоги до інформаційної взаємодії між закладами освіти та громадянами. Адміністративний персонал сучасних гімназій витрачає до 70% робочого часу на відповіді на однотипні запити від батьків та абітурієнтів (розклад, умови вступу, перелік документів), тоді як пік звернень (19:00-21:00) повністю виходить за межі робочого дня. Ефективним рішенням цієї проблеми є впровадження детермінованого чат-бота, що автоматизує обробку типових запитів у режимі 24/7 без залучення персоналу. Це зумовлює актуальність теми кваліфікаційної роботи.

Мета роботи – підвищення ефективності інформаційної взаємодії між адміністрацією навчального закладу та учасниками освітнього процесу шляхом розроблення та впровадження чат-бота для Гімназії №35 Полтавської міської ради.

Для досягнення мети визначено такі *завдання*:

1. Провести системний аналіз предметної області Гімназії №35: дослідити організаційну структуру, цільову аудиторію та наявні комунікаційні канали.
2. Здійснити порівняльний аналіз та обґрунтувати вибір технологічної бази: месенджера (Telegram Bot API), архітектурного підходу (детермінованої моделі) та інструментального середовища (платформи UChat).
3. Сформувати інформаційну базу, спроектувати логіку діалогів і реалізувати чат-бот із подальшою інтеграцією в Telegram та на вебсайт гімназії.
4. Провести комплексне тестування розробленої системи за методикою чек-ліста.
5. Розрахувати економічну ефективність впровадження та обґрунтувати доцільність проєкту.

Об'єкт дослідження – процес інформаційної взаємодії між адміністрацією Гімназії №35 та учасниками освітнього процесу.

Предмет дослідження – методи, засоби та технології розробки детермінованого чат-бота на базі платформи UChat та Telegram Bot API для автоматизації типових інформаційних запитів у закладі загальної середньої освіти.

Методи дослідження: системний аналіз, порівняльний аналіз, метод систематичного огляду (PRISMA), методи проєктування інформаційних систем, тестування програмного забезпечення, техніко-економічний аналіз.

Інформаційну базу становлять: офіційна технічна документація Telegram Bot API та платформи UChat; наукові публікації з баз Web of Science та Scopus; нормативно-правові акти України у сфері освіти; дані дослідницького порталу Statista; внутрішня документація та статистичні відомості Гімназії №35, зібрані під час переддипломної практики.

Практична значущість роботи полягає у розробці повністю функціонального двоканального чат-бота (Telegram + вебвіджет), готового до промислової експлуатації. Запропонований підхід є тиражованим та може бути адаптований для будь-якого іншого закладу загальної середньої освіти.

Структура та обсяг кваліфікаційної роботи. Робота складається зі вступу, трьох розділів, висновків, списку використаних джерел та 8 додатків. Загальний обсяг основної частини – 54 сторінок, містить 41 рисунків та 7 таблиць.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ЗАКЛАДУ СЕРЕДНЬОЇ ОСВІТИ

1.1 Загальна характеристика закладу та мета дослідження

Метою роботи є підвищення ефективності інформаційної взаємодії між адміністрацією навчального закладу та учасниками освітнього процесу шляхом розроблення та впровадження інтелектуального чат-боту для Гімназії №35 Полтавської міської ради. Заклад знаходиться за адресою: м. Полтава, майдан Холодноярський, 1. Гімназія є навчальним закладом, що забезпечує реалізацію права громадян на здобуття повної загальної середньої освіти. Основні реквізити та характеристики установи наведено у таблиці 1.1. Колектив гімназії налічує висококваліфікованих педагогічних працівників: 2 вчителі-методисти, 3 старших вчителі, 7 вчителів вищої категорії та інших фахівців.

Таблиця 1.1 – Загальні відомості про Гімназію №35

Параметр	Значення
Повна назва	КЗ «Гімназія №35 Полтавської міської ради»
Адреса	м. Полтава, Холодноярський майдан, 1
ЄДРПОУ	22547319
Рік заснування	1889
Ступінь освіти	I–II ступінь (початкова та базова середня)
Режим роботи	Пн–Пт, 8:00–17:00
Контактний телефон	(0532) 674117
Електронна пошта	pl35@i.ua
Загальна площа	839 кв. м (у т.ч. 68 кв. м здано в оренду)
Кількість класних кімнат	9 (загальна площа – 335 кв. м)
Їдальня	Так (48 посадкових місць)
Бібліотечний фонд	4056 примірників, у т.ч. 2560 підручників

Організаційна структура включає адміністративний корпус (директор, заступник з навчальної та виховної роботи, завідувач господарством), педагогічний колектив в кількості 16 чоловік, учнівський склад (134 учні) та батьківський комітет, власну бухгалтерію. Гімназія таким чином охоплює три

рівні управління, що зображено вигляді схеми в Додатку А. Стратегічний рівень представлений директором: загальне керівництво, фінансовий контроль та представництво в держорганах. Тактичний рівень включає педагогічний колектив: заступники директора з НВР, класні керівники, вчителі предметники та соціальна і психологічна служба. Операційний рівень формує допоміжний персонал (секретаріат, бібліотекар). Саме секретаріат є основним вузлом комунікації з батьками та учнями, а автоматизація через чат-бот безпосередньо розвантажить цей підрозділ.

Офіційний вебсайт закладу реалізований на платформі Google Sites та має адаптивний дизайн із чіткою ієрархічною структурою навігації і вбудованою функцією пошуку. Вбудована функція пошуку по сайту знаходить як сторінки, так і файли форматів .pdf та .pptx, що підтверджено практичним тестуванням на рисунку 1.1. Оновлення контенту здійснює вчитель інформатики, який через значне навчальне навантаження не завжди встигає своєчасно актуалізувати інформацію.

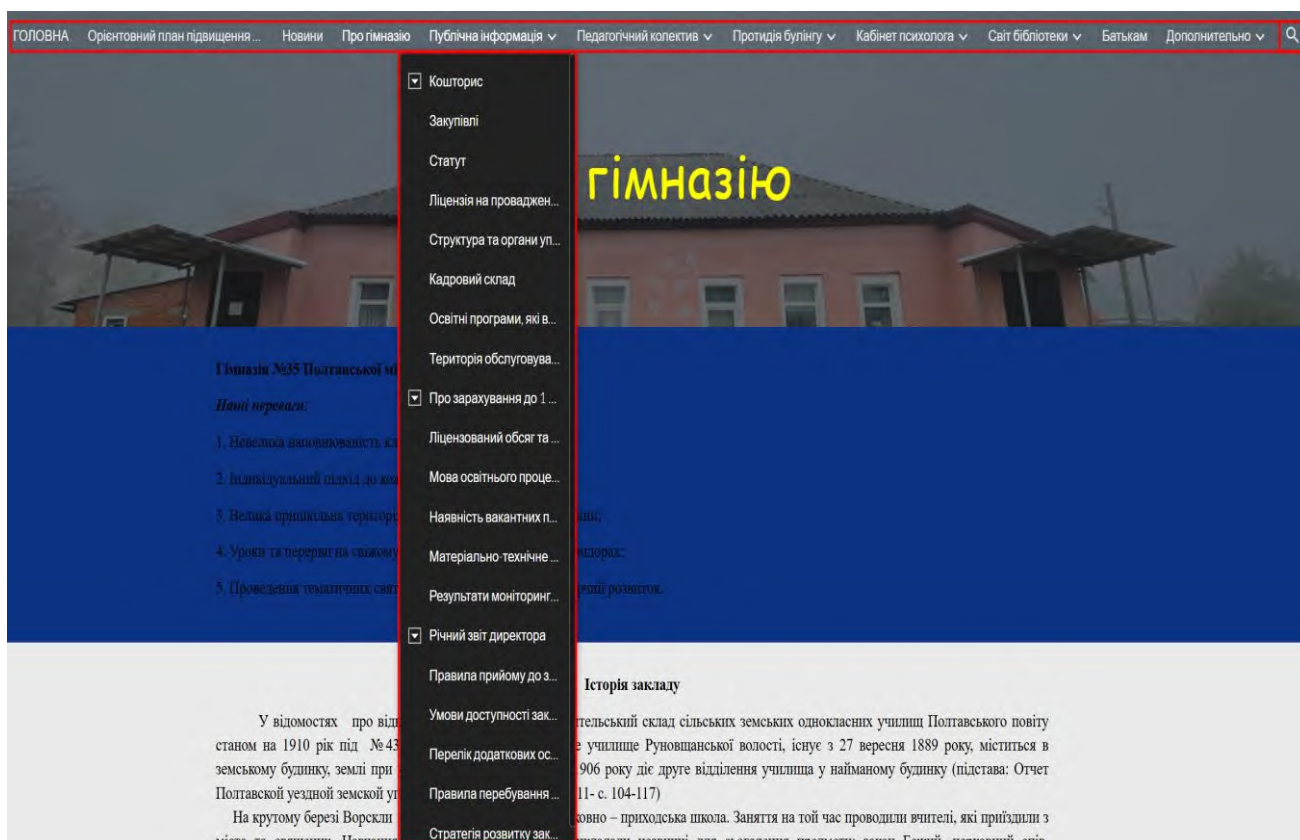


Рисунок 1.1 – Розгорнуте навігаційне меню сайту Гімназії №35

1.2 Аналіз цільової аудиторії та її інформаційних потреб

Під час дослідження було проаналізовано систему комунікації між адміністрацією Гімназії №35 та цільовою аудиторією. Основними каналами зовнішньої комунікації є: офіційний вебсайт (Google Sites), Telegram-канал, Viber-групи, Facebook-сторінка, стаціонарний телефон та дошка оголошень. Порівняльний аналіз каналів зведено у таблиці 1.2. Батьки найчастіше звертаються через Telegram-канал та телефон. Найпоширенішими запитами є умови вступу, розклад уроків та профорієнтація. За оцінками керівництва, обробка однотипних запитів займає близько 10% робочого часу секретаря та заступника директора.

Таблиця 1.2 – Аналіз каналів комунікації Гімназії №35

Канал комунікації	Аудиторія	Переваги	Недоліки
Вебсайт (Google Sites)	Всі	Доступ 24/7	Оновлюється невчасно
Telegram-канал	Батьки, учні	Оперативне сповіщення	Повідомлення губляться в чаті
Viber-групи класів	Батьки + кл. керівник	Адресна комунікація	Фрагментованість
Телефон (0532) 674117	Всі	Прямий контакт	Обмежений робочий час, навантаження на персонал
Facebook	Широка аудиторія	Охоплення нових аудиторій	Лише анонси заходів
Дошка оголошень	Відвідувачі закладу	Наочність	Лише при фіз. відвідуванні

Цільова аудиторія інформаційної системи поділяється на три групи зовнішніх користувачів, що показано на рисунку 1.2. Вона демонструє, що інформаційна система виступає єдиним комунікаційним центром між закладом освіти та всіма категоріями користувачів, забезпечуючи доступ до актуальної та структурованої інформації. Абітурієнти потребують інформації про умови зарахування та перелік документів. Учням потрібен розклад дзвінків, уроків та позакласних заходів. Батьки цікавляться безпекою, харчуванням та успішністю

учнів. Всі три групи взаємодіють з інформаційною системою (чат-ботом / вебплатформою), що надає актуальні відповіді в автоматичному режимі та зменшує навантаження на персонал.



Рисунок 1.2 – Цільова аудиторія гімназії

Детальний аналіз бізнес-процесів гімназії дозволив виявити низку системних недоліків у поточній моделі обслуговування відвідувачів вебсайту, що проілюстровано на рисунку 1.3. По-перше, користувачі страждають від відсутності персоналізації контенту та неможливості налаштувати систему Push-оповіщень для своєчасного інформування. По-друге, існуючі канали зворотного зв'язку є вкрай обмеженими та не задовольняють потребу у швидкій комунікації. По-третє, ефективність роботи з сайтом знижується через значний інформаційний шум та заплутану навігацію, яка суттєво ускладнює пошук потрібних даних.

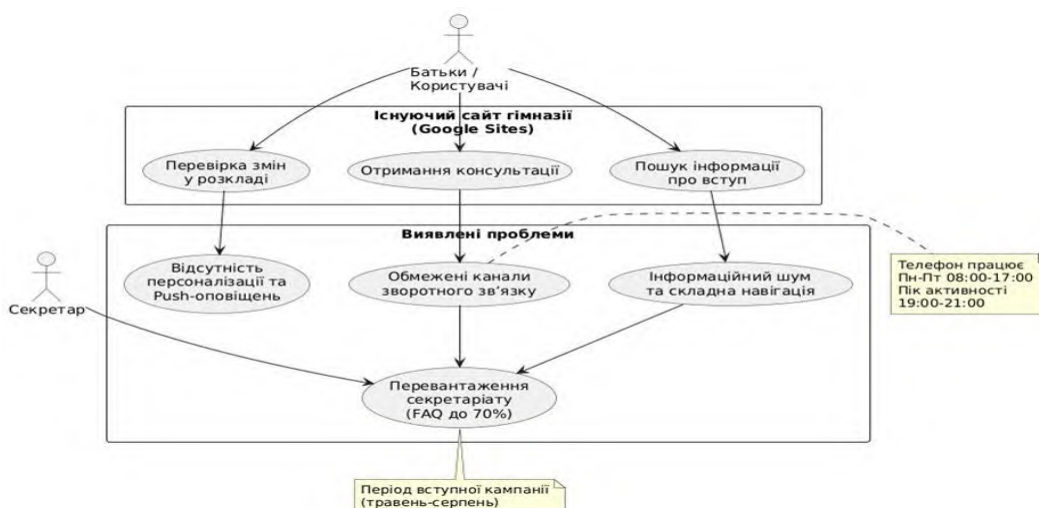


Рисунок 1.3 – Аналіз недоліків бізнес-процесів Гімназії №35

Усі три проблеми зводяться до одного наслідку, перевантаження секретаріату, оскільки FAQ запити займають до 70% часу, причому пік активності звернень з 19:00 до 21:00 повністю виходить за межі робочого дня. Ключовими недоліками, що підтверджені в ході аналізу каналів комунікації та бізнес-процесів гімназії, є те, що повідомлення губляться у переповнених чатах, сайт оновлюється несвоєчасно, а також відсутній єдиний цілодобовий інформаційний канал для зовнішніх запитів.

1.3 Програмно-технічне забезпечення закладу

Загальний парк комп'ютерної техніки гімназії налічує 37 одиниць, з яких 7 підлягають списанню, 4 є неробочими (ефективний парк становить близько 26 пристроїв). Адміністративні ноутбуки (Dynabook, Lenovo, Dell) відповідають сучасним вимогам. У навчальних кабінетах встановлено 8 мультимедійних комплексів. Гімназія підключена до Інтернету через оптоволоконний кабель зі швидкістю від 30 до 100 Мбіт/с. Документація зберігається на локальних дисках, і це ускладнює синхронізацію при формуванні бази знань чат-бота. Перелік прикладного програмного забезпечення наведено у таблиці 1.3.

Таблиця 1.3 – Прикладне програмне забезпечення Гімназії №35

Група	Назва ПЗ	Призначення	Недоліки
Офісні програми	Microsoft Office	Документообіг, звітність	Відсутня централізована БД документів
Управління навч. процесом	«Нові Знання», Курс: Школа, АІКОМ 1 і 2	Електронний журнал, автоматизація	Нестабільна синхронізація між системами
Держ. реєстр	Дія	Взаємодія з держ. послугами	Залежить від Інтернету
Бухгалтерія / ERP	М.Е.Дос, ЕІСУБ, ІS-Pro	Звітність, фінансовий облік	Складний інтерфейс, витрати на оновлення
Дистанційне навчання	Zoom	Відеоуроки	Залежність від якості Інтернету

Найбільш критичним недоліком, що впливає прикладного програмного забезпечення гімназії, є нестабільна синхронізація між системами «Курс: Школа» та «АІКОМ 1 і 2», що призводить до дублювання введення даних. Загалом ПЗ закладу охоплює всі необхідні задачі, проте є фрагментованим і не інтегрованим в єдину систему.

1.4 Огляд наукових публікацій та обґрунтування технологічного рішення для розробки чат-бота

Відповідно до Концепції «Нова українська школа» (Розпорядження КМУ № 988-р від 14.12.2016) одним із ключових напрямів реформи є формування інформаційно-цифрової компетентності та запровадження електронних форм комунікації між закладом, учнями та батьками [1]. На практиці більшість закладів загальної середньої освіти стикається з обмеженою оперативністю оновлення інформації та відсутністю цілодобових каналів зворотного зв'язку. Переваги й недоліки вебсайту гімназії систематизовано у таблиці 1.4.

Таблиця 1.4 – Аналіз переваг і недоліків вебсайту Гімназії №35

Переваги	Недоліки
Безкоштовне розміщення та хостинг від Google	Відповідальна особа має надмірне навчальне навантаження
Адаптивний дизайн (відображається на мобільних)	Інформація оновлюється невчасно
Вбудована функція пошуку (знаходить .pdf, .pptx)	Відсутній інтерактивний канал зворотного зв'язку 24/7
Проста структура, зрозуміла для користувачів	Неможливість відповідати на запити в реальному часі
Інтеграція з екосистемою Google (Drive, Docs)	Відсутня можливість динамічних компонентів без сторонніх сервісів

За даними систематичного огляду 485 наукових публікацій з баз Web of Science та Scopus [2], більшість досліджень з теми освітніх чат-ботів зосереджена на вищій освіті та вивченні іноземних мов. Застосування чат-ботів у закладах

загальної середньої освіти автори прямо визначають як молодосліджений напрям та пріоритет для майбутніх досліджень. Кількість публікацій стрімко зростає: з 19 у 2021 році до 200 у 2024 році. Методологія відбору публікацій за PRISMA наведена на рисунку 1.4, а динаміка зростання на рисунку 1.5.

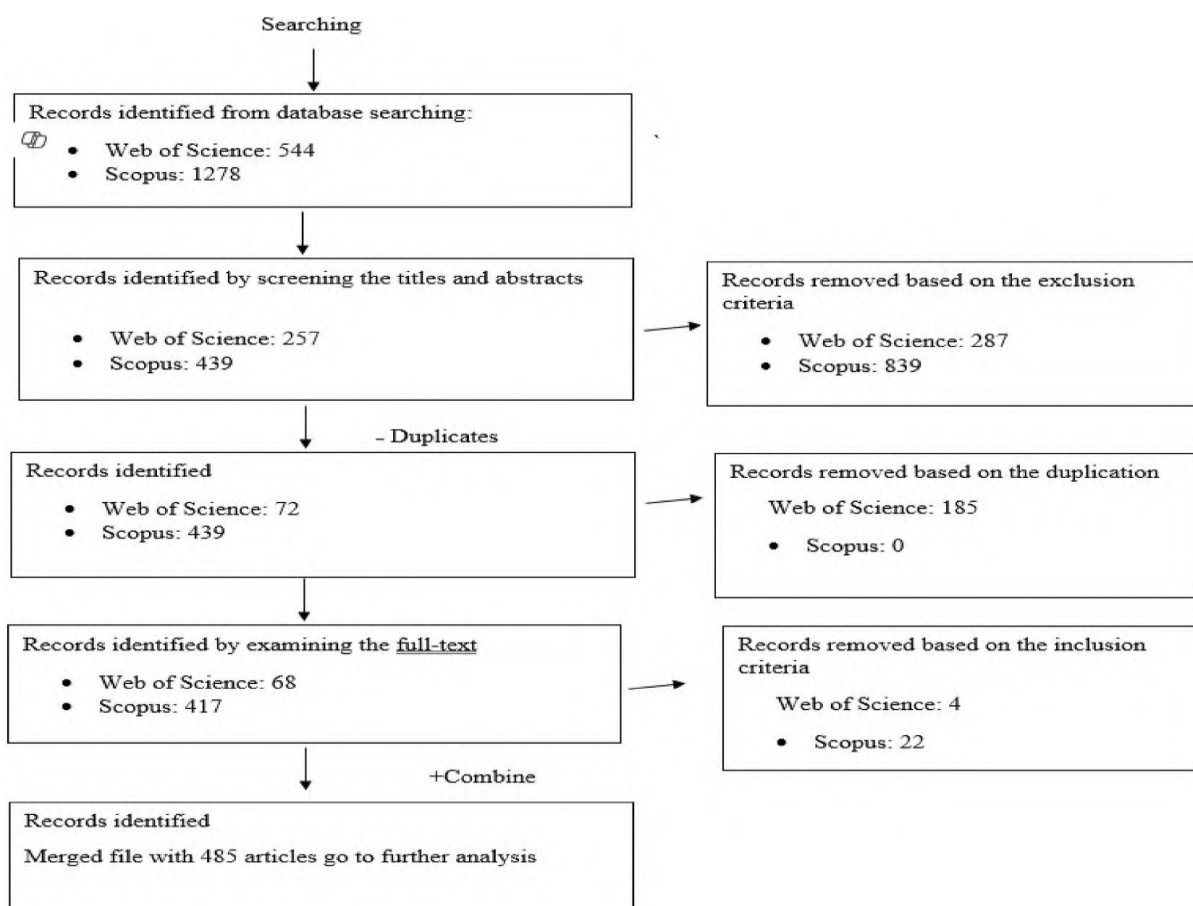


Рисунок 1.4 – Блок-схема відбору публікацій за методологією PRISMA

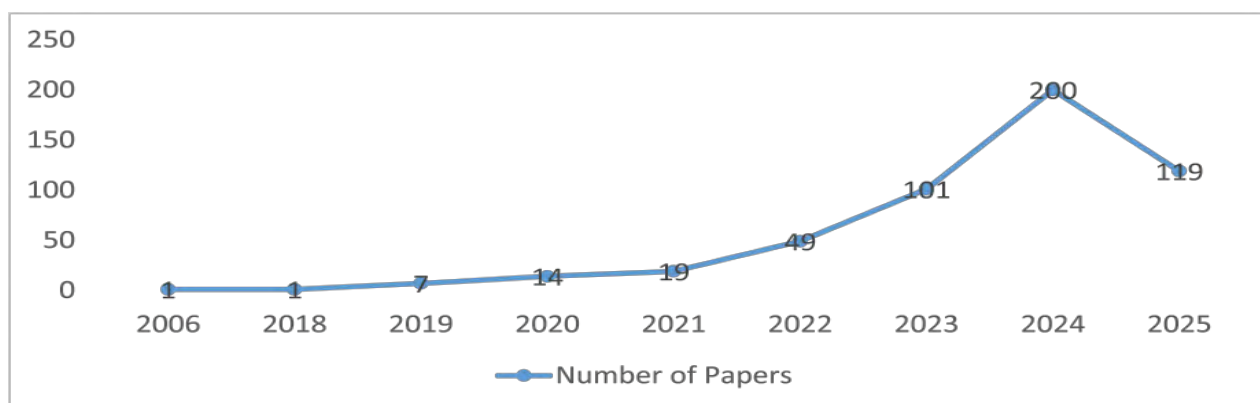


Рисунок 1.5 – Динаміка зростання кількості наукових публікацій з теми освітніх чат-ботів у 2006–2025 роках

Дослідники Д. Гальвонік та Й. Капуста з Університету Константина Філософа (Словаччина) провели шестимісячний порівняльний експеримент між детермінованою моделлю (M1) та моделлю GPT (M2) [3]. Етапи експерименту наведено на рисунку 1.6, а результати метрик BLEU на рисунку 1.7.

