

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджментуКафедра ІнформатикиРівень вищої освіти перший (бакалаврський)Спеціальність 122 Комп'ютерні науки
(код і повна назва)Тип програми освітньо-професійнаОсвітня програма Інформатика
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« _____ » _____ 2026 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУздобувачеві Кириченку Кирилу Юрійовичу
(прізвище, ім'я, по батькові)1. Тема роботи Розроблення бота в Telegram для підбору відеоігор з використанням штучного інтелекту

затверджена наказом університету від 26 травня 2026 року № 435Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 27 червня 2026 р.

3. Вихідні дані до роботи науково-методична та науково-технічна література, матеріали конференцій, електронні ресурси мережі Інтернет, документація Telegram Bot API, OpenAI API, RAWG API та CheapShark API, засоби розроблення Python, бібліотека aiogram та система керування базами даних SQLite.

4. Перелік питань, що потрібно опрацювати в роботі _____

1. Аналіз існуючих рекомендаційних систем відеоігор та можливостей використання штучного інтелекту.

2. Проєктування структури та алгоритмів роботи Telegram-бота для взаємодії з користувачами та зовнішніми сервісами.

3. Розроблення, тестування та оцінювання ефективності Telegram-бота для підбору відеоігор на основі OpenAI API.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Архітектура Telegram-бота для підбору відеоігор із використанням штучного інтелекту, схема взаємодії користувача із системою, структура бази даних SQLite, інтерфейс головного меню Telegram-бота, результати пошуку інформації про відеоігри, результати перегляду повного опису та скріншотів ігор, результати роботи системи рекомендацій на основі OpenAI API, результати пошуку актуальних знижок на відеоігри, результати роботи списку бажаних ігор(Wishlist), результати роботи профілю користувача, результати формування тематичних добірок відеоігор за жанрами, результати роботи календаря релізів та пошуку дати виходу гри, результати тестування функціональних можливостей Telegram-бота.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	13.04.2026	
2	Аналіз завдання, підбір літератури	16.04.26-26.04.26	
3	Аналіз літератури з проблеми, що вирішується	26.04.26-01.05.26	
4	Аналіз методів рекомендаційних систем та технологій штучного інтелекту	01.05.26-10.05.26	
5	Формування кроків вирішення проблеми	11.05.26-20.05.26	
6	Програмна реалізація	05.05.26-31.05.26	
7	Аналіз отриманих результатів	01.06.26-05.06.26	
8	Оформлення пояснювальної записки	05.06.26-09.06.26	
9	Перевірка на нормоконтроль	10.06.26-19.06.26	
10	Перевірка на плагіат	11.06.26-30.06.26	
11	Рецензування	20.06.26-30.06.26	
12	Підготовка презентації та доповіді	20.06.26-30.06.26	
13	Занесення роботи в електронний архів	30.06.26-09.07.26	
14	Попередній захист кваліфікаційної роботи	30.06.26-09.07.26	

Дата видачі завдання 13 квітня 2026 р.

Здобувач _____
(підпис)

Керівник роботи _____ доц. Тітова О. В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 91 с., 9 табл., 33 рис., 31 джерело.

ВІДЕОІГРИ, РЕКОМЕНДАЦІЙНА СИСТЕМА, TELEGRAM-БОТ, ШТУЧНИЙ ІНТЕЛЕКТ, API, GPT, PYTHON.

Об'єктом роботи є розроблення Telegram-бота для надання інформації про відеоігри.

Метою роботи є створення Telegram-бота, який забезпечує отримання інформації про ігри, перегляд майбутніх релізів та формування персоналізованих рекомендацій на основі текстових запитів користувача з використанням моделі GPT.

Під час роботи використано методи штучного інтелекту та обробки природної мови, а також інтеграцію API.

Розроблено Telegram-бот, що забезпечує надання інформації про ігри та формування рекомендацій у діалоговому форматі.

Область застосування: ігрові сервіси, інформаційні системи, рекомендаційні системи.

VIDEO GAMES, RECOMMENDATION SYSTEM, TELEGRAM BOT, ARTIFICIAL INTELLIGENCE, API, GPT, PYTHON.

The object of this work is the development of a Telegram bot for providing information about video games.

The goal of this work is to create a Telegram bot that provides game information, shows upcoming releases, and generates personalized recommendations based on user queries using a GPT model.

The following methods were used: artificial intelligence and natural language processing methods, as well as API integration.

A Telegram bot has been developed that provides game information and recommendations in a conversational format.