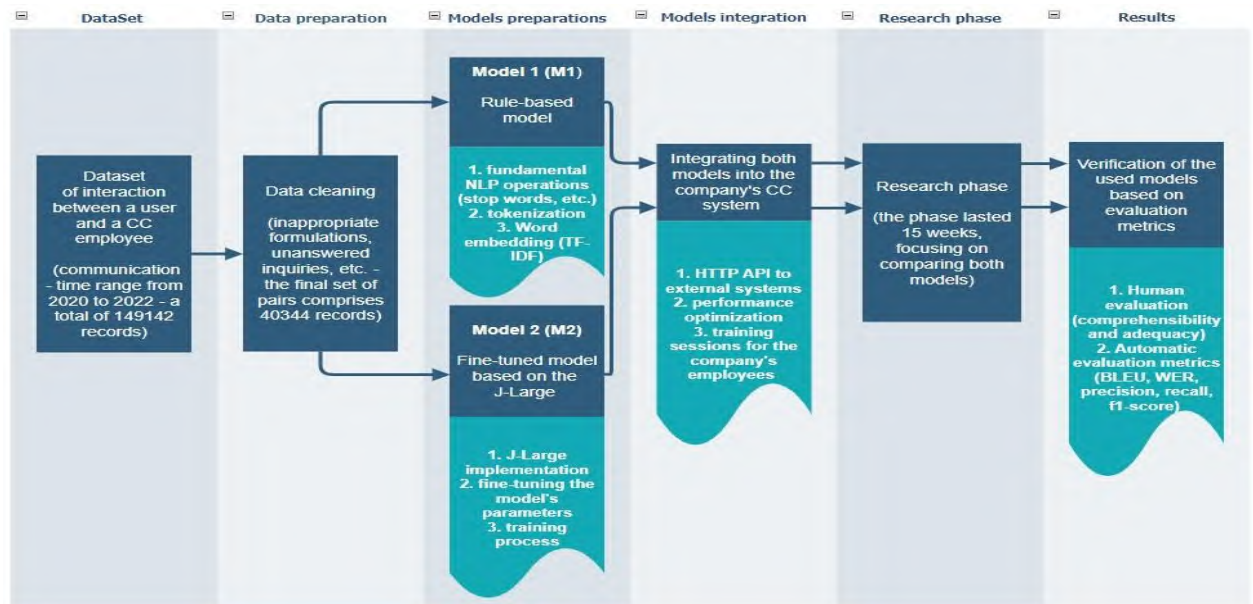


Рисунок 1.6 – Кроки експерименту з порівняння GPT-моделі (M2) та правило-орієнтованої моделі (M1)

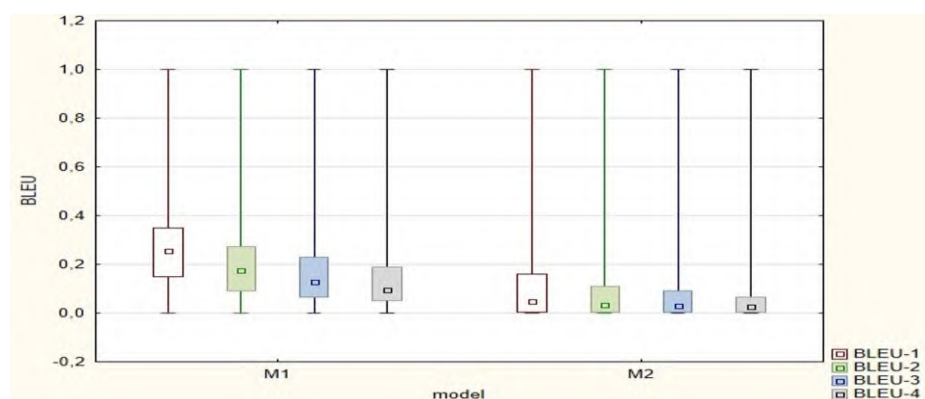


Рисунок 1.7 – BLEU-метрики порівняння детермінованої та GPT-моделі

Детерміновану модель обирали у 36% випадків, тоді як GPT лише у 7%. Навчання моделей GPT вимагає значних ресурсів, що робить їх малопридатними для освітніх закладів. Це підтверджує обґрунтованість детермінованого підходу для чат-бота гімназії.

Д-р С. А. Тхорат та В. Д. Джадхав (Технологічний інститут ім. Раджарамбапу, Індія) реалізували чат-бот для вебсайту навчального закладу на платформах Google Dialogflow та IBM Watson [4], що показано на рисунку 1.8. Google Dialogflow відзначається кращою документацією та зручнішим процесом конфігурації. Саме цей кейс є найближчим до задачі, що вирішується в межах поточної кваліфікаційної роботи.



Рисунок 1.8 – Робота чат-бота на платформах Dialogflow та IBM Watson

У першому розділі проаналізовано навчальний заклад Гімназії №35 Полтавської міської ради як об'єкт цифровізації. Аналіз показав, що секретаріат закладу критично перевантажений повторюваними запитами від батьків та абітурієнтів. Попри достатнє технічне забезпечення для хмарних рішень, наявне ПЗ фрагментоване, а вебсайт не має інтерактивного зворотного зв'язку. Огляд наукових публікацій засвідчив стрімке зростання інтересу до освітніх чат-ботів. Отже, заклад освіти є типовим, але складним об'єктом автоматизації. Впровадження чат-бота переведе його комунікацію на новий рівень, забезпечивши всім цілодобовий доступ до інформації.

РОЗДІЛ 2

СТВОРЕННЯ ІНФОРМАЦІЙНОЇ БАЗИ ТА РОЗРОБКА ЧАТ-БОТА

2.1 Обґрунтування вибору платформи та технологій розробки

Розвиток сучасних інформаційних технологій та глобальна цифровізація освітнього простору призвели до виникнення нових форм взаємодії між закладами освіти та користувачами. Одним із найбільш перспективних напрямів є впровадження інтелектуальних діалогових агентів, відомих як чат-боти.

Чат-бот [5] – це комп'ютерна програма, призначена для імітації письмової або розмовної людської розмови під час спілкування з користувачами, щоб допомогти відповісти на запитання або вирішити проблеми. Чат-ботів можна знайти на вебсайтах, додатках, соціальних мережах та розумних пристроях. Вони виконують такі завдання, як підтримка клієнтів, навігація програмного забезпечення та особиста допомога, наприклад, запам'ятовування списків покупок або надсилання сповіщень і нагадувань.

З метою створення чат-боту потрібно визначитися з вибором платформи для його розробки та середовища розгортання. Для обґрунтованого вибору необхідно проаналізувати сучасний ринок мобільних месенджерів. Згідно з даними дослідницького порталу Statista, представленими на рисунку 2.1, станом на жовтень 2025 року світовий ринок мобільних додатків для обміну повідомленнями демонструє стабільне зростання та високий рівень залученості користувачів у різних регіонах світу. Абсолютним лідером є WhatsApp із показником у 3 мільярди щомісячних активних користувачів. Друге місце посідає китайський WeChat (1,41 млрд). Месенджер Telegram закріпився на третій позиції, досягнувши позначки в 1 мільярд активних користувачів щомісяця, випередивши Facebook Messenger, аудиторія якого становить 942 мільйони [6]. Таким чином, сучасний ринок месенджерів характеризується високою концентрацією користувачів у кількох глобальних платформах, що формують цифрову інфраструктуру комунікації у світовому масштабі.

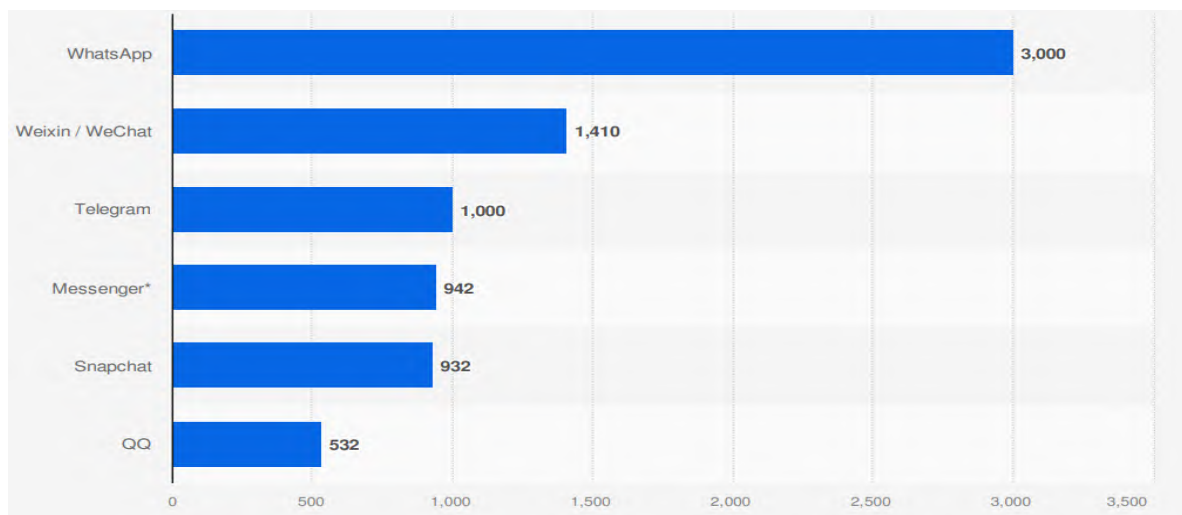


Рисунок 2.1 – Статистика популярних месенджерів у світі згідно з даними дослідницького порталу Statista (на жовтень 2025 року)

Враховуючи цю статистику, а також специфіку цільової аудиторії закладу середньої освіти, доцільно провести глибокий технічний аналіз платформ WhatsApp, Facebook Messenger, Viber та Telegram на основі їхньої офіційної документації.

Аналіз офіційної документації WhatsApp Cloud API [7] свідчить, що платформа побудована на базі Graph API та використовує протокол HTTP. Основною технічною та адміністративною перепорою для використання цієї платформи навчальним закладом є жорстка прив'язка до бізнес-інфраструктури: розробнику необхідно створити бізнес-портфоліо та спеціальний аккаунт WhatsApp Business. API також передбачає збір аналітики розцінок із детальною розбивкою по доставлених повідомленнях, що підтверджує комерційний характер використання інструменту. Приклад архітектури взаємодії через WhatsApp Business API наведено на рисунку 2.2. Ліва частина рисунка ілюструє базову модель звичайної програми для обміну повідомленнями. У цьому випадку реалізована проста схема спілкування, де кінцеві користувачі (фізичні особи) взаємодіють один з одним безпосередньо через мобільні застосунки. Базовий сервер у цій моделі виконує виключно роль закритого транзитного вузла, який приймає та пересилає повідомлення між пристроями без можливості підключення зовнішніх програмних систем. Права частина схеми деталізує

архітектуру корпоративного програмного інтерфейсу, яка є багаторівневою та орієнтованою на системну інтеграцію. Головною відмінністю є поява проміжного вузла у вигляді сервера корпоративної інтеграції. Цей компонент виступає в ролі шлюзу, який з'єднує базову інфраструктуру служби повідомлень із внутрішніми інформаційними системами організації.

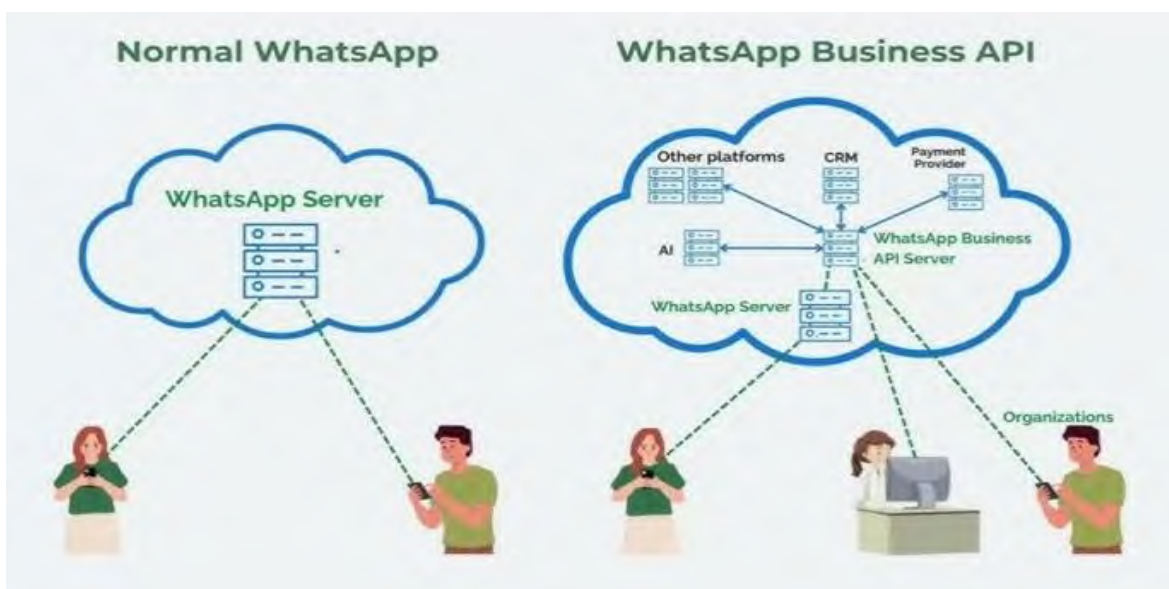


Рисунок 2.2 – Схема обміну повідомленнями через WhatsApp Business API

Інфраструктура Facebook Messenger (Meta Messenger Platform) [8] надає розробникам потужний інструментарій для створення діалогових рішень. Сучасна архітектура платформи об'єднує в собі рішення «Messenger для бізнесу» та «Обмін повідомленнями в Instagram», що дозволяє охопити ширшу аудиторію. Серед ключових технічних переваг API розробники виділяють: вбудовану обробку природної мови (NLP) для аналізу запитів, підтримку вебперегляду для створення вбудованих вебботів, а також розширені інструменти аналітики для відстеження результативності. Незважаючи на оновлення API у липні 2024 року, яке дещо спростило роботу з професійними акаунтами Instagram без жорсткої прив'язки до сторінки Facebook. Візуалізацію функціональних можливостей наведено на рисунку 2.3. Проте, незважаючи на багатий функціонал та відсутність плати за базові вхідні повідомлення, ключовою перепорою для використання цієї платформи у комунальному закладі освіти є екосистемна

залежність. Інформаційна взаємодія вимагає наявності у кінцевого користувача акаунта у соціальних мережах компанії Meta. Враховуючи, що значна частина цільової аудиторії гімназії (зокрема здобувачі освіти 1–9 класів) не є активними користувачами Facebook, розгортання інформаційної системи на базі Messenger призведе до штучного обмеження доступу до публічної інформації, що є неприпустимим для державного закладу.

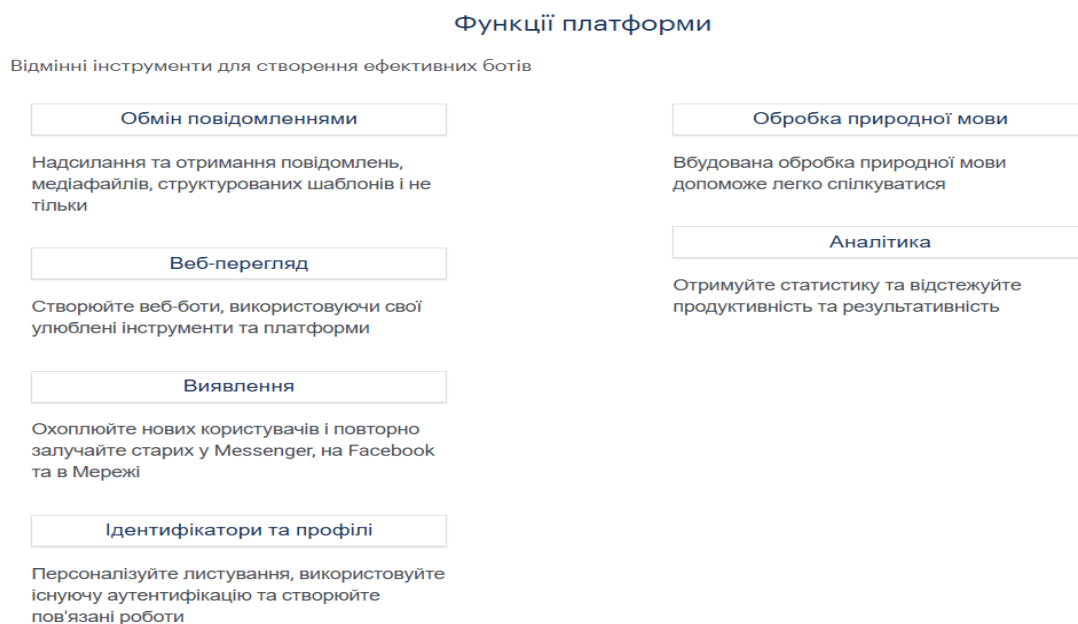


Рисунок 2.3 – Функціональні можливості платформи Facebook Messenger

Розглядаючи платформу Viber [9], варто зазначити, що її REST API вимагає налаштування вебхука з обов'язковою наявністю офіційного SSL-сертифіката від довіреного центру сертифікації. Вирішальним фактором відмови від цієї платформи є зміна її політики: згідно з офіційною документацією розробника, починаючи з 5 лютого 2024 року, боти у Viber можуть бути створені виключно на комерційних умовах. Архітектура API безпосередньо містить параметр `billing_status`, який визначає тарифікацію повідомлень, зокрема стягнення плати за повідомлення поза активною сесією. Генерування таких витрат є неприпустимим для державного закладу освіти. Структуру типового повідомлення у Viber подано на рисунку 2.4.

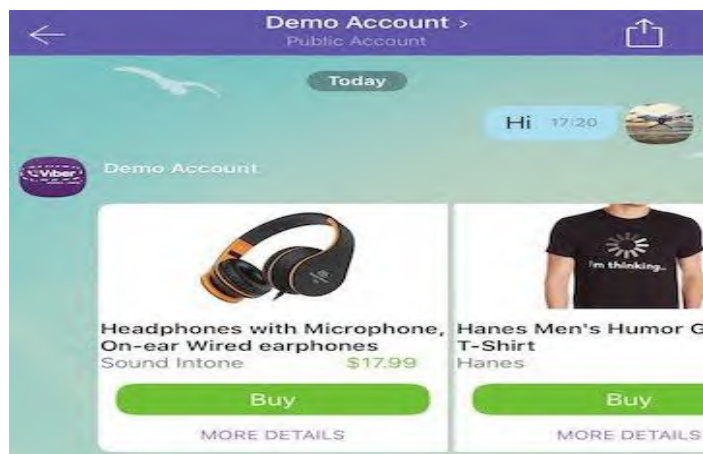


Рисунок 2.4 – Візуалізація формату Rich Media у Viber REST API

На противагу вищезазначеним рішенням, інфраструктура Telegram [10] та її офіційний Bot API пропонують кардинально інший, відкритий підхід до розробки. Згідно з офіційною документацією, Telegram Bot API є спеціалізованим HTTP-інтерфейсом, створеним для розробників діалогових програм. Усі запити до API обслуговуються виключно через безпечний протокол HTTPS і підтримують як GET, так і POST HTTP-методи передачі даних. Обмін інформацією відбувається за допомогою стандартизованих JSON-об'єктів, що робить інтеграцію максимально універсальною. Архітектурну схему взаємодії клієнта, сервера Telegram та бекенду розробника наведено на рисунку 2.5, що розглядається у публікації розробки чат-боту Т. Владика [11].

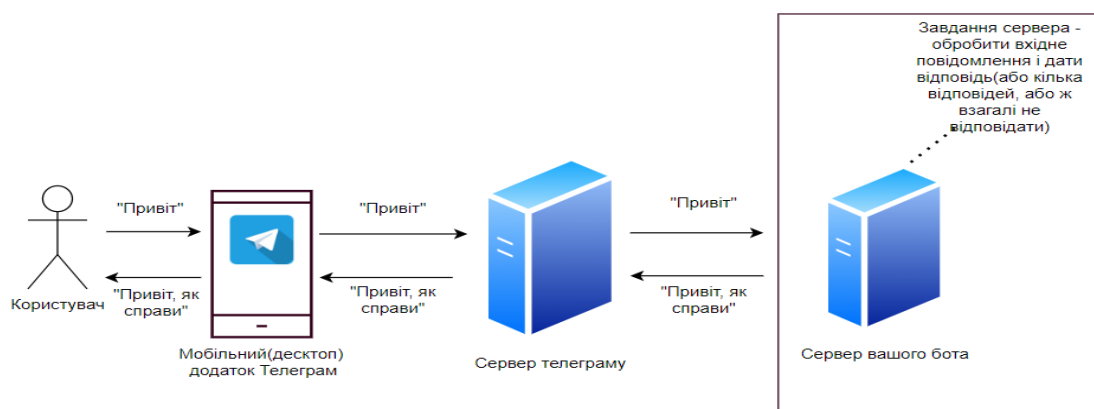


Рисунок 2.5 – Візуалізація архітектурної схеми взаємодії клієнта, сервера Telegram та бекенду розробника

Процес ініціалізації бота є максимально спрощеним: він створюється через внутрішнього системного бота @BotFather, який генерує унікальний токен авторизації. На відміну від WhatsApp Business, боти в Telegram є спеціальними незалежними акаунтами, для налаштування яких не потрібен номер телефону чи складні процедури верифікації юридичної особи. Важливою архітектурною особливістю є можливість розгортання локального сервера, що дозволяє зняти обмеження на розмір файлів та використовувати будь-які локальні IP-адреси для вебхуків. Крім класичного текстового інтерфейсу з використанням вкладених Inline- та Reply-клавіатур, сучасна інфраструктура Telegram підтримує технологію Mini Apps. Це дозволяє розробникам створювати гнучкі графічні інтерфейси, які працюють безпосередньо всередині месенджера і здатні повноцінно замінити традиційні вебсайти. Головною перевагою платформи для Гімназії №35 є її абсолютна безкоштовність та відсутність комерційних обмежень. Боти можуть бути легко інтегровані в існуючі канали чи групи навчального закладу. Відсутність лімітів на кількість вхідних чи вихідних повідомлень робить Telegram ідеальним, економічно виправданим та технологічно досконалим середовищем для розгортання шкільної інформаційної системи. Детальні результати аналізу офіційних документацій за ключовими критеріями представлено у вигляді порівняльної таблиці в Додатку Б.

На основі поглибленого системного аналізу офіційних специфікацій та технічної документації провідних месенджерів можна зробити обґрунтований висновок щодо вибору оптимального середовища розгортання інформаційної системи. API платформ WhatsApp та Viber жорстко орієнтовані на комерційний сектор послуг. Вони передбачають складні процедури верифікації бізнес-акаунтів та генерують постійні фінансові витрати на маршрутизацію повідомлень, використання шаблонів чи підтримку активних діалогових сесій. Для комунального закладу загальної середньої освіти, який фінансується за рахунок бюджетних коштів, інтеграція рішень із платною інфраструктурою є економічно недоцільною та нежиттєздатною у довгостроковій перспективі.

За результатами комплексного порівняння, Telegram Bot API виявився єдиним рішенням, яке повністю задовольняє як жорсткі технічні, так і соціально-економічні вимоги Гімназії №35. В умовах фінансування комунального закладу освіти критично важливим є використання платформи, що забезпечує абсолютно безкоштовну, відкриту та незалежну від обов'язкової прив'язки до мобільного номера чи профілю у соціальних мережах архітектуру. Це гарантує рівний і безперешкодний доступ до інформації для всіх учасників освітнього процесу. З технічної точки зору, наявність потужного інструментарію для візуальної кастомізації інтерфейсу дозволяє надзвичайно ефективно структурувати та сегментувати великі масиви даних. Це ідеально підходить для дистрибуції інформації щодо розкладу навчального процесу, алгоритмів зарахування під час вступної кампанії, публічної звітності та питань безпеки. Завдяки такій логічній структурованості безпосередньо вирішується ключова проблема «інформаційного шуму» та складної навігації, яка була виявлена на етапі системного аналізу предметної області. Таким чином, зважаючи на стабільно високий рівень популярності месенджера серед усіх вікових категорій населення України, простоту розгортання та повну відсутність прихованих комерційних обмежень чи тарифікації повідомлень, Telegram визначено як безальтернативний та технологічно найдосконаліший інструмент.

2.2 Дослідження підходів до побудови архітектури чат-ботів

Після затвердження Telegram як базової комунікаційної платформи, наступним етапом проєктування інформаційної системи є визначення її внутрішньої логіки та архітектурного підходу.

Сучасна інженерія програмного забезпечення виділяє два фундаментальні підходи до побудови архітектури діалогових агентів:

- підхід на основі машинного навчання та обробки природної мови (AI/ML);

– детермінований підхід на основі набору правил і дерев рішень (Rule-based).

Сучасним напрямом розробки інтелектуальних діалогових систем є використання методів штучного інтелекту та машинного навчання (Machine Learning). Для глибокого розуміння архітектури таких систем і оцінки доцільності їх впровадження в інформаційному середовищі освітнього закладу необхідно детально розглянути декілька фундаментальних технологій: Natural Language Processing (NLP), Natural Language Understanding (NLU) та Artificial Neural Networks (ANN).

Обробка природної мови є базовою та первинною технологією для створення сучасних діалогових інтерфейсів. Згідно з матеріалами та дослідженнями, проведеними у межах європейського проєкту FITPED-AI [12], Natural Language Processing (NLP) визначається як надзвичайно складна міждисциплінарна галузь, що знаходиться на стику комп'ютерних наук, методів штучного інтелекту та прикладної лінгвістики. Її головною та найбільш амбітною метою є забезпечення безперешкодної, природної взаємодії між обчислювальними машинами та людською мовою. Дослідники наголошують, що ця технологія покликана подолати бар'єр між машинним кодом та людським спілкуванням, надаючи комп'ютерним системам здатність читати, глибоко аналізувати та самостійно генерувати текстові дані таким чином, щоб максимально імітувати людське розуміння. У контексті розробки комунікаційних чат-ботів NLP виступає найпершим рівнем обробки неструктурованої вхідної інформації. Головне завдання цієї технології полягає у тому, щоб перетворити вільний текст, введений користувачем (який для звичайної програми є просто неструктурованим набором символів), у структурований, машинозчитуваний формат, підготувавши масив даних для подальшого програмного розбору алгоритмами системи. Для наочного розуміння структурної ролі цієї технології варто звернутися до наукового дослідження С. Сінгха [13], де детально розглядається архітектурне місце NLP у сучасних людино-машинних інтерфейсах. На рисунку 2.6 відображено основні

етапи цього процесу: від введення запиту користувачем до інтерпретації змісту та передачі даних до відповідних інтелектуальних підсистем.



Рисунок 2.6 – Структурна діаграма системи обробки природної мови в архітектурі людино-машинного інтерфейсу

Наступним, значно складнішим етапом багаторівневої логічної обробки текстової інформації є розуміння природної мови NLU. Як докладно зазначають вітчизняні дослідники С. Переяславська та О. Смагіна [14], класичний конвеєр роботи таких систем включає кілька послідовних кроків: токенизацію (процес розбиття суцільного тексту на окремі значеннєві одиниці – токени або слова), синтаксичний аналіз (перевірку граматичної структури речення) та безпосередньо семантичний аналіз. NLU виступає як інтелектуальна підсистема, яка відповідає не просто за структурування, а безпосередньо за вилучення контекстуального сенсу з повідомлення. Її ключовим завданням у функціонуванні чат-ботів є розпізнавання прихованих намірів (інтентів) користувача та точне вилучення конкретних змінних сутностей з тексту. Завдяки цьому чат-бот здатен відфільтрувати синтаксичний «шум» та виокремити головну мету звернення. Проте впровадження алгоритмів розуміння природної мови для потреб державного закладу освіти має критичний недолік. Як ґрунтовно пояснюється у навчально-методичному посібнику В. В. Троцька [15],

інтелектуальні системи у процесі розпізнавання сенсу та прийняття рішень спираються на евристичні алгоритми. Це означає, що система працює в умовах математичної невизначеності та неповної інформації, генеруючи не абсолютно точну відповідь, а лише найбільш імовірну. Це створює перманентний ризик того, що система внаслідок збою у семантичному аналізі неправильно класифікує намір користувача і згенерує хибну або неточну відповідь (так звану «галюцинацію»). Для гімназії видача некоректних даних (наприклад, помилкового розкладу занять, хибних строків подачі заяв чи неправильних алгоритмів дій під час повітряної тривоги) є абсолютно неприпустимою та порушує регламент закладу. Для глибшого розуміння внутрішньої архітектури підсистеми NLU доцільно звернутися до досліджень С. Даблдея, С. Тротта та Дж. Фельдмана [16]. У своїй науковій праці автори наочно демонструють конвеєр системи розуміння природної мови, який розроблено для перетворення лінгвістичних конструкцій користувача у виконувані семантичні схеми (дії системи). Як показано на структурній діаграмі на рисунку 2.7, система NLU приймає попередньо оброблений текст і через механізми глибокого семантичного аналізу формує конкретні системні виклики та параметризовані дії. Цей принцип лежить в основі логіки будь-якого сучасного інтелектуального діалогового агента.

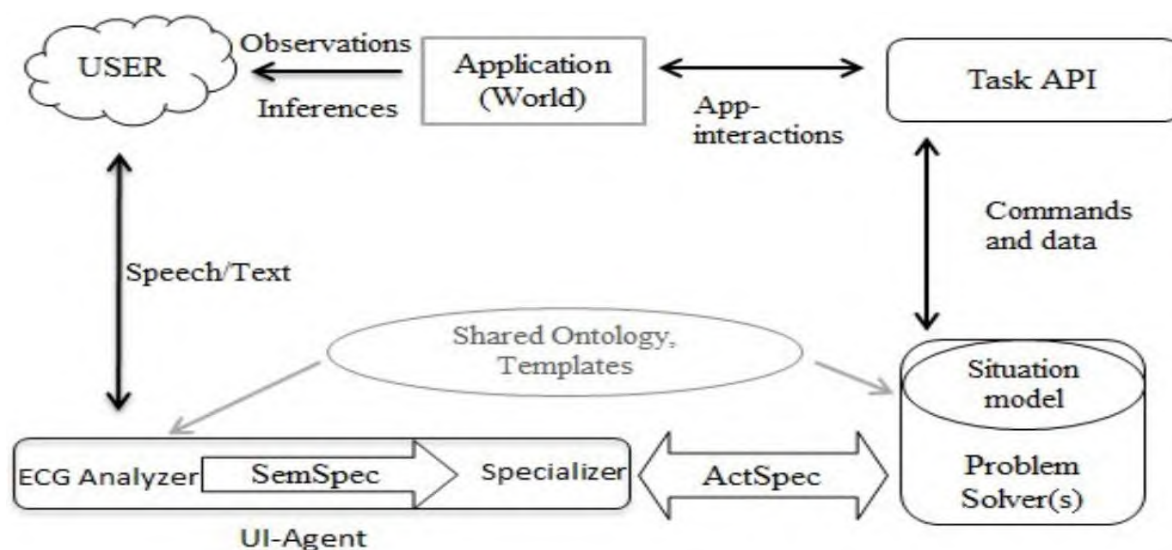


Рисунок 2.7 – Архітектура підсистеми розуміння природної мови (NLU)

Технологічним, архітектурним та математичним базисом, що фізично забезпечує функціонування вищезазначених систем глибокого аналізу мови, є Штучні нейронні мережі (Artificial Neural Networks). Відповідно до ґрунтовних досліджень Р. Камара та Б. А. Зардарі [17], архітектура та базові принципи роботи штучних нейронних мереж були безпосередньо натхненні будовою біологічного мозку людини. Автори класифікують сучасні нейронні мережі як масивно-паралельні обчислювальні системи, що складаються з величезної кількості взаємопов'язаних базових процесорів – штучних нейронів. Ці обчислювальні елементи математично реплікують спосіб комунікації та безперервного обміну сигналами реальних біологічних клітин.

Для кращого розуміння структурної організації цих обчислювальних систем доцільно звернутися до оглядової наукової праці Т. Л. Макосо, А. Алмактуфа та К. Або-Аль-Еза [18]. У своєму дослідженні автори зазначають, що архітектура нейронної мережі формується шляхом об'єднання нейронів у специфічні шари математичної обробки (вхідні, приховані та вихідні), які у сукупності забезпечують максимально точне реагування на вхідні дані. Наочну візуалізацію топології такої мережі, де штучні нейрони згруповані у взаємопов'язані шари передачі сигналів, наведено на рисунку 2.8.

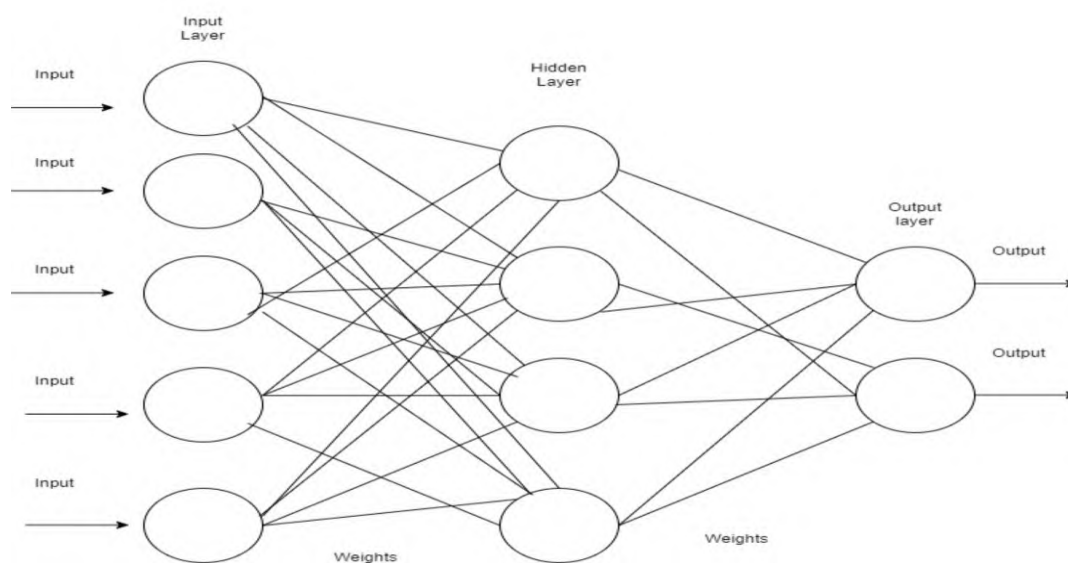


Рисунок 2.8 – Топологія та взаємодія шарів в архітектурі штучної нейронної мережі

Незважаючи на величезну обчислювальну потужність цієї технології, процес розгортання та забезпечення стабільної працездатності ANN є надзвичайно складним і ресурсоємним завданням, що робить його абсолютно непридатним для інтеграції у шкільне середовище. Згідно з науковими працями С. О. Субботіна [19], процес навчання штучних нейронних мереж вимагає складного ітеративного налаштування вагових коефіцієнтів синаптичних зв'язків. Цей процес може бути успішним виключно за умови використання величезних масивів розмічених історичних даних, які називаються навчальними вибірками (датасетами). Навчальний заклад освіти об'єктивно не володіє необхідним обсягом десятків тисяч текстових діалогів для повноцінного навчання власної локальної моделі. Крім того, залучення значних обчислювальних потужностей для постійної підтримки роботи масивно-паралельних мереж генеруватиме постійні операційні фінансові витрати, що повністю суперечить фінансовим можливостям та кошторису бюджетної установи.

Альтернативою нейромережевим алгоритмам є детермінована архітектура Rule-based, яка функціонує за принципом дерева рішень. Згідно з дослідженнями Е. Соломон та С. Л. Тілахуна [20], чат-боти на основі правил є значно простішими в програмній реалізації та генерують мінімальну кількість системних і логічних помилок. На відміну від моделей штучного інтелекту, такі системи не намагаються самостійно згенерувати текст, а видають користувачу чіткі, заздалегідь підготовлені відповіді на основі жорстко заданих алгоритмів. Як зазначається у профільних матеріалах з розробки діалогових інтерфейсів, Rule-based системи базуються на структурованих діалогових деревах. Замість вільного введення тексту, яке вимагає складного NLP-аналізу, розробники використовують так звані інтерактивні кнопки. Це обмежує спілкування чітко визначеною темою, але робить навігацію максимально передбачуваною та зрозумілою для кінцевого користувача.

Згідно з матеріалами, наведеними експертом Ясмін Альтман [21], принцип роботи Rule-based чат-бота представлено у вигляді структурної блок-схеми на

рисунку 2.9. Процес розробки такого бота зводиться до визначення завдань, збору необхідних даних, проєктування блок-схем маршрутів бесіди та їхнього перекладу у програмний код.

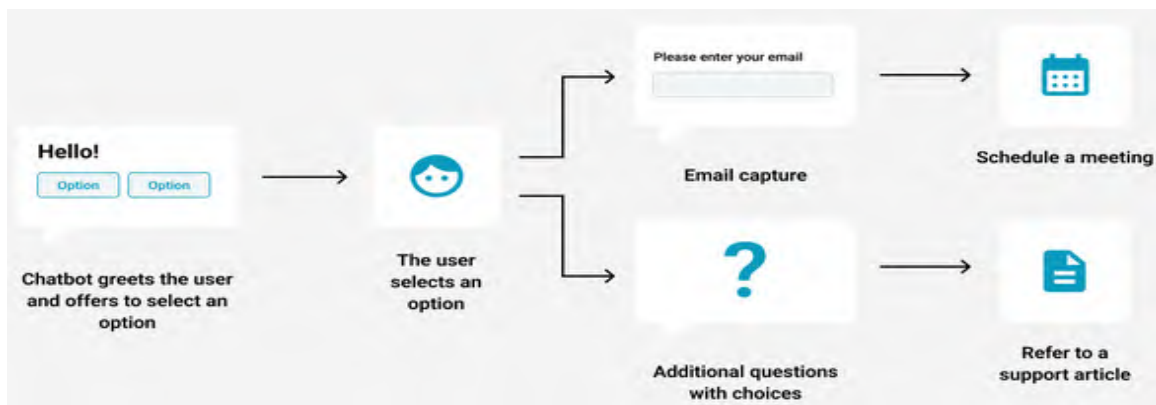


Рисунок 2.9 – Структурна блок-схема роботи детермінованого Rule-based чат-бота

Для реалізації інформаційної системи Гімназії №35 детермінований підхід на основі правил є безальтернативно найкращим вибором. У Додатку В наведено порівняльну таблицю характеристик Rule-based та AI/ML чат-ботів, розроблену на основі аналізу експерта Роберта Вебера [22]. Згідно з цими даними, детермінований підхід на основі правил є найкращим вибором для створення інформаційної системи Гімназії №35.

Як видно з наведеного аналізу, використання штучного інтелекту є виправданим лише для компаній, де необхідне ведення складних неструктурованих бесід. Для комунального закладу загальної середньої освіти ключовою вимогою є надання абсолютно достовірної, верифікованої та регламентованої інформації (наприклад, алгоритмів зарахування до 1-го класу, розкладу дзвінків чи правил поведінки в укритті). Вона не потребує збору величезних баз даних для навчання моделі та не генерує жодних фінансових витрат на підтримку хмарних обчислень. Крім того, жорстко структуроване кнопкове меню безпосередньо вирішує головну проблему гімназії – «інформаційний шум», дозволяючи батькам та учням знаходити потрібні документи у 2-3 кліки, що суттєво розвантажує адміністративний персонал.

Проведений комплексний аналіз доводить, що використання штучного інтелекту для чат-бота школи є технологічно та економічно надлишковим. Оптимальним, безпечним та найбільш ефективним вибором є детермінована архітектура на основі правил (Rule-based). Вона гарантує 100% точність наданих даних, простоту розгортання та ідеально поєднується з обраною раніше платформою Telegram Bot API.