Applications: gaming services, information systems, recommendation systems.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	7
Вступ.....	8
1 Аналіз предметної області та постановка задачі	10
1.1 Загальний огляд розвитку індустрії відеоігор у сучасному цифровому середовищі.....	10
1.2 Інформаційне перевантаження та складність вибору відеоігор.....	14
1.3 Особливості взаємодії користувача з інформаційними системами у сфері відеоігор.....	17
1.4 Використання чат-ботів як сучасного інструменту доступу до інформації	18
1.5 Використання штучного інтелекту та мовних моделей для формування рекомендацій у сфері відеоігор	20
1.6 Порівняльний аналіз існуючих інформаційних та рекомендаційних систем у сфері відеоігор	23
1.7 Постановка задачі	26
2 Математичне обґрунтування вибраних методів та особливості розроблення методу для рекомендацій відеоігор	28
2.1 Аналіз методів рекомендаційних систем	28
2.2 Використання моделей обробки природної мови для формування рекомендацій	33
2.3 Архітектура системи рекомендацій відеоігор.....	40
2.4 Реалізація системи та аналіз результатів.....	46
3 Реалізація задачі, тестування та аналіз результатів.....	53
3.1 Вибір засобів програмної реалізації.....	53
3.2 Створення та налаштування Telegram-бота	54
3.3 Реалізація пошуку та отримання інформації про відеоігри за допомогою RAWG API.....	58
3.4 Реалізація системи рекомендацій на основі OpenAI API.....	63

3.5	Реалізація підсистеми пошуку знижок на відеоігри за допомогою CheapShark API.....	68
3.6	Реалізація бази даних користувачів на основі SQLite.....	72
3.7	Тестування програмного продукту та аналіз результатів.....	77
3.8	Перспективи розвитку системи	85
Висновки		87
Перелік джерел посилання		90

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

III – штучний інтелект

API – Application Programming Interface (інтерфейс програмування застосунків)

Async processing – асинхронна обробка запитів у системі

Backend – серверна частина інформаційної системи

Embedding – векторне представлення текстових даних

GPT – Generative Pre-trained Transformer (генеративна попередньо навчена трансформерна модель)

HTTP – HyperText Transfer Protocol (протокол передачі даних)

IGDB – Internet Games Database (база даних відеоігор)

JSON – JavaScript Object Notation (формат обміну даними)

LLM – Large Language Model (велика мовна модель)

ML – Machine Learning (машинне навчання)

NLP – Natural Language Processing (обробка природної мови)

Prompt – текстовий запит користувача до мовної моделі

RAWG API – сервіс для отримання інформації про відеоігри

Recommendation System – рекомендаційна система для підбору контенту

REST API – архітектурний стиль взаємодії програмних систем через HTTP-запити

Telegram Bot – програмний бот у середовищі Telegram

ВСТУП

Сучасні інформаційні технології охоплюють широкий спектр напрямків, серед яких особливе місце займають системи обробки даних, рекомендаційні системи та інтелектуальні інтерфейси взаємодії з користувачем. У контексті розвитку цифрового середовища важливого значення набуває забезпечення ефективного доступу до інформації, її аналізу та персоналізованого представлення.

Рекомендаційні системи є одним із ключових інструментів, що дозволяють користувачам орієнтуватися у великих обсягах даних. Вони застосовуються у різних сферах, зокрема у електронній комерції, медіасервісах та індустрії відеоігор. Основним завданням таких систем є підбір релевантного контенту відповідно до потреб та вподобань користувача.

Особливого значення рекомендаційні системи набувають у сфері відеоігор, де користувач стикається з великою кількістю ігрового контенту. Це включає як нові релізи, так і вже існуючі ігри, що підтримуються оновленнями. У таких умовах процес вибору гри стає складним і потребує врахування багатьох факторів, таких як жанр, платформа, рейтинг, технічні характеристики та індивідуальні вподобання користувача.

Однією з сучасних тенденцій розвитку інформаційних систем є використання чат-орієнтованих інтерфейсів. Чат-боти дозволяють реалізувати природну форму взаємодії, при якій користувач може формулювати запити у довільній формі та отримувати відповіді у вигляді діалогу. Такий підхід значно спрощує процес пошуку інформації та підвищує зручність використання систем.

Важливим напрямком розвитку є також застосування технологій штучного інтелекту, зокрема мовних моделей, які здатні аналізувати текстові запити, враховувати контекст та генерувати змістовні відповіді. Це дозволяє створювати більш гнучкі та адаптивні системи, здатні працювати з неструктурованими даними.