2.3 Порівняльний аналіз інструментальних засобів розробки та вибір платформи

Після визначення Rule-based підходу як оптимальної архітектури для інформаційної системи закладу освіти, наступним кроком є вибір безпосереднього інструментарію для її реалізації. Створити чат-бота можна двома принципово різними способами: написати програмний код власноруч або зібрати логіку в спеціалізованому візуальному конструкторі.

Згідно з ґрунтовним дослідженням Р. Хана та А. Даса [23] розробка діалогових систем концептуально поділяється на використання програмних фреймворків та готових візуальних платформ. Аналізуючи перший підхід, автори наголошують, що для створення бота програмним шляхом розробник повинен на високому рівні володіти мовами програмування (наприклад, JavaScript), вміти працювати з пакетними менеджерами (зокрема, Node Package Manager), налаштовувати серверні середовища на базі операційних систем Linux або Macintosh, а також оперувати командним рядком і власноруч проєктувати бази даних. Такий метод є вкрай ресурсоємним, вимагає значних часових витрат та залучення висококваліфікованих ІТ-спеціалістів. Альтернативним підходом є використання візуальних конструкторів. Їхня головна перевага полягає у тому, що весь складний програмний код повністю абстрагується під графічним інтерфейсом. Замість написання скриптів розробник використовує готові графічні блоки для побудови логіки повідомлень. Це робить процес розробки

інтуїтивно зрозумілим і доступним для користувачів без профільної технічної підготовки, що особливо актуально для бюджетних установ. Враховуючи специфіку комунального закладу освіти та відсутність у штаті профільних програмістів, використання візуального конструктора є єдиним доцільним рішенням.

Для обґрунтованого вибору необхідно провести порівняльний аналіз існуючих на ринку платформ-конструкторів: UChat, Gerabot, Flow XO та SendPulse. Порівняння проводитиметься за трьома ключовими параметрами: гнучкість налаштування логічних ланцюжків, можливості інтеграції із зовнішніми базами даних та зручність інтерфейсу.

Розглянемо платформу UChat [24]. Згідно з офіційною документацією, цей сервіс позиціонується як багатофункціональне та масштабоване рішення для побудови інтелектуальних діалогових інтерфейсів із повноцінною підтримкою омніканальності. Слід зазначити, що омніканальність [25] – це програмне забезпечення, створене для того, щоб на одній інформаційній панелі користувач міг взаємодіяти з різними цифровими каналами, тож ним легко керувати без необхідності відвідувати канали один за іншим. Ця архітектурна особливість дозволяє розробникам створювати єдиного універсального чат-бота з об'єднаною базою даних, який здатен одночасно та синхронно функціонувати у 12 різних комунікаційних каналах, серед яких Telegram, Facebook Messenger, WhatsApp, Viber, SMS, голосові мережі, а також корпоративні платформи на зразок Slack чи Intercom. Головною технологічною складовою та ядром системи є інтуїтивно зрозумілий візуальний конструктор діалогів. Цей інструмент надає можливість розробнику візуально формувати надзвичайно складні маршрути та блок-схеми, де кожен вузол (текстове повідомлення, логічна умова, затримка часу або дія) розташовується на єдиному робочому полотні. Такий підхід ідеально відповідає парадигмі розробки Rule-based архітектури, дозволяючи фізично проєктувати багаторівневі дерева рішень без необхідності написання складного програмного коду. Візуальне подання робочого середовища конструктора потоків наведено на рисунку 2.10.

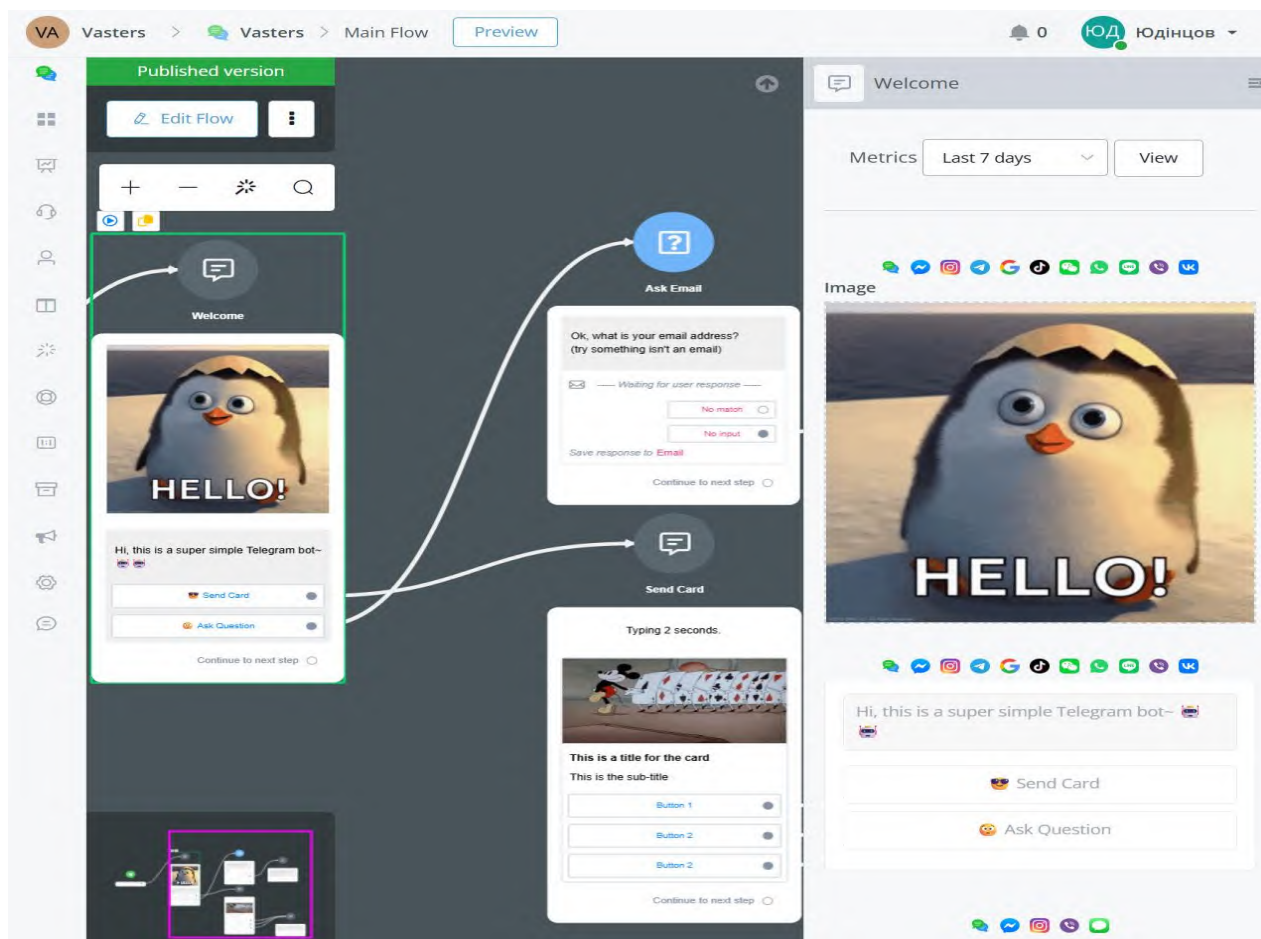


Рисунок 2.10 – Інтерфейс візуального конструктора діалогових потоків платформи UChat

Окрім розширеного конструювання логіки, UChat пропонує потужні можливості злиття із зовнішніми інформаційними системами. Це дозволяє діалоговим потокам динамічно обмінюватися масивами даних зі сторонніми API, базами даних та хмарними таблицями в режимі реального часу. Важливим адміністративним модулем платформи є Centralize Inbox & Live Chat, який агрегує всі повідомлення від користувачів з усіх підключених каналів у єдину панель керування. Система управління платформою організована за принципом ізольованих робочих просторів, що дозволяє гнучко налаштовувати права доступу для різних членів команди. Завдяки наявності мобільних додатків та модулю Live Chat, адміністратори мають змогу відстежувати історію взаємодій та за необхідності миттєво і безшовно перемикає діалог з автоматизованого бота на живого оператора. Додатково платформа містить вбудовані модулі

електронної комерції та інструменти для створення голосових алгоритмів, що робить її універсальним середовищем для проєктів будь-якого рівня складності.

Наступним інструментом, що підлягає аналізу, є вітчизняна розробка Gerabot [26]. Згідно з офіційною технічною документацією, ця платформа позиціонується насамперед як комплексне рішення для вебсайтів, спрямоване на збільшення конверсії, лідогенерацію (збір контактів) та оптимізацію бізнес-процесів компанії. Процес розгортання системи передбачає генерацію спеціального коду віджета та його встановлення на сторінки сайту, візуальне подання інтерфейсу віджета та панелі керування наведено на рисунку 2.11.

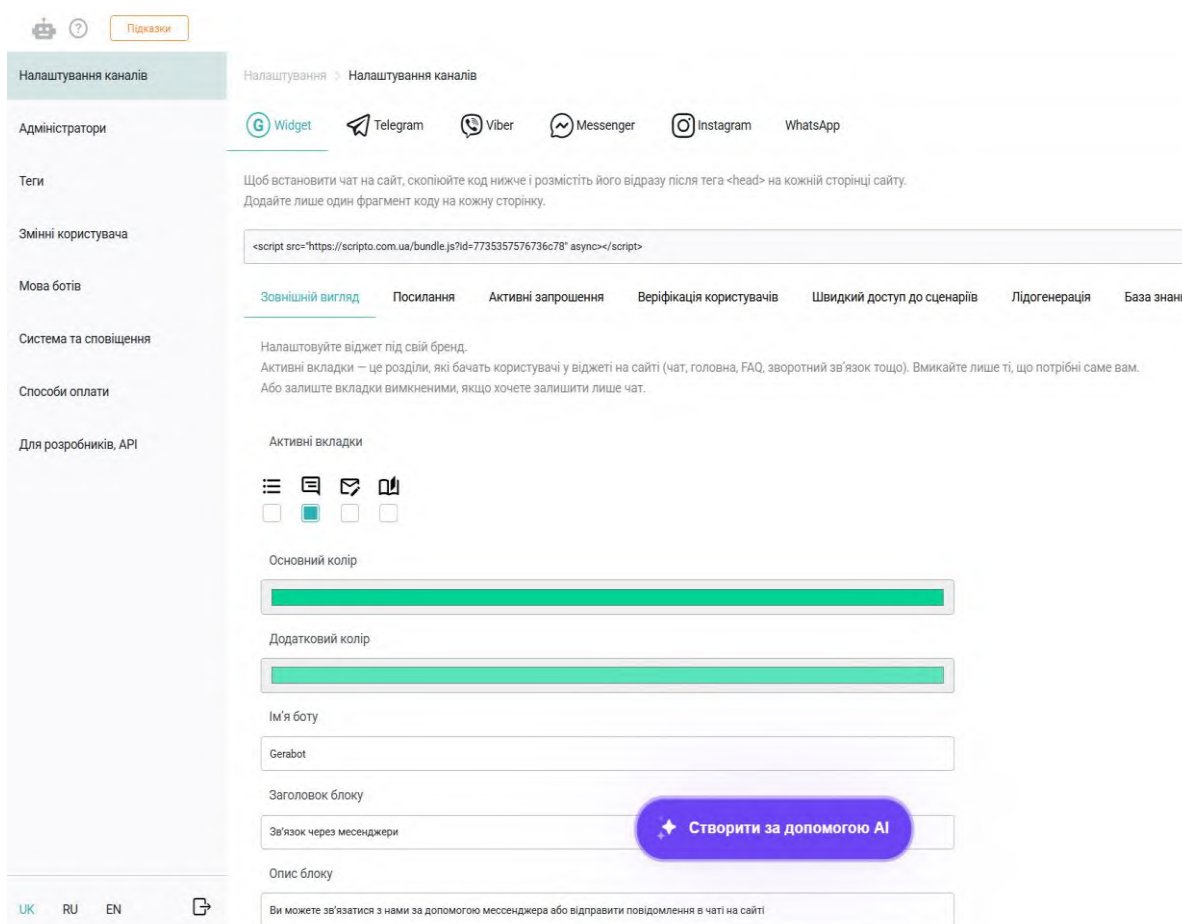


Рисунок 2.11 – Інтерфейс панелі керування та клієнтського віджета платформи Gerabot

Головною технологічною особливістю Gerabot є глибока інтеграція автоматизованого чат-бота з модулем онлайн-консультанта (живого оператора). Платформа здійснює детальний трекінг цільової аудиторії: система автоматично

збирає та передає оператору дані про місцезнаходження користувача, його IP-адресу, тип браузера, пристрій, історію переходів сторінками сайту та час перебування на них. Більше того, інструментарій дозволяє оператору бачити текст, який користувач лише набирає, ще до моменту його відправки. Уся зібрана інформація автоматично зберігається у вбудованій CRM-системі у так званий «карті клієнта». Платформа забезпечує поглиблену аналітику ефективності взаємодії: вона відстежує етапи, на яких комунікація зупиняється зі сторони клієнта, з метою подальшого коригування маркетингової стратегії та зменшення відтоку аудиторії. За заявами розробників, такий підхід дозволяє підвищити рівень лідогенерації на 35% та зменшити показник відмов на 50%. Незважаючи на високу технологічність системи трекінгу та аналітики, використання платформи Gerabot для створення інформаційної системи Гімназії №35 є недоцільним. Архітектура та весь набір інструментів цієї платформи жорстко оптимізовані під потреби відділів продажів, електронної комерції та комерційного обслуговування. Для комунального закладу загальної середньої освіти функціонал збору IP-адрес, аналізу поведінки на сайті для «дотискання» клієнтів та ведення воронок продажів у CRM є абсолютно нерелевантним і надлишковим.

Наступним інструментом для порівняльного аналізу є Flow XO [27] – потужне середовище для побудови та хостингу кросплатформних діалогових систем. Згідно з офіційною документацією, платформа вирізняється високою універсальністю каналів зв'язку: вона дозволяє одночасно розгортати ботів для Facebook Messenger, Telegram, SMS (за допомогою сервісу Twilio), Slack, а також надає готовий віджет для інтеграції безпосередньо на вебсайти. Архітектура розробки у цій системі базується на концепції потоків. Як зазначає експерт з діалогових архітектур С. К. Сінгх [28], потік – це логічна одиниця розмови, створена для певної теми або завдання. Потoki діють як модульні механізми чат-бота, які інкапсулюють конкретні домени комунікації, роблячи систему масштабованою, зручною для підтримки та логічно сегментованою. Фактично, потік являє собою набір станів та маршрутів, зосереджених на конкретній

функції. Процес розробки у цій платформі формалізовано і розбито на чіткі етапи, починаючи від проєктування цілей бота до його розгортання. Етапи розробки та інтерфейсу побудови потоків наведено на рисунку 2.12.

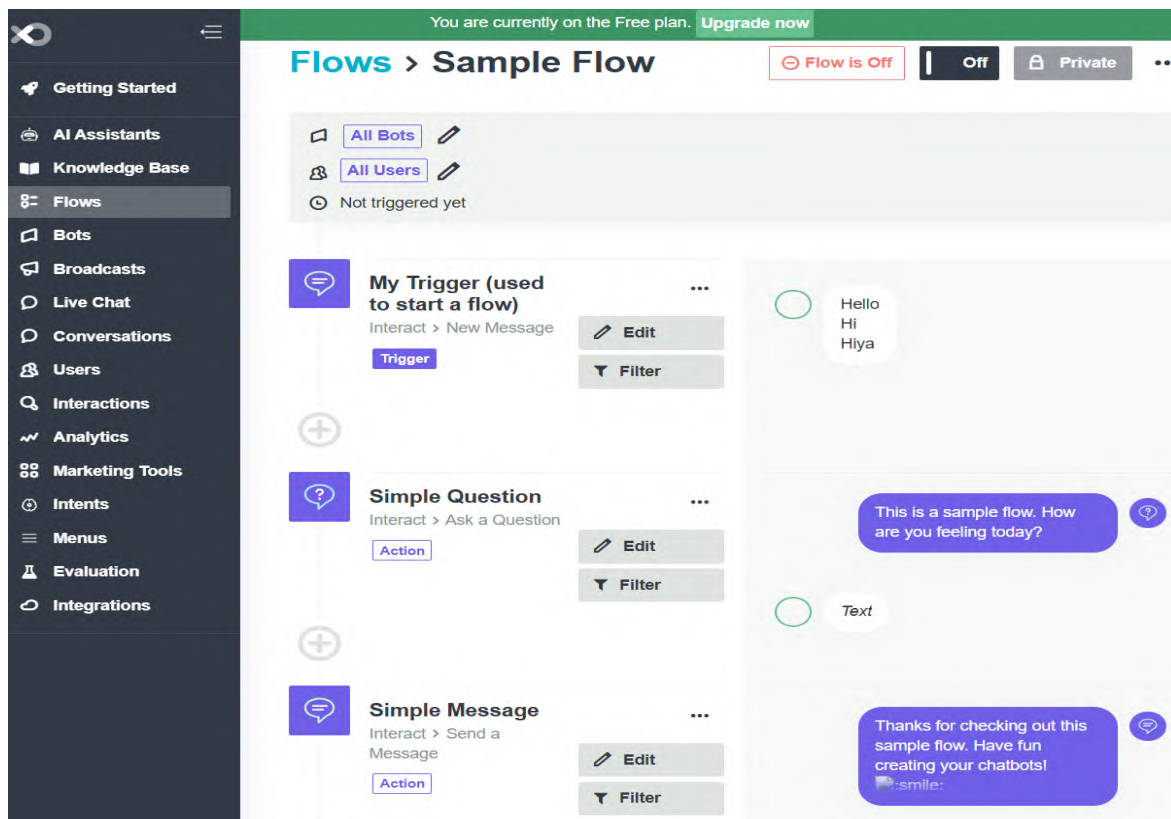


Рисунок 2.12 – Інтерфейс побудови діалогових потоків у платформі Flow XO

Ключовою технічною перевагою Flow XO є її деталізована система журналювання та зневадження (дебагінгу). Модуль журналу взаємодій (Interactions log) надає розробнику доступ до історії виконання кожного потоку: він дозволяє детально переглядати кожен виконаний крок, аналізувати вхідні та вихідні змінні, а також відстежувати хронометраж виконання. Для моніторингу стабільності роботи системи передбачено окремий модуль Error Log, який формує графічне відображення системних помилок за їхніми типами. Крім цього, платформа підтримує велику бібліотеку інтеграцій зі сторонніми додатками, модулі управління аудиторією, масових розсилок та можливість безшовного перемикавання діалогу на живого оператора. Однак, незважаючи на

потужний аналітичний та інтеграційний інструментарій, платформа має певні обмеження в контексті розробки довідкової системи для закладу освіти. Інтерфейс створення потоків у Flow XO є переважно лінійним та формалізованим, що робить його значно менш наочним при проєктуванні складних, розгалужених дерев рішень порівняно з класичними редакторами. До того ж, значна частина функціоналу системи (зокрема набір інструментів Web Tools для прив'язки ботів до Facebook Ads або створення маркетингових послідовностей) орієнтована виключно на бізнес-сектор, електронну комерцію та лідогенерацію. Для державної гімназії, яка потребує інтуїтивно зрозумілого візуального конструктора для швидкого створення довідкових багаторівневих меню, такий перевантажений лінійний інтерфейс є неоптимальним рішенням.

Останнім інструментом, що розглядається у межах порівняльного аналізу, є платформа SendPulse [29], варто зазначити, що вона є багатофункціональним інструментом для автоматизації комунікацій. Згідно з офіційною документацією, сервіс пропонує конструктор ланцюжків повідомлень із зручним інтерфейсом, візуальне подання якого наведено на рисунку 2.13. Цей конструктор містить понад 10 базових елементів для побудови різноманітних сценаріїв від простих привітань до складних воронки продажів. Платформа підтримує омніканальність, дозволяючи створювати та керувати ботами для Telegram, Facebook Messenger, WhatsApp, Viber та онлайн-чатів на сайтах з єдиного інтерфейсу. Важливою технічною особливістю є можливість збору та збереження інформації про користувачів за допомогою системи змінних та тегів, що дозволяє сегментувати аудиторію та персоналізувати повідомлення. Для розширення базового функціоналу SendPulse надає широкі можливості інтеграції із зовнішніми системами. Це реалізується за допомогою блоку «Запит API», підтримки Chatbots service API та можливості налаштування вебхуків для передачі даних або запуску сценаріїв на основі зовнішніх подій. Крім того, платформа має вбудовану систему детальної статистики, яка відстежує кількість сесій, повідомлень, кліків на кнопки та конверсії за кожним окремим ланцюжком.

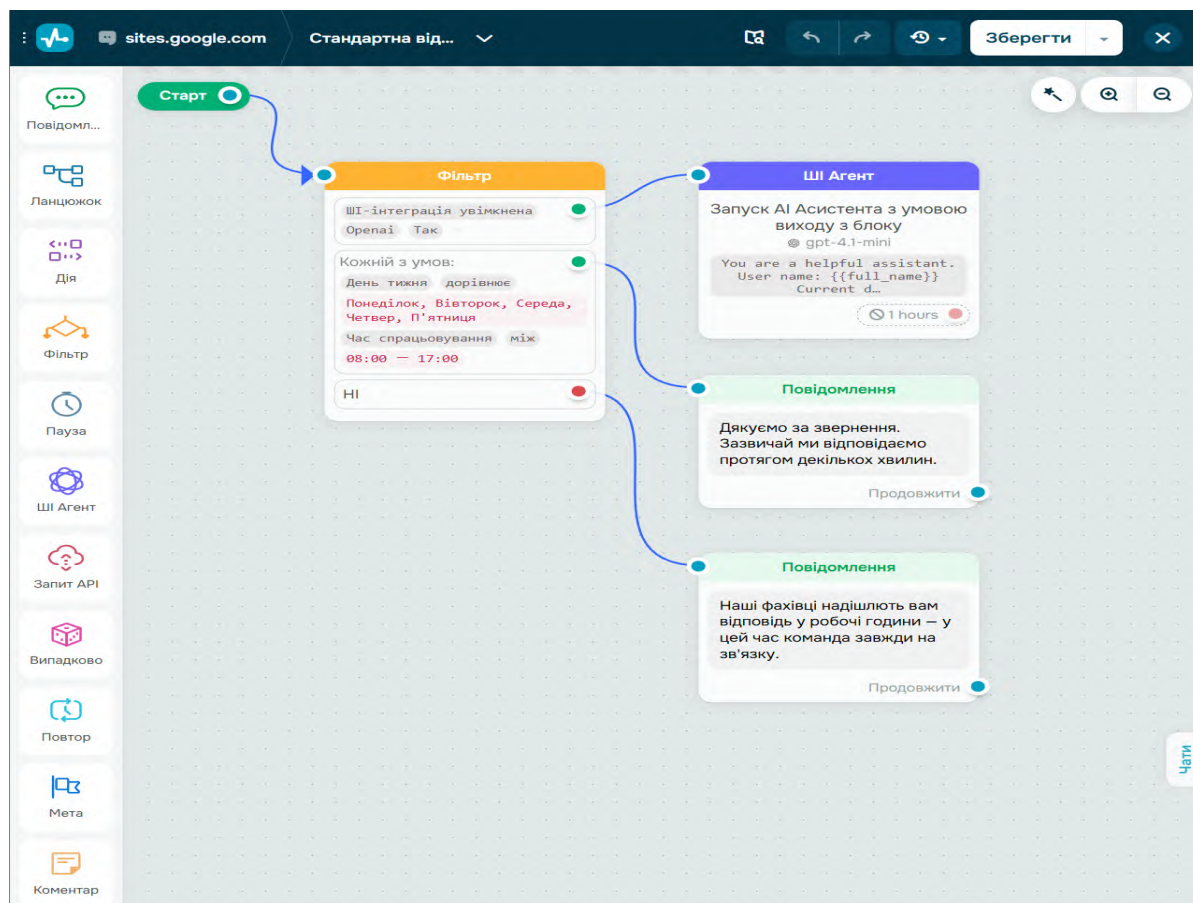


Рисунок 2.13 – Інтерфейс візуального конструктора ланцюжків повідомлень платформи SendPulse

Проте, незважаючи на потужний інструментарій, архітектура платформи має яскраво виражене маркетингове та комерційне спрямування. Її логіка тісно пов'язана з іншими сервісами власної екосистеми: вбудованою CRM-системою для автоматичного створення контактів та управління угодами (наприклад, зміна статусу угоди після оплати в боті), модулем реєстрації на курси та інструментами для запуску масових Email-, SMS- та Push-розсилок. Така жорстка орієнтація на лідогенерацію, монетизацію та воронки продажів робить інструментарій платформи перевантаженим і технологічно надлишковим для створення строгої довідкової системи державного закладу освіти, головною метою якої є зручна дистрибуція регламентованої публічної інформації, а не стимулювання продажів.

Оцінювання платформ для розробки інформаційної системи Гімназії №35 проводилося за вісьмома ключовими технічними критеріями, які безпосередньо

впливають на архітектуру розгортання довідкового чат-бота. Для об'єктивного вибору інструментальних засобів результати цього аналізу зведено у єдину порівняльну таблицю, яку наведено у Додатку Г.

Проведений порівняльний аналіз доводить, що створення чат-бота за допомогою написання програмного коду є недоцільним для освітнього закладу через високі вимоги до кваліфікації та часових витрат. Використання готових візуальних конструкторів є найефективнішим інженерним рішенням. Зіставивши можливості платформ SendPulse, Gerabot, Flow XO та UChat, було визначено, що вузькоспеціалізовані маркетингові сервіси не відповідають цілям гімназії. Найкращим інструментом обрано платформу UChat, яка органічно поєднує гнучке візуальне програмування, потужну підтримку омніканальності, наявність мобільного клієнта та розширені можливості інтеграції із зовнішніми базами даних. Використання саме цього середовища дозволить швидко та якісно розгорнути детермінованого довідкового чат-бота для Гімназії №35.

2.4 Формування інформаційної бази та проєктування логіки діалогів

Перш ніж розпочати розробку чат-бота, необхідно створити репрезентативну інформаційну базу, яку система використовуватиме для генерації відповідей на запити користувачів. Проєктована структура меню повинна обов'язково базуватися на інтегративному підході. Відповідно до досліджень Н. Б. Ларіонової та Н. М. Стрельцової [30], інтегративний підхід в умовах закладу освіти передбачає цілісне об'єднання різних відокремлених складових освітнього середовища у єдину, взаємопов'язану систему. У контексті розробки нашого чат-бота застосування цього підходу означає, що його інформаційна архітектура має органічно охоплювати адміністративні, куррикулярні (розклад, навчання) та позакласні аспекти життя гімназії, надаючи користувачу комплексну картину. Для формування такої бази даних Гімназії №35 було проведено збір інформації з офіційного вебсайту закладу на платформі

Google Sites, а також проаналізовано найбільш поширені запити, що надходять до адміністрації:

- загальна інформація про розклад та канікули;
- інформація про дистанційне навчання;
- необхідні документи та правила зарахування до 1 класу;
- наявність вакантних місць та територія обслуговування;
- інформація про адміністрацію та публічні документи;
- організація харчування та робота бібліотеки;
- психологічна підтримка та протидія булінгу;
- інформація про укриття та безпеку.

Зібраний неструктурований контент був поділений на чотири ключові інформаційні блоки: «Навчальний процес», «Вступна компанія», «Про Гімназію» та «Допомога та безпека». Для забезпечення інтуїтивно зрозумілої навігації було спроектовано логічну модель у вигляді ієрархічного дерева рішень на рисунку 2.14. Вона описує всі можливі стани чат-бота та варіанти переходів між ними залежно від вибору користувача.

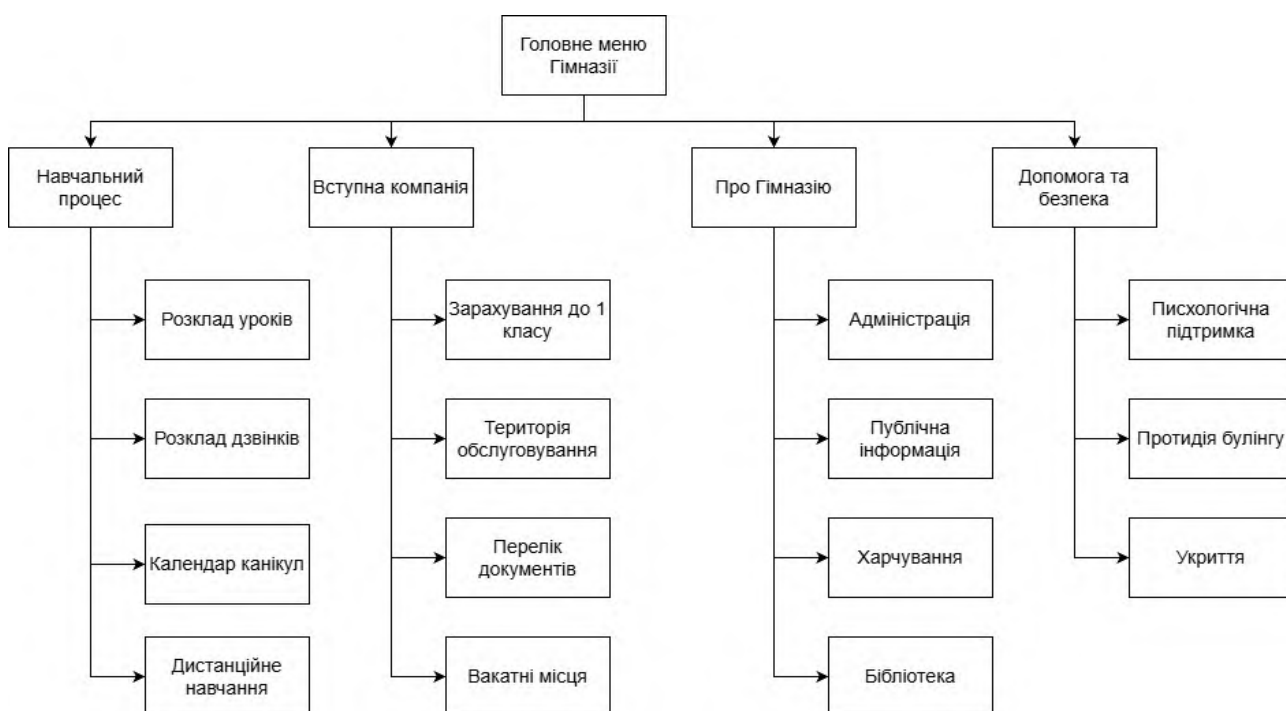


Рисунок 2.14 – Ієрархічна модель навігації для чат-бота Гімназії №35

Побудована вище ієрархія формує логічне дерево рішень. Шлях користувача починається зі стартового повідомлення. Бот обробляє цей запит і виводить на екран привітання разом із чотирма основними розділами головного меню. Обравши певний розділ (наприклад, «Вступна компанія»), користувач спускається на другий рівень ієрархії, де система пропонує йому уточнити свій запит за допомогою підменю. Після вибору конкретного пункту (наприклад, «Перелік документів»), бот звертається до відповідної комірки інформаційної бази та миттєво генерує кінцеву відповідь – текст із прикріпленим файлом. Ключовою особливістю розробленої логіки є те, що кожна кінцева відповідь обов'язково супроводжується кнопкою «Повернутися до головного меню». Це унеможливорює потрапляння користувача у «глухий кут» діалогу та забезпечує безперервність навігації системою.

2.5 Практична реалізація функціоналу чат-бота

Після завершення етапу проєктування інформаційної бази та формування логічної моделі дерева рішень, було розпочато безпосередній процес технічного втілення проєкту. Практична реалізація інформаційної системи Гімназії №35 на базі обраної платформи UChat складалася з чотирьох послідовних етапів: реєстрації бота, конструювання діалогових потоків, програмування інтерфейсу та налаштування тригерної системи.

Першим етапом стала генерація API-токена та його інтеграція з платформою розробки. API (Application Programming Interface) [31] – це посередник між програмами, який задає правила «спілкування». Тому в його назві закладено поняття «інтерфейс» – межа між об'єктами. Одній програмі не важливо, як працює інша. Розробники просто користуються інтерфейсом: надсилають запит і отримують зворотну відповідь. Створення будь-якого чат-бота для месенджера Telegram розпочинається у його внутрішньому системному середовищі. Для цього було використано офіційного бота-конфігуратора

@BotFather. Безпосередній процес реєстрації розпочався із введення системної команди /newbot. Спершу було задано публічне ім'я бота – «Гімназія №35 Чат-Бот», яке відображатиметься користувачам у списку контактів та в заголовку діалогу. На вимогу BotFather, було вказано унікальний ідентифікатор (username) – Gymnasium35Bot. Після успішної перевірки унікальності нікнейму, системний бот підтвердив реєстрацію та згенерував унікальний HTTP API-токен, який зображений на рисунку 2.15.

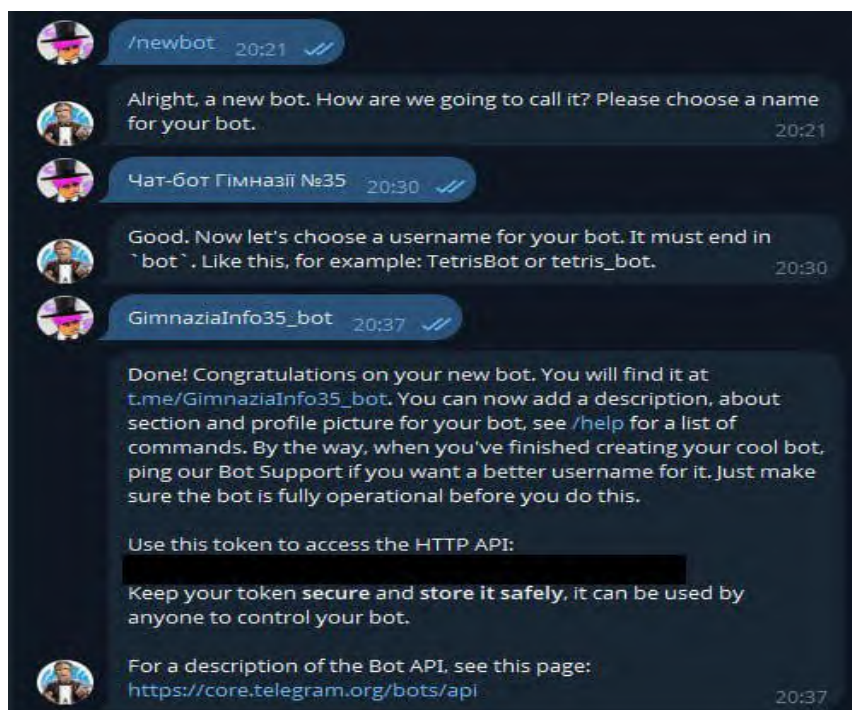


Рисунок 2.15 – Реєстрація профілю бота та генерація HTTP API-токена у середовищі @BotFather

Наступний крок конфігурації профілю розпочався із введення системної команди /setabouttext. У відповідь на запит системи було обрано створений раніше бот @GimnaziaInfo35_bot та введено текст опису: «Офіційний бот-помічник Гімназії №35», що зображено на рисунку 2.16. Для візуальної ідентифікації закладу освіти було завантажено офіційний логотип гімназії, використовуючи команду /setuserpic, результат якої наведено на рисунку 2.17. Наявність унікального профільного зображення підвищує впізнаваність інформаційної системи серед користувачів месенджера.

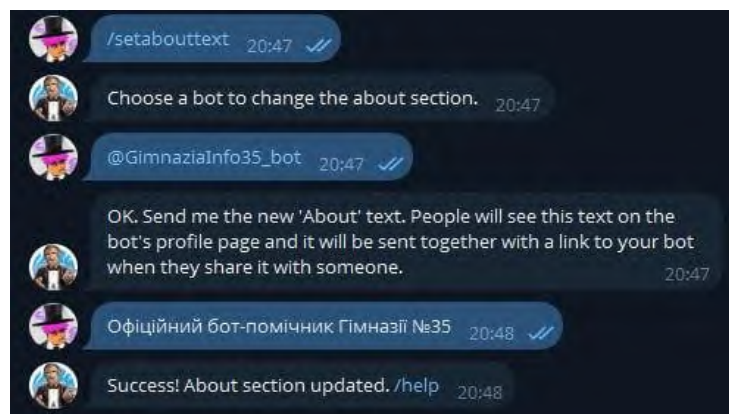


Рисунок 2.16 – Налаштування інформаційного блоку профілю бота у середовищі @BotFather



Рисунок 2.17 – Налаштування інформаційного блоку профілю бота у середовищі @BotFather

Подальше налаштування здійснювалося у робочому просторі платформи UChat було створено новий проєкт для каналу Telegram, куди інтегровано отриманий API-токен, зображений на рисунку 2.18. Це забезпечило платформі UChat права на прослуховування вхідних повідомлень та надсилання відповідей від імені створеного шкільного бота.

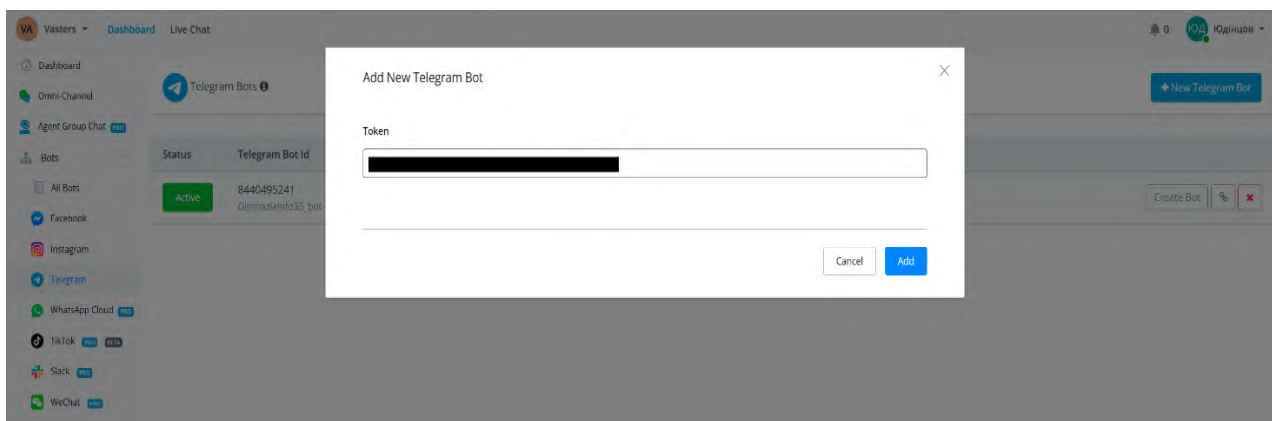


Рисунок 2.18 – Ініціалізація підключення каналу Telegram у робочому просторі платформи UChat

Другим етапом було конструювання Flow-ланцюжків у візуальному редакторі. Технічна реалізація спроектованого раніше ієрархічного здійснювалася у візуальному Drag & Drop редакторі платформи, що показано на рисунку 2.19. Відповідно до технічної документації [32], технологія Drag and Drop – це спосіб взаємодії з графічним інтерфейсом, що дозволяє користувачу захоплювати віртуальні елементи за допомогою комп'ютерного маніпулятора, переміщувати їх та фіксувати у визначеній цільовій області. Завдяки використанню такого інтерфейсу, архітектура системи була розбита на головний потік та кілька допоміжних підпотоків, що відповідають за окремі категорії. Процес конструювання полягав у розміщенні на робочому полотні функціональних вузлів і з'єднанні їх логічними зв'язками. Завдяки такому підходу кожен крок користувача був алгоритмізований, а розбиття масивного дерева на окремі підпотоки дозволило уникнути візуального перевантаження головного екрана конструктора. Третім етапом стало наповнення розроблених блоків контентом та створення багаторівневої Inline-клавіатури для навігації між вузлами системи. Для забезпечення безперервного користувацького досвіду та уникнення «тупикових» гілок діалогу, де система могла б зупинитися, у блоках були передбачені навігаційні кнопки повернення (наприклад, «Назад» або «Головне меню»), що дозволяють легко перемикатися між рівнями ієрархії. Частина налаштування такої багаторівневої навігації зображено на рисунку 2.20.

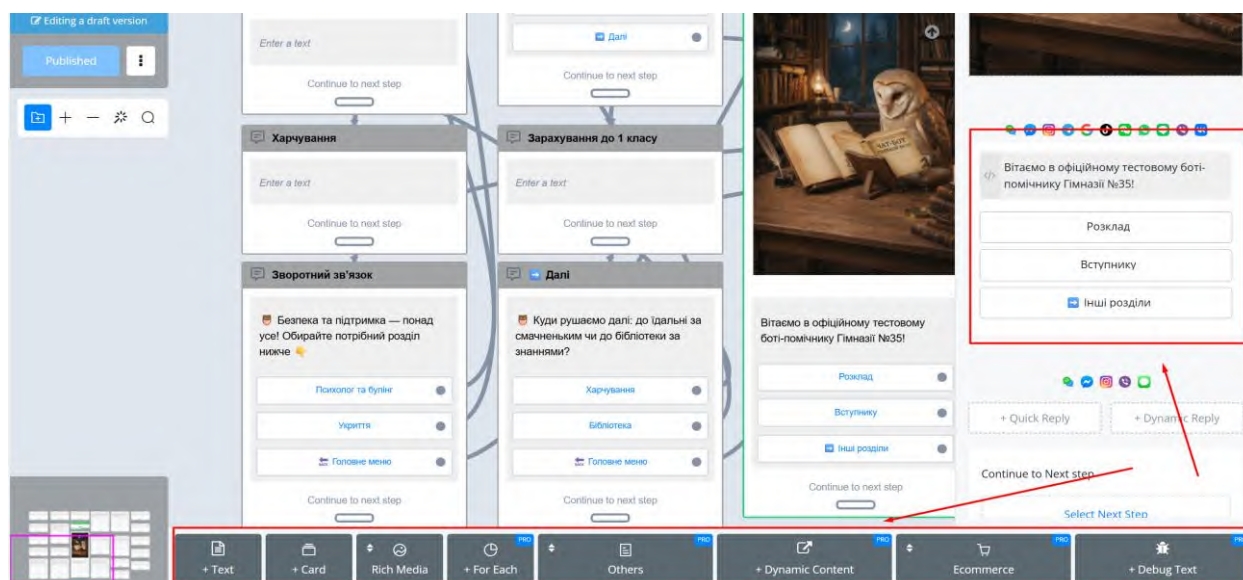


Рисунок 2.19 – Процес конструювання Flow-ланцюжків у візуальному Drag & Drop редакторі

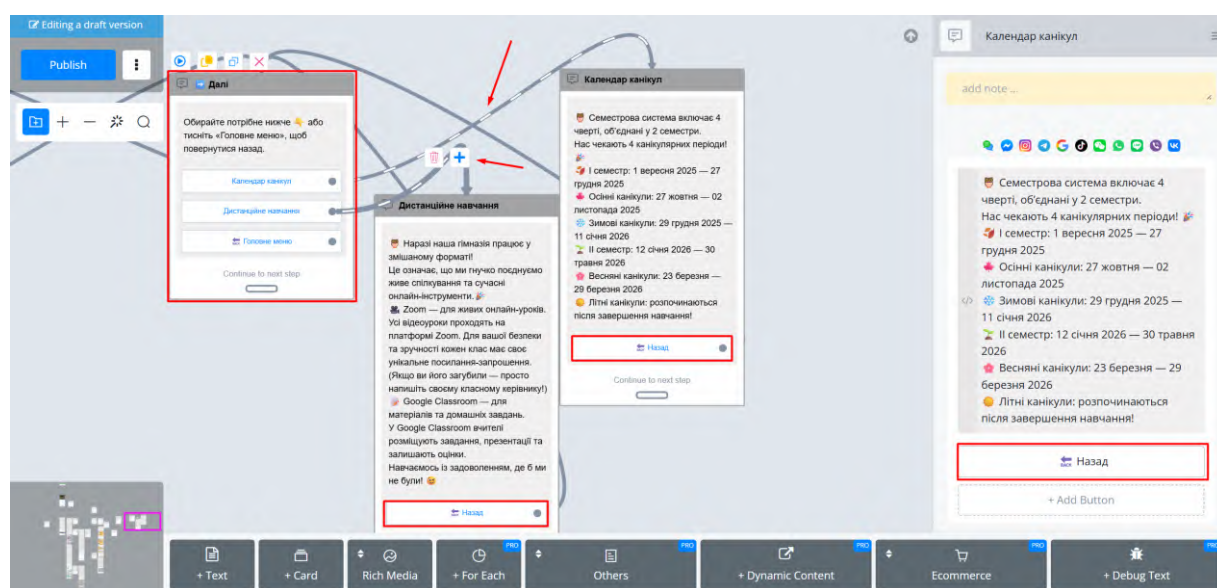


Рисунок 2.20 – Реалізація багаторівневої навігації та кнопок повернення

Після налаштування структури, блоки були заповнені контентом з використанням емоцій. Особливістю реалізації текстової частини стало використання вбудованої функції платформи – динамічної рандомізації тексту (Random Text). Як показано на рисунку 2.21, цей інструмент дозволив задати кілька варіантів однієї фрази. Під час кожного нового виклику меню система випадковим чином обирає один із заданих варіантів, що робить спілкування з ботом більш природним та менш шаблонним.

Random Text

Only one of the following variant text will be randomly picked and used.

Main Variant Text

Усе найцікавіше про гімназію та контакти для зв'язку сховані тут 70/100

Other Variant Text

Цікаво дізнатися про нас більше? Тисніть на кнопки, я залюбки розповім! 71/100

Other Variant Text

Маєте пропозиції чи хочете почитати про школу? Обирайте потрібну кнопку. 75/100

Other Variant Text

Відкриваємо всі секрети! Тут історія нашої гімназії та форма для зв'язку. 76/100

Other Variant Text

Бажаєте познайомитися ближче чи маєте запитання? Обирайте розділ нижче! 77/100

+ Add Variant Text

Рисунок 2.21 – Налаштування функції динамічної рандомізації тексту

На четвертому, завершальному етапі, відбулося налаштування тригерів та комплексна перевірка працездатності системи. У контексті розробки нашого чат-бота, тригер [33] – це механізм, який запускає автоматизацію при виникненні певної події або умови. В автоматизованих системах тригер визначає початкову умову для робочого процесу, вирішуючи, коли саме починається процес і за яких умов він має діяти. Вони є точкою входу до автоматизації, дозволяючи системі реагувати на події автоматично та відокремлюючи логіку роботи від ручного виконання. Особливістю архітектури обраної платформи є те, що головний стартовий тригер генерується системою автоматично під час створення нового діалогового ланцюжка, через що його складна ручна настройка є надлишковою. Як показано на рисунку 2.22, інтерфейс візуального редактора містить бічну панель із розширеним переліком доступних системних подій та панель швидкого додавання. Проте базовим тригером за замовчуванням визначено центральний стартовий блок, який автоматично обробляє системну команду /start.

У правій бічній панелі налаштувань цього вузла вже задано жорстку маршрутизацію, що ініціює перехід до наступного блоку відправки повідомлення. Саме цей підключений вузол активує привітальний текст із зображенням та виводить головне навігаційне меню. Також у системі задіяно

вбудований механізм платформи, що автоматично реагує на будь-який непередбачений текст або символи, забезпечуючи повернення користувача до коректної навігації. У разі ручного введення довільного повідомлення система розпізнає його як нецільове та відразу повторно генерує стартове меню для підтримки безперервного користувацького досвіду взаємодії.

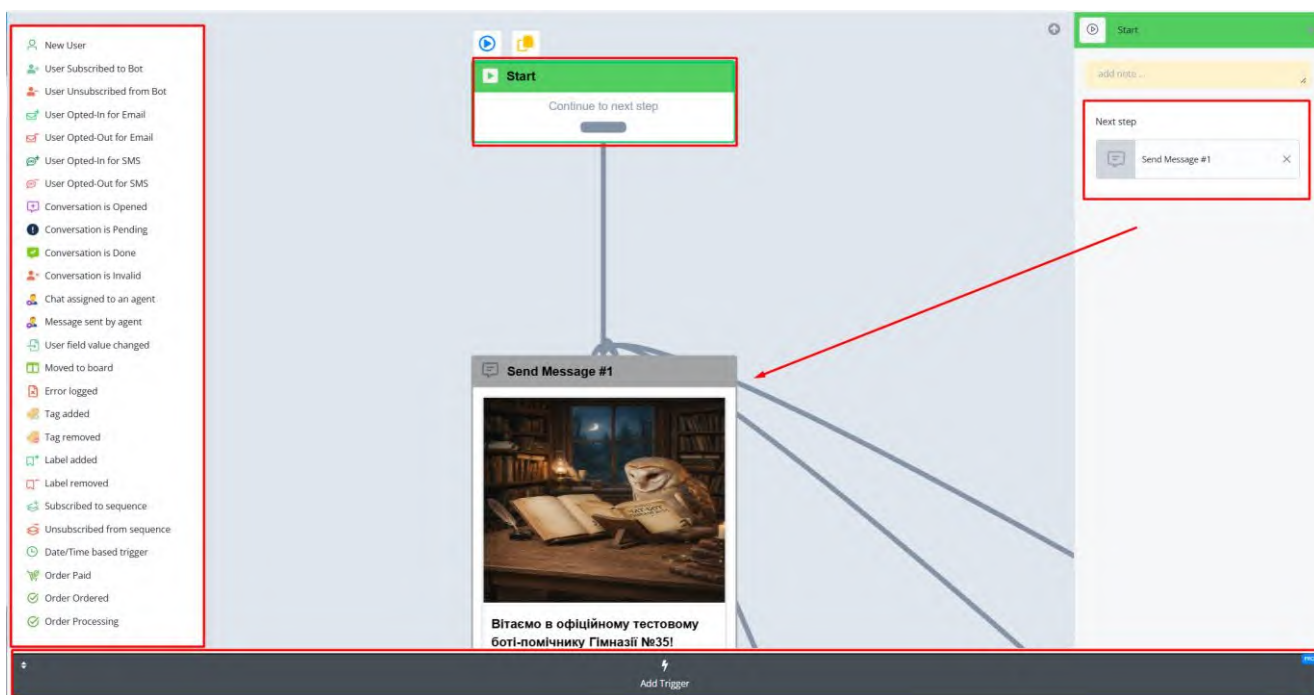


Рисунок 2.22 – Інтерфейс конфігурації тригерів та автоматичне налаштування маршрутизації стартової команди /start

Оскільки налаштування тригерів було фінальним кроком, після інтеграції необхідних медійних файлів у відповідні вузли процес розробки чат-бота було остаточно завершено. Загальну схему повністю побудованої архітектури наведено в Додатку Д.

Отже, у другому розділі було здійснено комплексне проєктування та практичну реалізацію чат-бота для Гімназії №35. Обґрунтовано вибір детермінованої архітектури для запобігання генерації некоректних відповідей та платформи UChat завдяки її омніканальності й гнучкому Drag & Drop інтерфейсу. Інформаційну базу систематизовано у чотири цільові блоки, навігацію між якими побудовано у вигляді ієрархічного дерева рішень.

РОЗДІЛ 3

ВПРОВАДЖЕННЯ, ТЕСТУВАННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ ЧАТ-БОТА

3.1 Обґрунтування вибору платформи та технологій розробки

Для максимального охоплення аудиторії реалізовано безшовну інтеграцію Telegram-бота з вебресурсом закладу освіти. Такий підхід узгоджується з дослідженням дидактичного потенціалу освітніх чат-ботів Д.М. Юдінцова [34]. Замість перенаправлення за посиланнями, бот працює безпосередньо в браузері як вбудований віджет. Технічно рішення опирається на клієнт-серверну архітектуру [35], де вебклієнт взаємодіє з серверною логікою. Для цього використано вбудований функціонал омніканальності платформи-конструктора UChat, який дозволяє транслювати всю готову логіку Telegram-бота у форматі вебвіджета. Процес інтеграції охоплює три ключові етапи: синхронізацію логіки через механізм Omnichannel, генерацію та кастомізацію віджета, а також вбудовування коду у Google Sites.

На першому етапі здійснювалася прив'язка вже створеної структури діалогів та інформаційної бази Telegram-бота до каналу вебчату в середовищі конструктора. Процес інтеграції розпочався у середовищі конструктора UChat із налаштування нового каналу зв'язку. На головній сторінці налаштувань було обрано вкладку «Omni Channel». Як показано на рисунку 3.1, наступним кроком став вибір функції конвертувати існуючого бота, що дозволяє перенести користувачів та логіку з уже готового каналу. На нижній частині колажу відображено завершальний етап цієї дії: у діалоговому вікні зі списку доступних інтеграцій було обрано цільовий Telegram-бот із назвою «Чат-Бот Гімназія №35» для його перетворення на єдину систему. Це ініціювало створення вебінтерфейсу на основі наявної логіки Telegram-бота. Подальше налаштування інтеграції продовжили в каналі «Web Chat», що наведено на рисунку 3.2.

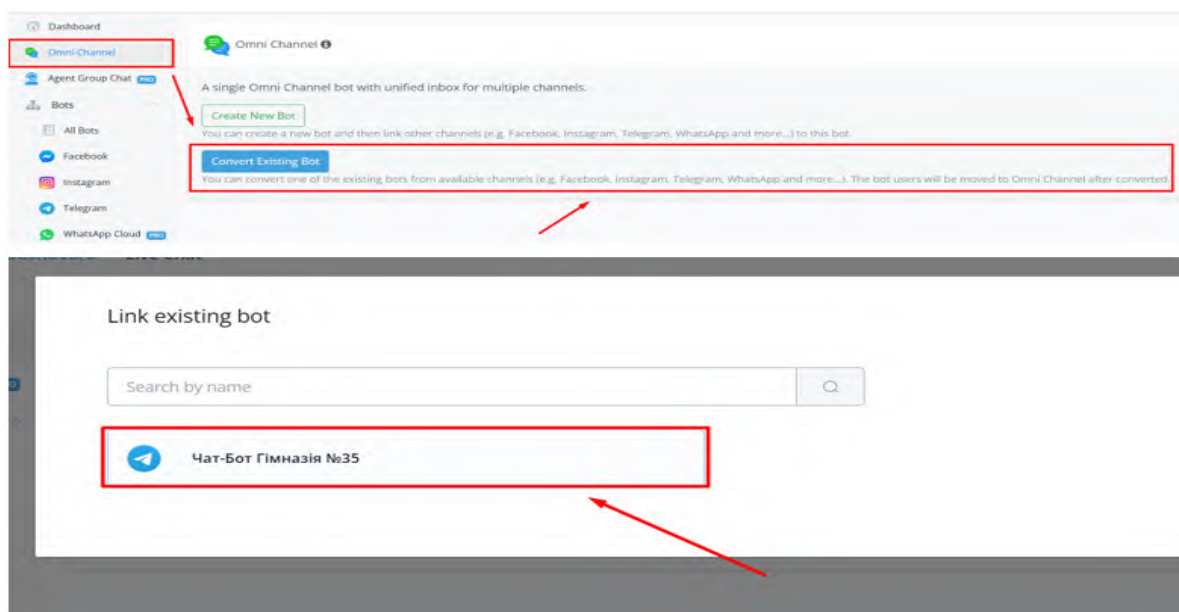


Рисунок 3.1 – Процес конвертації Telegram-бота в омніканальний формат

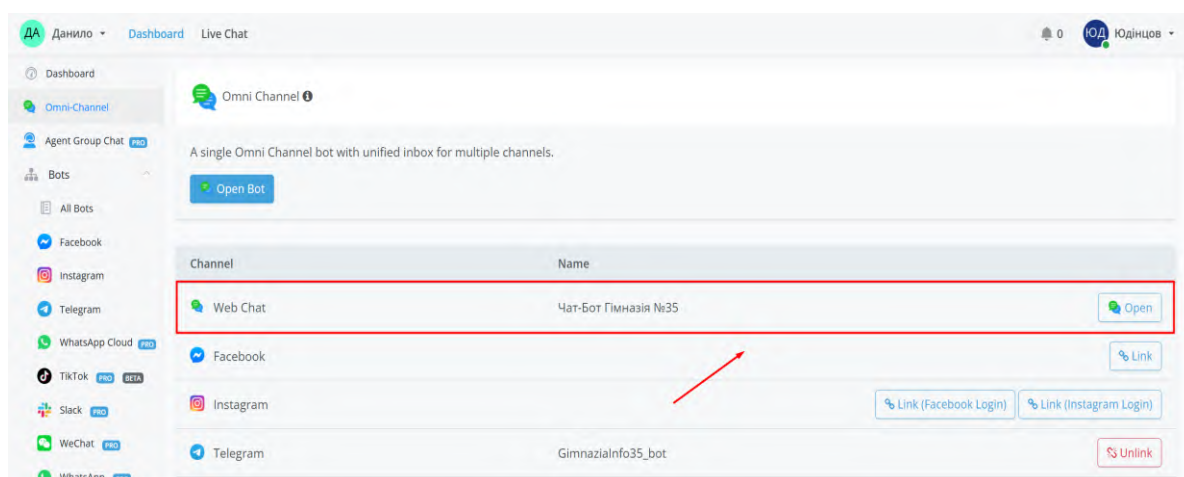


Рисунок 3.2 – Перехід до налаштувань каналу «Web Chat» в омніканальній системі

Наступним кроком стало підключення віджета чат-бота до вебсайту школи через омніканальний канал «Web Chat». Для цього в середовищі UChat у лівій навігаційній панелі було відкрито вкладку «WebChat Widget». Як видно з рисунка 3.3, візуальний редактор платформи дозволяє налаштовувати окремі елементи контенту та в режимі реального часу переглядати, як саме відвідувач сайту бачитиме інтегровані медійні файли та навігаційні кнопки, зокрема розклад уроків та дзвінків.



Рисунок 3.3 – Інтерфейс кастомізації вебвіджета у робочому просторі UChat

Для забезпечення безпеки роботи віджета та запобігання його несанкціонованому використанню на сторонніх ресурсах, у налаштуваннях було задано дозволений домен. Оскільки сайт гімназії розгорнуто на базі сервісу Google Sites, дозволений домен було визначено як «sites.google.com». Це унеможливорює використання згенерованого скрипта на будь-яких інших вебресурсах, як показано на рисунку 3.4.

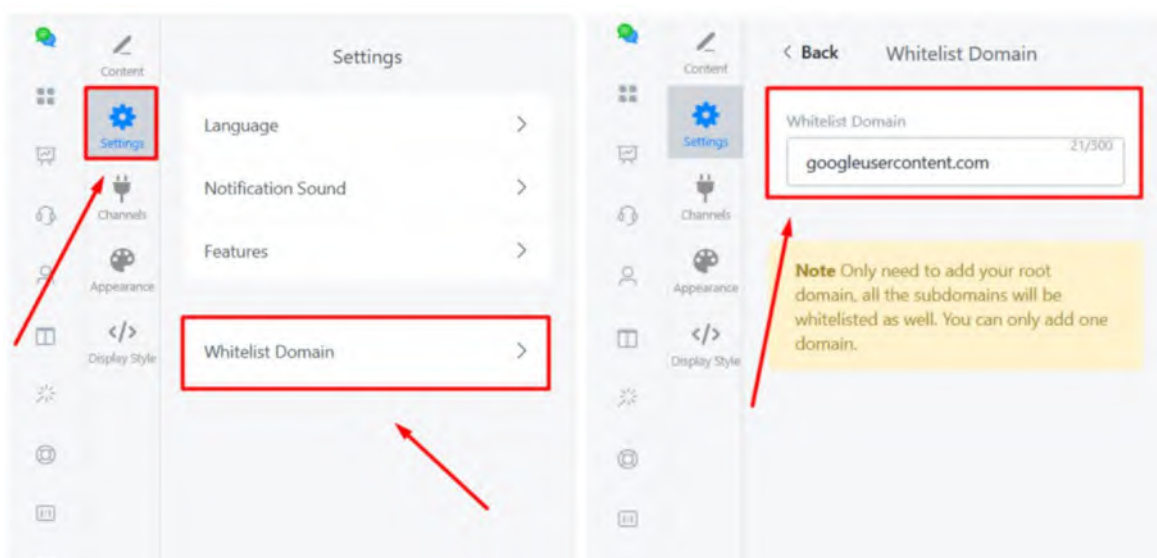


Рисунок 3.4 – Додавання домену до білого списку

Після завершення базових налаштувань та конфігурації безпеки, платформа UChat згенерувала спеціальний асинхронний HTML/JavaScript-код (скрипт) для подальшого вбудовування, який відображено на рисунку 3.4.

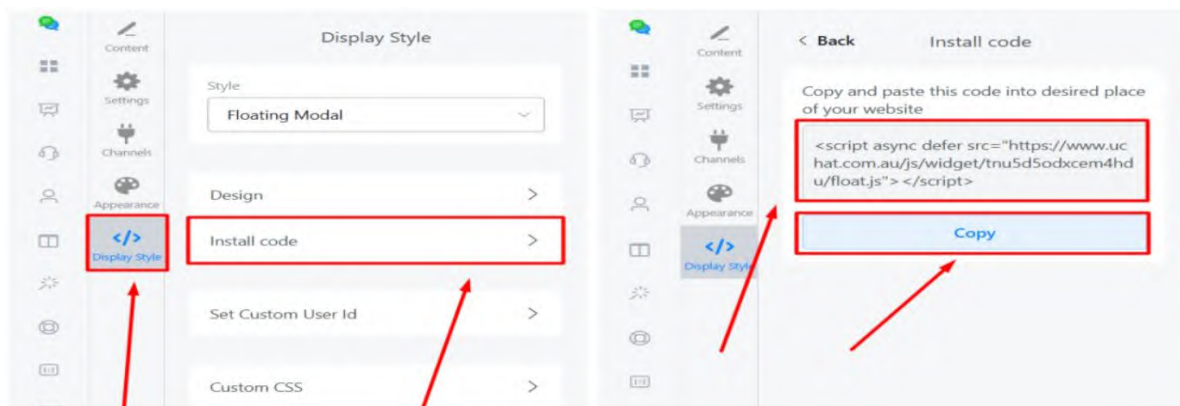


Рисунок 3.5 – Генерація унікального JavaScript-коду для інтеграції на сайт

Завершальним етапом стало безпосереднє вбудовування отриманого коду на головну сторінку та в ключові розділи офіційного сайту гімназії. Цей процес здійснювався за допомогою вбудованого інструмента платформи Google Sites – функції «Вбудувати». У режимі редагування сторінки було створено спеціальний блок вебконтенту, куди й було інтегровано скопійований скрипт, що показано на рисунку 3.6.

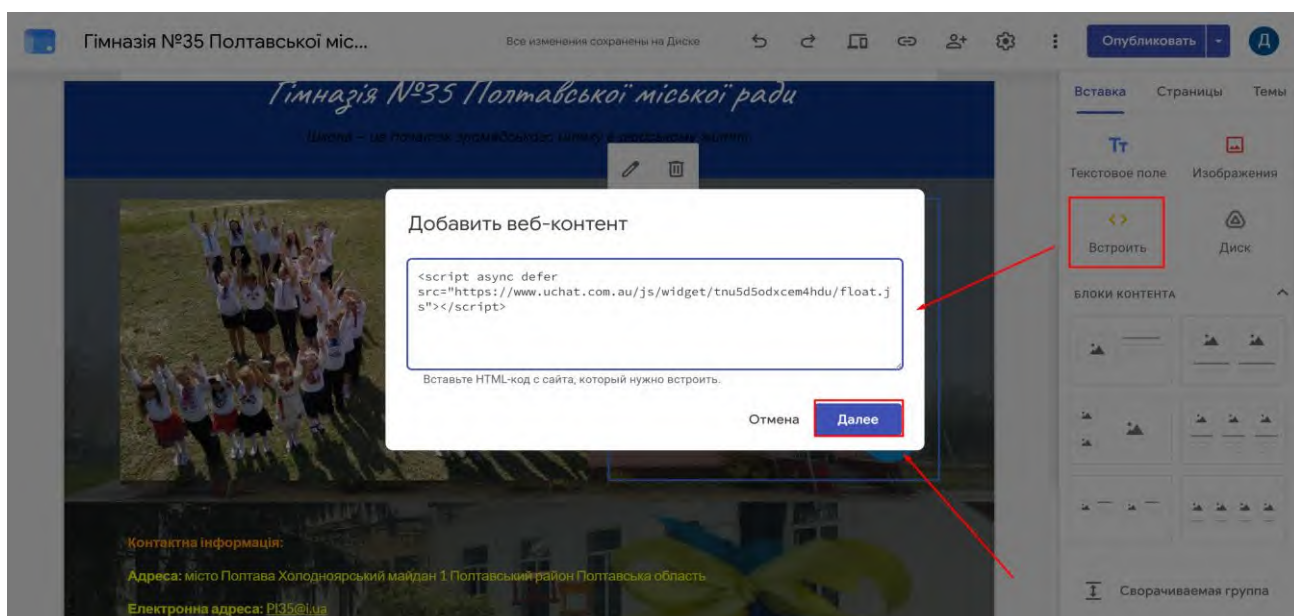


Рисунок 3.6 – Процес вбудовування HTML-коду чат-бота на сайт

У результаті такої інтеграції відвідувачі вебсайту отримали змогу взаємодіяти з чат-ботом та отримувати необхідну довідкову інформацію прямо у вікні браузера. На розгорнутому екрані віджета, зображеному на рисунку 3.7, користувача зустрічає кастомізоване привітальне повідомлення та інтерактивна кнопка «Розпочати розмову», яка запускає головне дерево навігації.

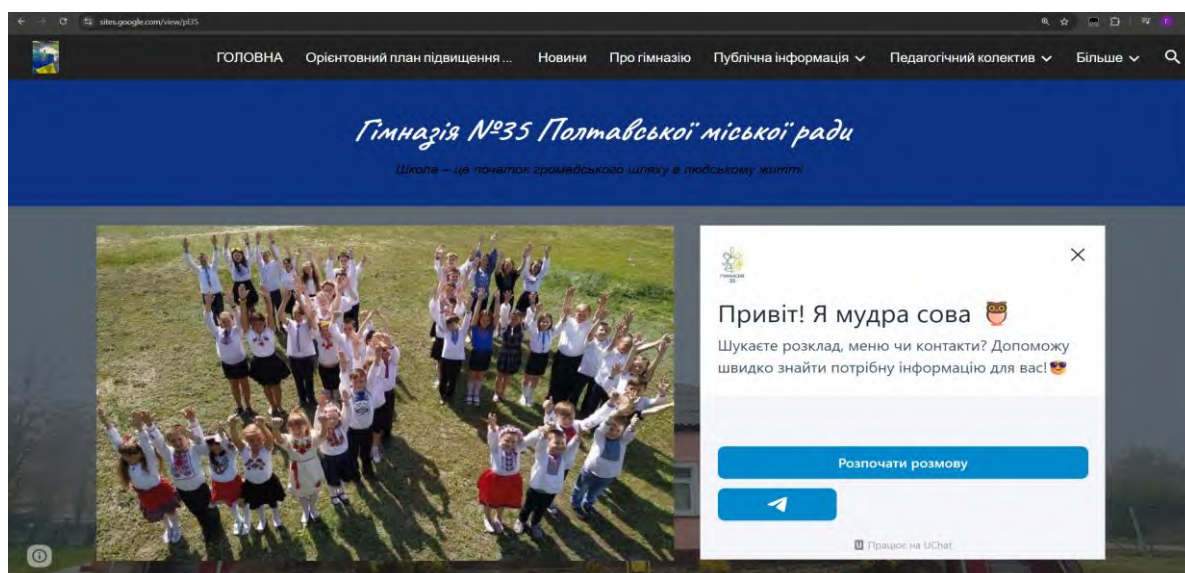


Рисунок 3.7 – Результат інтеграції віджета на вебсайт Google Sites

Це технічне рішення повністю усунуло необхідність переходити в сторонні додатки, такі як Telegram, або встановлювати їх, що значно знизило бар'єр входу та підвищило загальну доступність і зручність інформаційної системи школи.

3.2 Розробка методики та проведення тестування програмного продукту

Критично важливим етапом перед повноцінним запуском інформаційної системи стала перевірка її працездатності. Основу комплексного підходу до тестування склав спеціалізований чек-ліст, який дозволив верифікувати відповідність бота технічним вимогам. Процес апробації охоплював чотири ключові напрями.

У межах функціонального тестування здійснено перевірку коректності роботи всіх спроектованих гілок меню. Аналізувалася точність видачі інформації у відповідь на вибір кожного пункту меню. Перевірка користувацького досвіду у месенджері Telegram, результати якої наведено на рисунку 3.8, включала декілька тестових сценаріїв.

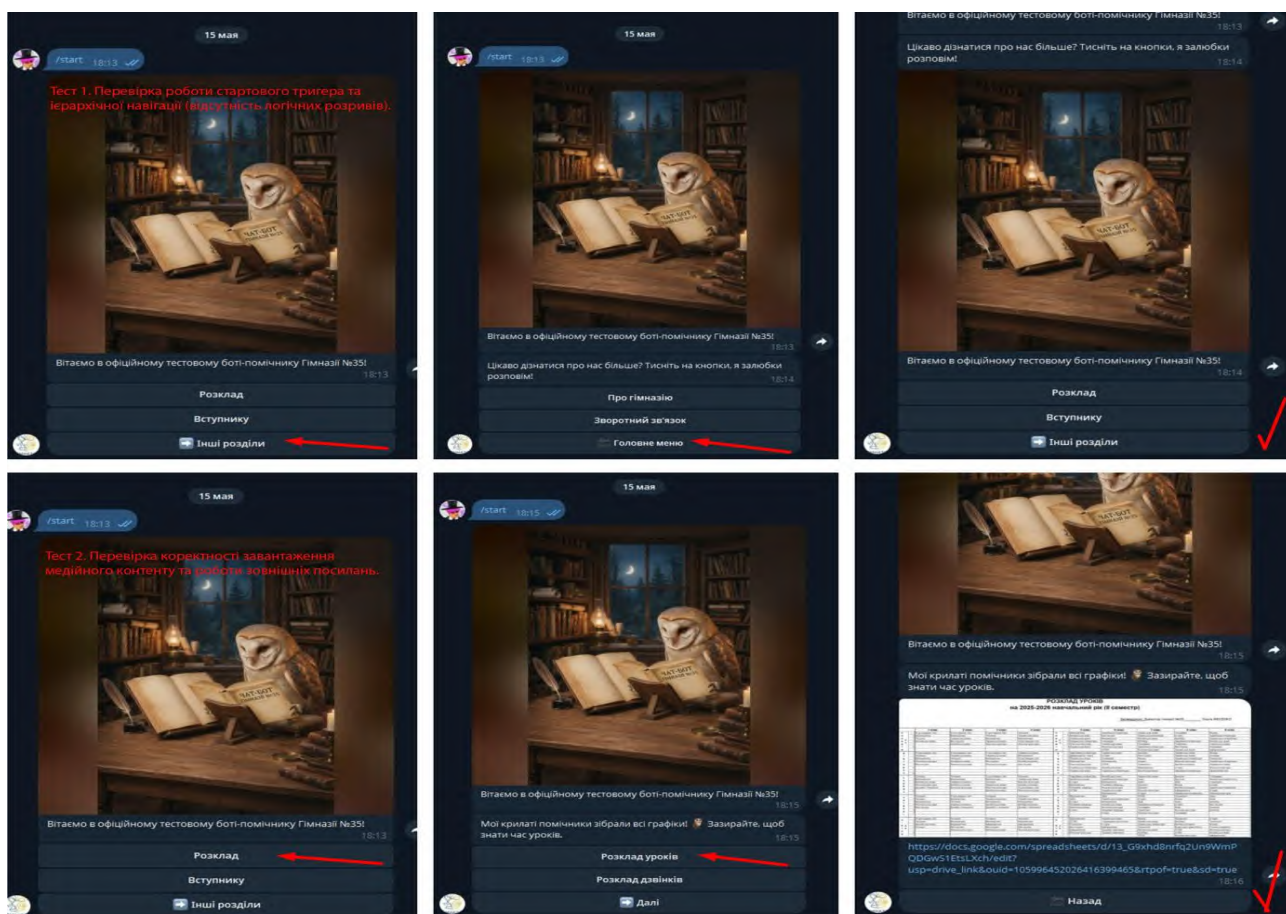


Рисунок 3.8 – Результати тестування базової навігації та відображення контенту в додатку Telegram

Перший тест підтвердив безперебійну роботу стартового тригера (/start) та відсутність логічних розривів при багаторівневій навігації з успішним поверненням до головного меню. Другий тест продемонстрував коректне відображення структури підрозділів у розділі «Вступна компанія» та точне надання інформації про документи для зарахування. Всі перевірені сценарії підтвердили відповідність фактичної поведінки системи очікуваним результатам.

Тестування інтерфейсу охоплювало оцінку візуального сприйняття контенту, читабельності тексту та коректності відображення інтерактивних кнопок на різних пристроях. Перевірці підлягали смартфони Android і десктопна версія вебсайту. За результатами тестування встановлено, що: всі кнопки меню відображаються коректно на екранах різних розмірів; текст повідомлень читабельний та структурований; вбудований вебвіджет на сайті гімназії коректно адаптується як під мобільний, так і під десктопний формат, що показано на рисунку 3.9.

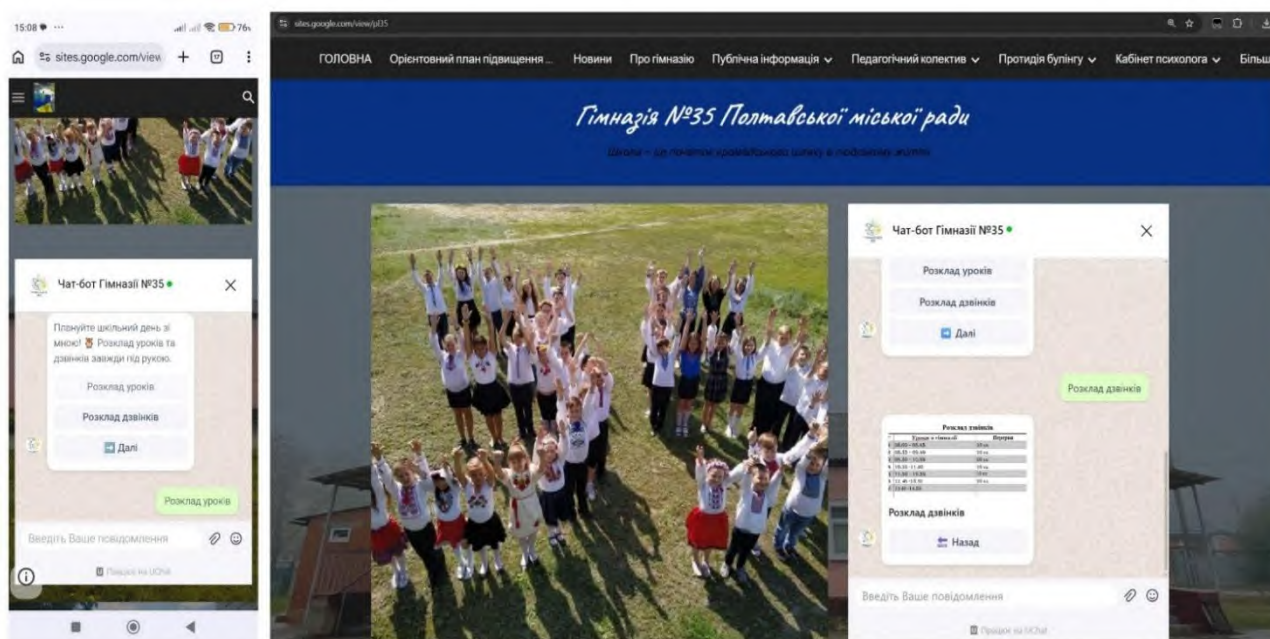


Рисунок 3.9 – Результати тестування вебвіджета на різних пристроях

Негативне тестування спрямоване на аналіз стійкості системи до нестандартних дій користувача. У ході тестування моделювалися ситуації, що виходять за межі передбаченого сценарію: введення довільного тексту (наприклад, запитання поза базою знань бота), надсилання медіафайлів (фотографій, голосових повідомлень), надсилання стікерів та емодзі. Як видно з рисунка 3.10, у всіх перевірених випадках чат-бот коректно опрацьовував непередбачувані введення: система автоматично відображала повідомлення-підказку з пропозицією скористатися кнопками головного меню. Це виключає можливість «зависання» бота у разі нестандартних дій користувача

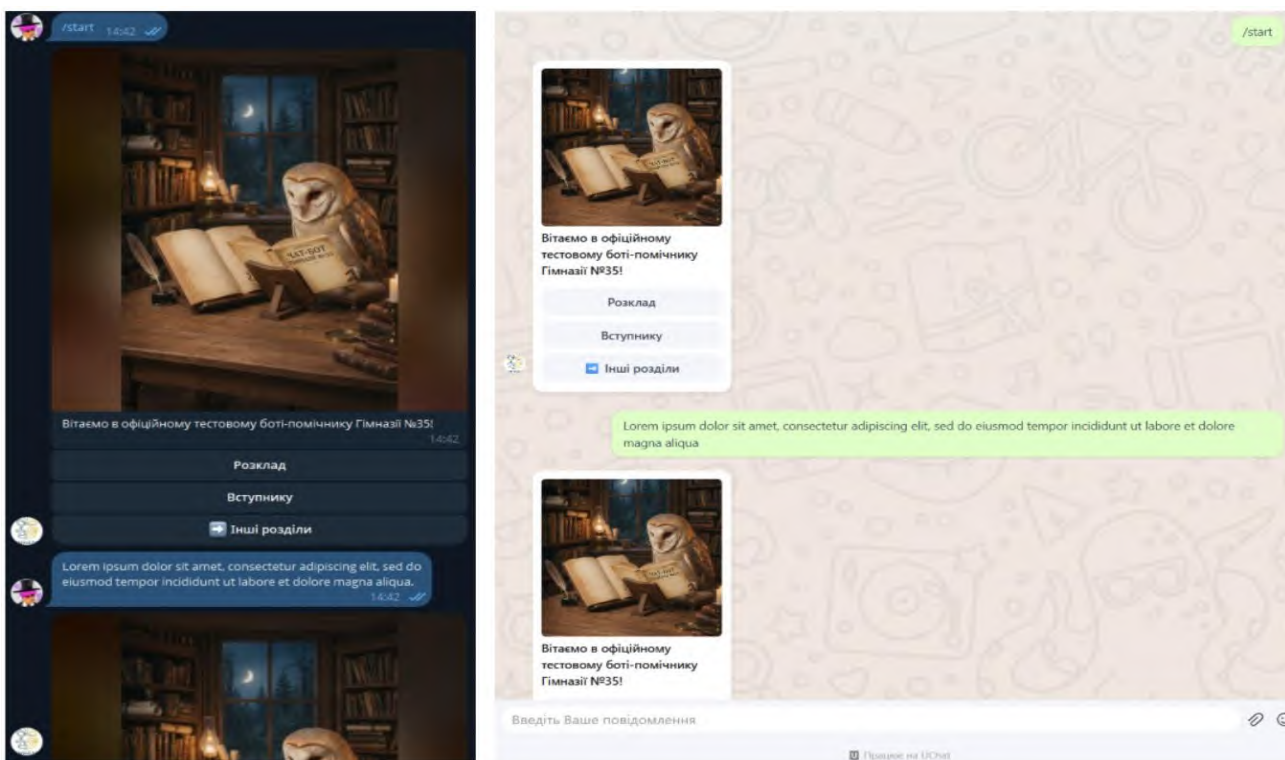


Рисунок 3.10 – Механізм повернення до головного меню в Telegram та вебвіджеті

Паралельно з основним процесом розробки здійснювалося комплексне кросплатформне тестування логіки роботи віртуального помічника у вбудованому вебвіджеті платформи UChat. Головною метою цього тестування було забезпечення стабільного функціонування всіх елементів системи незалежно від обраного середовища. Особлива увага приділялася завантаженню та коректному відображенню файлів формату PDF із актуальним розкладом занять та внутрішніми нормативними положеннями закладу. Важливо було переконатися, що навчальні матеріали відкриваються швидко і без системних помилок. Ключовою умовою успішного проходження цих тесту було збереження повного контексту розмови без збоїв алгоритму чи необхідності починати діалог спочатку. За загальними результатами проведеного кросплатформного тестування було успішно підтверджено, що поведінка інформаційної системи в месенджері Telegram та у вбудованому вебвіджеті є абсолютно ідентичною. Як ілюструє рисунок 3.11, кросплатформне тестування підтвердило абсолютно однакову поведінку системи в Telegram та вебвіджеті.

Тест 1. Перевірка коректності видачі файлів та роботи локальної навігації.

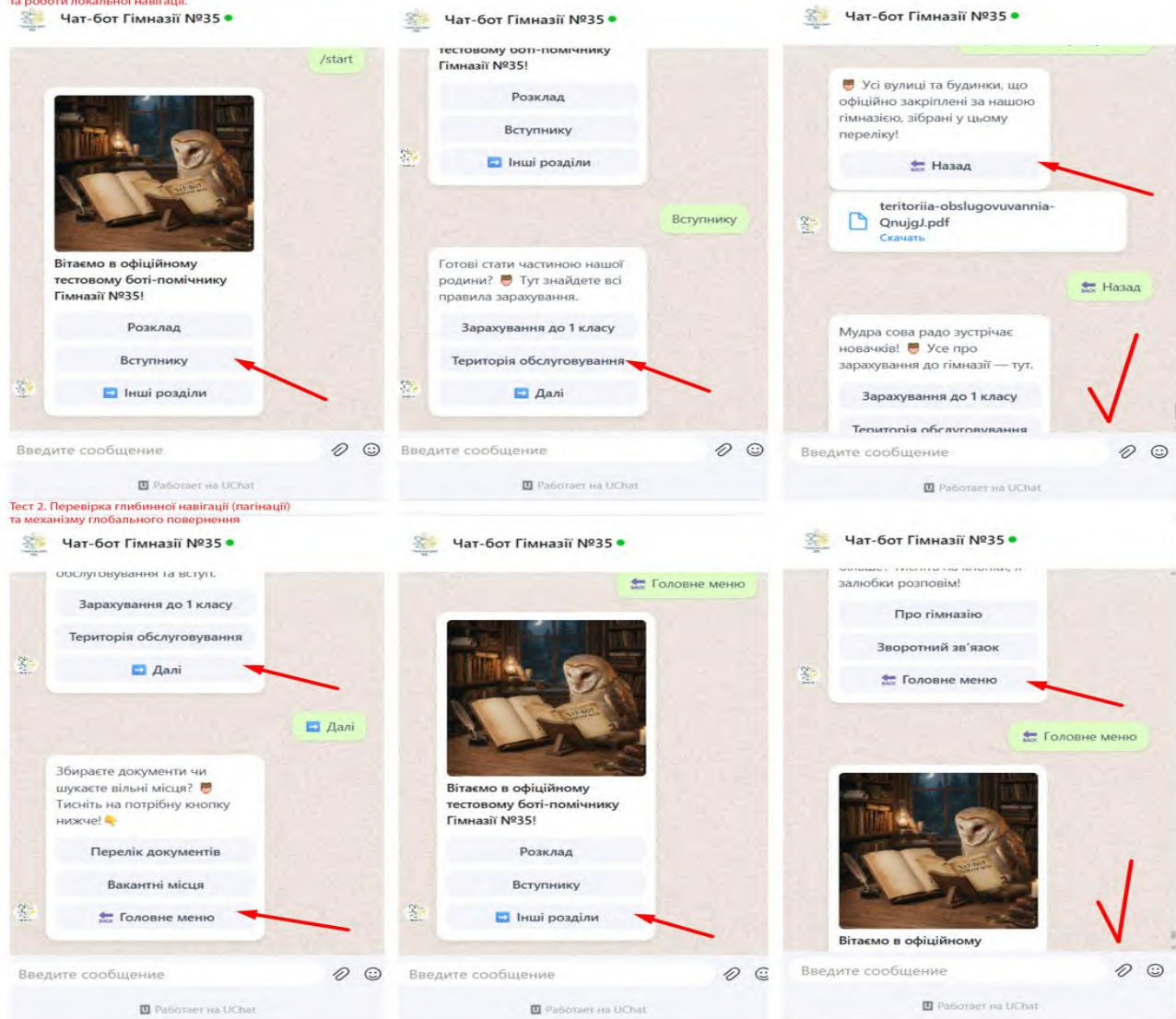


Рисунок 3.11 – Результати тестування видачі файлів та глибинної навігації у вебсередовищі Google Websites

Усі тестові сценарії, визначені у чек-лісті у Додатку Е, завершилися із статусом «Успіх». Жодного критичного збою, логічного розриву або невідповідності між очікуваним і фактичним результатами виявлено не було. Це свідчить про технічну готовність розробленої системи до повноцінної промислової експлуатації.

Таким чином, комплексне тестування підтвердило, що розроблений чат-бот Гімназії №35 відповідає функціональним вимогам, є стійким до нестандартних дій користувача та коректно працює на всіх цільових платформах і пристроях.

3.3 Розрахунок економічних показників та оцінка ефективності впровадження

Економічна ефективність впровадження інформаційних систем у бюджетних закладах освіти оцінюється через дві складові: пряму та непряму. У даному підрозділі обидві складові розглядаються комплексно.

Витрати на реалізацію проєкту поділяються на одноразові та постійні. Розробка чат-бота здійснювалася з використанням безкоштовного Telegram Bot API та платформи UChat, тому витрати на ліцензії та хостинг відсутні. Вартість трудовитрат визначається виходячи із середньої погодинної ставки ІТ-фахівця початкового рівня в Україні станом на 2025 рік – 150 грн/год. Загальна сума визначається за формулою (3.1):

$$V_{\text{розр}} = T \times \Gamma, \quad (3.1)$$

де $V_{\text{розр}}$ – витрати на розробку у гривнях;

T – обсяг трудовитрат розробника у годинах;

Γ – погодинна ставка розробника, грн/год.

Підставивши зазначені вище значення, отримаємо наступні показники:

$$V_{\text{розр}} = 320 \times 150 = 48000 \text{ грн.}$$

Щомісячна операційна вартість підписки UChat Business Plan становить 15 доларів США. За курсом НБУ (1 долар США = 42 грн) річна операційна вартість визначається за формулою (3.2):

$$V_{\text{рік}} = C_{\text{долар}} \times K \times 12, \quad (3.2)$$

де $V_{\text{рік}}$ – річні операційні витрати, грн/рік;

$C_{\text{долар}}$ – вартість підписки, долар США/міс;

K – курс долара США до гривні, грн/долар США.

Підставивши зазначені вище значення, отримаємо наступні показники:

$$B_{\text{рік}} = 15 \times 42 \times 12 = 7560 \text{ грн/рік.}$$

Таблиця 3.1 – Зведені витрати на впровадження та обслуговування чат-бота

Стаття витрат	Місяць, грн	Рік, грн
Трудовитрати розробника (одноразово)	-	48000
Підписка UChat Business	630	7560
Хостинг (Telegram Bot API)	0	0
Разом операційних витрат на рік	630	7560

Основним джерелом прямого економічного ефекту є вивільнення робочого часу адміністративного персоналу, що витрачається на обробку типових повторюваних запитів. За даними адміністрації гімназії, на відповіді на однотипні питання батьків та абітурієнтів витрачається близько 10% робочого часу секретаря та заступника директора. Чат-бот автоматизує 70% таких запитів.

Погодинна ставка персоналу розраховується за формулою (3.3):

$$C_{\text{год}} = Z_{\text{міс}} / \Phi_{\text{год}}, \quad (3.3)$$

де $C_{\text{год}}$ – погодинна ставка, грн/год;

$Z_{\text{міс}}$ – середньомісячна заробітна плата, грн/міс;

$\Phi_{\text{год}}$ – фонд робочого часу на місяць (168 год/міс).

Підставивши зазначені вище значення, отримаємо наступні показники:

$$C_{\text{год}_c} = 14500 / 168 = 86,31 \text{ грн/год (секретар);}$$

$$C_{\text{год}_\text{зд}} = 18000 / 168 = 107,14 \text{ грн/год (заступник директора).}$$

Обсяг автоматизованого часу на місяць для кожного працівника, визначається за формулою (3.4):

$$T_{\text{авт}} = \Phi_{\text{год}} \times d_{\text{тип}} \times d_{\text{авт}}, \quad (3.4)$$

де $T_{\text{авт}}$ – автоматизований час, год/міс;
 $d_{\text{тип}}$ – частка часу на типові запити;
 $d_{\text{авт}}$ – частка автоматизації чат-ботом.

Підставивши зазначені вище значення, отримаємо наступні показники:

$$T_{\text{авт}} = 168 \times 0,10 \times 0,70 = 11,76 \text{ год/міс.}$$

Загальна щомісячна пряма економія фонду оплати праці визначається за формулою (3.5):

$$E_{\text{міс}} = (C_{\text{год}_c} + C_{\text{год}_зд}) \times T_{\text{авт}}, \quad (3.5)$$

де $E_{\text{міс}}$ – загальна щомісячна пряма економія фонду оплати праці, грн;
 $C_{\text{год}_c}$ – погодинна ставка, грн/год;
 $C_{\text{год}_зд}$ – погодинна ставка заступника;
 $T_{\text{авт}}$ – автоматизований час, год/міс.

Підставивши розраховані раніше ставки, отримаємо наступні показники:

$$E_{\text{міс}} = (86,31 + 107,14) \times 11,76 = 2275 \text{ грн/міс.}$$

Річна пряма економія фонду оплати праці визначається за формулою (3.6):

$$E_{\text{рік}} = E_{\text{міс}} \times 12, \quad (3.6)$$

де $E_{\text{рік}}$ – річна пряма економія фонду оплати праці, грн/рік;
 $E_{\text{міс}}$ – загальна щомісячна пряма економія фонду оплати праці, грн.

Підставивши значення щомісячної економії, отримаємо наступні показники:

$$E_{\text{рік}} = 2275 \times 12 = 27300 \text{ грн/рік.}$$

Чиста річна економія – це різниця між прямою економією фонду оплати праці та річними операційними витратами, що визначається за формулою (3.7):

$$E_{\text{чиста}} = E_{\text{рік}} - B_{\text{рік}}, \quad (3.7)$$

де $E_{\text{чиста}}$ – чиста річна економія, грн/рік;

$E_{\text{рік}}$ – річна пряма економія фонду оплати праці, грн/рік;

$V_{\text{рік}}$ – річні операційні витрати, грн/рік.

Підставивши отримані раніше значення, отримаємо наступні показники:

$$E_{\text{чиста}} = 27300 - 7560 = 19740 \text{ грн/рік.}$$

Строк окупності проєкту (консервативний сценарій, з урахуванням умовної вартості розробки) визначається за формулою (3.8):

$$T_{\text{ок}} = V_{\text{розр}} / E_{\text{чиста}}, \quad (3.8)$$

де $T_{\text{ок}}$ – строк окупності проєкту, роки;

$V_{\text{розр}}$ – витрати на розробку, грн;

$E_{\text{чиста}}$ – чиста річна економія, грн/рік.

Підставивши всі витрати консервативного сценарію, отримаємо наступні показники:

$$T_{\text{ок}} = 48000 + 7560 / 19740 \approx 2,8 \text{ роки.}$$

У реальному сценарії, де розробка здійснювалася в рамках кваліфікаційної роботи, строк окупності визначається за формулою (3.9):

$$T_{\text{ок_реал}} = V_{\text{рік}} / E_{\text{рік}}, \quad (3.9)$$

де $T_{\text{ок_реал}}$ – строк окупності в реальному сценарії, роки;

$V_{\text{рік}}$ – річні операційні витрати, грн/рік;

$E_{\text{рік}}$ – річна пряма економія фонду оплати праці, грн/рік.

За результатами підстановки визначених величин поточних операційних витрат отримаємо такі значення:

$$T_{\text{ок_реал}} = 7560 / 27300 \approx 0,28 \text{ роки} \approx 3,4 \text{ місяці.}$$

Коефіцієнт економічної ефективності визначається за формулою (3.10):

$$E_{\text{коеф}} = E_{\text{чиста}} / V_{\text{рік}}, \quad (3.10)$$

де $E_{\text{коеф}}$ – коефіцієнт економічної ефективності;

$E_{\text{чиста}}$ – чиста річна економія, грн/рік;

$V_{\text{рік}}$ – річні операційні витрати, грн/рік.

Підставивши фінальні значення економії та витрат, отримаємо наступні показники:

$$E_{\text{коеф}} = 19740 / 7560 \approx 2,61.$$

Значення $E_{\text{коеф}} = 2,61 > 1$ підтверджує економічну доцільність проекту: кожна вкладена в обслуговування системи гривня забезпечує 2,61 грн умовної економії.

Таблиця 3.2 – Зведені показники економічної ефективності впровадження чат-бота

Показник	Значення	Одиниця
Пряма річна економія ФОП	27300	грн/рік
Річні операційні витрати	7560	грн/рік
Чиста річна економія	19740	грн/рік
Одноразові витрати на розробку	48000	грн
Строк окупності (консервативний)	$\approx 2,8$	роки
Строк окупності (реальний)	$\approx 3,4$	місяці
Коефіцієнт економічної ефективності	2,61	-
Вивільнено людино-годин на рік (11,76 год/міс $\times$ 12 міс $\times$ 2 працівники)	282,24	год/рік

Окрім економічної вигоди, впровадження чат-бота має значний соціальний ефект, суттєво покращуючи якість інформаційного обслуговування учасників освітнього процесу.

Таблиця 3.3 – Порівняння показників інформаційного обслуговування до та після впровадження чат-бота

Критерій	До впровадження	Після впровадження
Доступність	45 год/тиж (Пн-Пт)	168 год/тиж (24/7)
Час відповіді	від 15 хв до кількох год	1-2 секунди
Паралельних звернень	1	Необмежено
Навантаження на персонал	10% робочого часу	$\sim 3\%$ (70% автоматизовано)
Канали доступу	Телефон, особисто	Telegram + Вебсайт
Вартість обслуговування	Витрати ФОП	630 грн/міс

Впровадження чат-бота забезпечило цілодобовий доступ до інформації та скоротило час очікування відповіді з кількох годин до 1-2 секунд. На відміну від телефону, система паралельно опрацьовує необмежену кількість звернень, що особливо важливо під час вступної кампанії. Робота сервісу через Telegram та веббраузери без реєстрації максимально знижує бар'єр входу для всіх категорій користувачів.

Отже, у третьому розділі кваліфікаційної роботи вирішено три взаємопов'язані практичні завдання, що забезпечили впровадження розробленої інформаційної системи. Реалізовано двоканальну інтеграцію чат-бота: система функціонує як Telegram-бот і вбудований вебвіджет на офіційному сайті Гімназії №35. Проведено комплексне тестування за чотирма напрямками: функціональне, UI/UX, негативне та кросплатформне. Усі 10 тестових сценаріїв завершилися зі статусом «Успіх», що підтверджує готовність системи до експлуатації. Також розраховано економічну ефективність впровадження: пряма річна економія фонду оплати праці становить 27300 грн, чиста річна економія – 19740 грн, коефіцієнт економічної ефективності – 2,61, а строк окупності – близько 3,4 місяця.

ВИСНОВКИ

У кваліфікаційній роботі вирішено науково-практичне завдання розроблення та впровадження інформаційної системи у вигляді детермінованого чат-бота для Гімназії №35 Полтавської міської ради. За результатами проведеного дослідження зроблено такі висновки.

У першому розділі на основі системного аналізу виявлено критичне перевантаження адміністрації гімназії однотипними запитами (до 70% робочого часу секретаріату). Для автоматизації комунікації обґрунтовано вибір детермінованої архітектури чат-бота, яка на відміну від AI-систем гарантує 100% точність подачі офіційної інформації. Як оптимальну платформу для комунального закладу обрано Telegram Bot API завдяки відсутності комерційних обмежень та широкому охопленню аудиторії.

У другому розділі обґрунтовано вибір платформи UChat, що дозволило реалізувати омніканальність без дублювання технічної підтримки. Сформовано інформаційну базу та спроектовано чотириблочну структуру діалогів. Успішно здійснено двоканальну інтеграцію: чат-бот функціонує в Telegram та як вбудований вебвіджет на сайті гімназії. Комплексне QA-тестування (10 сценаріїв за 4 напрямками) підтвердило повну функціональну відповідність системи.

У третьому розділі реалізовано двоканальну інтеграцію системи: чат-бот функціонує як у Telegram, так і у вигляді вбудованого веб-віджета на офіційному сайті гімназії через механізм омніканального каналу платформи UChat. Проведено комплексне тестування за чотирма напрямками (функціональне, UI/UX, негативне, кросплатформне); усі сценарії чек-листа завершилися зі статусом «Успіх». Розраховано економічну ефективність: чиста річна економія – 19740 грн, коефіцієнт економічної ефективності – 2,61, строк окупності у реальному сценарії – 3,4 місяці. Таким чином, поставлена мета досягнута: розроблена інформаційна система є повністю функціональною, технічно перевіреною та економічно доцільною.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Про схвалення Концепції реалізації державної політики у сфері реформування загальної середньої освіти «Нова українська школа» на період до 2029 року: розпорядження Кабінету Міністрів України від 14.12.2016 № 988-р. URL: <https://zakon.rada.gov.ua/laws/show/988-2016-%D1%80> (дата звернення: 21.05.2026).
2. Wut T.-m., Low M. P., Tam K.-w. Using chatbot in education – A systematic review and research agenda. *Social Sciences & Humanities Open*. 2026. Vol. 13. P. 102956. URL: <https://doi.org/10.1016/j.ssaho.2026.102956> (дата звернення: 21.05.2026).
3. Halvoník D., Kapusta J. Large Language Models and Rule-Based Approaches in Domain-Specific Communication. *IEEE Access*. 2024. P. 1. URL: <https://doi.org/10.1109/access.2024.3436902> (дата звернення: 21.05.2026).
4. Thorat S. A., Jadhav V. A Review on Implementation Issues of Rule-based Chatbot Systems. *SSRN Electronic Journal*. 2020. URL: <https://doi.org/10.2139/ssrn.3567047> (дата звернення: 21.05.2026).
5. What is a chatbot? SAP. URL: <https://bit.ly/4dWfb9x> (дата звернення: 21.05.2026).
6. Most popular global mobile messaging apps 2025. Statista. URL: <https://bit.ly/3RNoH6o> (дата звернення: 21.05.2026).
7. WhatsApp Business Platform Documentation. URL: <http://bit.ly/4arAGNI> (дата звернення: 21.05.2026).
8. Messenger Platform Documentation (Facebook Graph API). URL: <https://bit.ly/4amYC4o> (дата звернення: 21.05.2026).
9. Viber REST Bot API Documentation. URL: <https://bit.ly/4uBOPj8> (дата звернення: 21.05.2026).
10. Telegram Bot API Documentation. URL: <https://core.telegram.org/bots/api> (дата звернення: 21.05.2026).

11. Будуємо телеграм чат-бот на Java: від ідеї до деплою. Частина 1. DOU. URL: <https://dou.ua/forums/topic/38358/> (дата звернення: 21.05.2026).
12. Kapusta J. et al. Natural Language Processing. Nitra : Constantine the Philosopher University in Nitra, 2024. 261 p. URL: <https://doi.org/10.17846/fpvai-2024-74> (дата звернення: 21.05.2026).
13. Singh S. The role of speech technology in biometrics, forensics and man-machine interface. *International Journal of Electrical and Computer Engineering (IJECE)*. 2019. Vol. 9, no. 1. P. 281. URL: <https://doi.org/10.11591/ijece.v9i1.pp281-288> (дата звернення: 21.05.2026).
14. Переяславська С., Смагіна О. Вивчення методів обробки природної мови для створення інтелектуальних систем, таких як чат-боти та системи автоматичного перекладу. *Вісник Хмельницького національного університету. Технічні науки*. 2025. Т. 349, № 2. С. 100–108. URL: <https://doi.org/10.31891/2307-5732-2025-349-14> (дата звернення: 21.05.2026).
15. Троцько В. В. Методи штучного інтелекту: навчально-методичний і практичний посібник. Київ: Університет економіки та права «КРОК», 2020. 86 с. URL: <https://bit.ly/4dSjhzz> (дата звернення: 21.05.2026).
16. Doubleday S., Trott S., Feldman J. Processing Natural Language About Ongoing Actions. arXiv. 2016. 6 p. URL: <https://arxiv.org/abs/1607.06875v1> (дата звернення: 21.05.2026).
17. Qamar R., Ali Zardari B. Artificial Neural Networks: An Overview. *Mesopotamian Journal of Computer Science*. 2023. P. 130–139. URL: <https://doi.org/10.58496/mjcsc/2023/015> (дата звернення: 21.05.2026).
18. Dovhan O. Architectural originality of artificial neural networks for linguistic markers recognition. *Bulletin of Science and Education*. 2024. No. 7(25). URL: [https://doi.org/10.52058/2786-6165-2024-7\(25\)-17-36](https://doi.org/10.52058/2786-6165-2024-7(25)-17-36) (дата звернення: 21.05.2026).
19. Субботін С. О. Нейронні мережі: теорія та практика: навч. посіб. Житомир: Вид. О. О. Євенок, 2020. 184 с. URL: <https://bit.ly/4wHg7WF> (дата звернення: 21.05.2026).

20. Solomon E., Tilahun S. L. Rule based chatbot design methods: A review. *Journal of Computational Science and Data Analytics*. 2024. Vol. 01, no. 1. P. 75–84. URL: <https://doi.org/10.69660/jcsda.01012405> (дата звернення: 21.05.2026).
21. Rule-based Chatbots. OMQ Lexicon. URL: <https://bit.ly/4dLbEuL> (дата звернення: 21.05.2026).
22. Rule-Based Chatbots. Moin.ai Chatbot Wiki. URL: <https://bit.ly/4a9g1NX> (дата звернення: 21.05.2026).
23. Khan R., Das A. Build Better Chatbots. Berkeley, CA : Apress, 2018. URL: <https://doi.org/10.1007/978-1-4842-3111-1> (дата звернення: 21.05.2026).
24. UChat Introduction. Official UChat documentation. URL: <https://bit.ly/4uabwd6> (дата звернення: 21.05.2026).
25. Apa Itu Omni Channel Keuntungan Kerugian Dan Jebakan. Artatel. URL: <https://bit.ly/4o5CotA> (дата звернення: 21.05.2026).
26. Gerabot Features of the service for creating chat bots. Official Gerabot documentation URL: <https://gerabot.com/en/features>. (дата звернення: 21.05.2026).
27. Introduction to Flow XO. Official documentation Flow XO. URL: <https://support.flowxo.com/article/132-introduction> (дата звернення: 21.05.2026).
28. Dialogflow CX Demystified: Agents, Flows, Pages, and Routes for Scalable Bots. Medium. URL: <https://bit.ly/4fhocv7> (дата звернення: 21.05.2026).
29. Початок роботи з чат-ботами в SendPulse. Офіційна документація SendPulse. URL: <https://sendpulse.ua/knowledge-base/chatbot/chatbot-basics> (дата звернення: 21.05.2026).
30. Ларіонова Н. Б., Стрельцова Н. М. Інтегративний підхід: актуальність, сутність, особливості впровадження в умовах початкової школи: навчально-методичний посібник. Харків: «Друкарня Мадрид», 2018. 76 с. (дата звернення: 21.05.2026).
31. Що таке API: простими словами про складне. HOSTiQ. URL: <https://hostiq.ua/blog/ukr/what-is-api/> (дата звернення: 21.05.2026).

32. HTML Drag and Drop API. MDN Web Docs. URL: <https://bit.ly/4uWnUPd> (дата звернення: 21.05.2026).

33. What is a trigger in automation. Robomotion. 2026. URL: <https://robomotion.io/blog/what-is-a-trigger-in-automation> (дата звернення: 21.05.2026).

34. Юдінцов Д.М. Можливості чат-боту для закладу освіти. Збірник XXI щорічної студентської наукової конференції «Сучасні інформаційні технології та інноваційні методики в економіці, менеджменті та бізнесі» Полтавського державного аграрного університету.(квітень 2026 р., м. Полтава). Полтава ПДАУ, 2026 р. С. 59-61.

35. Юдінцов Д.М. Можливості чат-боту для закладу освіти. Збірник XXI щорічної студентської наукової конференції «Сучасні інформаційні технології та інноваційні методики в економіці, менеджменті та бізнесі» Полтавського державного аграрного університету.(квітень 2026 р., м. Полтава). Полтава ПДАУ, 2026 р. С. 59-61.