Актуальність роботи полягає у необхідності розробки ефективних інструментів доступу до інформації про відеоігри та формування персоналізованих рекомендацій. Зростання обсягів ігрового контенту та складність його вибору зумовлюють потребу у створенні систем, які поєднують можливості рекомендаційних алгоритмів, обробки природної мови та зручних інтерфейсів взаємодії.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Загальний огляд розвитку індустрії відеоігор у сучасному цифровому середовищі

Сучасний етап розвитку інформаційних технологій характеризується активним впровадженням цифрових сервісів, систем обробки даних та інтелектуальних алгоритмів у різні сфери діяльності людини. Особливо інтенсивно цифровізація відбувається у сфері розваг, де значного поширення набули мультимедійні платформи, потокові сервіси та рекомендаційні системи. Одним із найбільш технологічно розвинених напрямів цієї галузі є індустрія відеоігор, яка поєднує сучасні досягнення програмної інженерії, мережевих технологій та штучного інтелекту. У роботах, присвячених інтелектуальному аналізу даних [1], підкреслюється важливість використання адаптивних методів оброблення інформації для підвищення ефективності сучасних цифрових систем.

Розвиток індустрії відеоігор значною мірою пов'язаний із поширенням високошвидкісного доступу до мережі Інтернет, розвитком мобільних платформ та цифрових сервісів розповсюдження контенту. Сучасні користувачі мають можливість взаємодіяти з ігровими платформами, ігровими сервісами за допомогою персональних комп'ютерів, смартфонів та ігрових консолей. Це призводить до постійного збільшення кількості користувачів та обсягів цифрового контенту. Важливою особливістю сучасного ринку є суттєве збільшення кількості ігрових продуктів та цифрових сервісів. Якщо раніше основну частину ринку представляли великі компанії-розробники, то сьогодні важливу роль відіграють незалежні студії та окремі розробники програмного забезпечення. Використання сучасних засобів програмної розробки та цифрових платформ дистрибуції значно спростило процес створення програмних продуктів і прискорило їх вихід на ринок.

Разом із перевагами така різноманітність створює проблему інформаційного перевантаження. Користувач змушений аналізувати значний обсяг інформації, порівнювати характеристики відеоігор, оцінювати рейтинги та враховувати власні вподобання. Подібні проблеми характерні для більшості сучасних інформаційних систем, що функціонують в умовах великих масивів даних. Методи інтелектуального аналізу інформації, алгоритми кластеризації та системи підтримки прийняття рішень дозволяють автоматизувати процес оброблення інформації та підвищити ефективність цифрових сервісів [2].

Сучасна індустрія відеоігор супроводжується активним розвитком інформаційних платформ, до яких належать цифрові магазини, спеціалізовані вебресурси, стрімінгові сервіси та рекомендаційні системи. Такі платформи забезпечують доступ до значного обсягу інформації про відеоігри, включаючи жанри, технічні характеристики, рейтинги та відгуки користувачів. У результаті виникає потреба у використанні інтелектуальних засобів аналізу даних та персоналізації інформації (табл. 1.1).

Таблиця 1.1 – Основні технології сучасних цифрових сервісів у сфері відеоігор

Технологія	Призначення	Приклад використання
1	2	3
Обробка природної мови	Аналіз текстових запитів користувачів	Обробка повідомлень у чат-ботах
Machine Learning	Аналіз поведінки користувачів	Формування персоналізованих рекомендацій
Інтерфейс програмування застосунків	Інтеграція із зовнішніми сервісами	RAWG API, IGDB
Великі мовні моделі	Генерація текстових відповідей	GPT-моделі

Продовження таблиці 1.1

1	2	3
Cloud Services	Масштабування системи	Хмарна обробка даних
Recommendation Algorithms	Формування рекомендацій	Підбір відеоігор

Важливим фактором розвитку цифрових сервісів є глобалізація інформаційного простору та підтримка багатомовного середовища. Сучасні інформаційні системи повинні забезпечувати можливість роботи з великою кількістю текстових даних різними мовами, що формує необхідність використання універсальних інструментів для автоматизованого пошуку, аналізу та систематизації інформації. Особливого значення набувають технології обробки природної мови, які дозволяють працювати з неструктурованими текстовими запитами користувачів. Трансформерні архітектури та моделі обробки природної мови (Natural Language Processing, NLP) [3] активно використовуються у сучасних цифрових сервісах для аналізу текстових даних та підвищення якості взаємодії між користувачем і системою. Одним із перспективних напрямів розвитку сучасних інформаційних систем є використання технологій штучного інтелекту та великих мовних моделей (Large Language Models, LLM). Сучасні LLM-моделі забезпечують можливість аналізу текстових запитів, генерації відповідей та формування персоналізованих рекомендацій відповідно до потреб користувача. Значна увага приділяється використанню моделей Generative Pre-trained Transformer (GPT) та комбінованих підходів штучного інтелекту [4] для створення інтелектуальних цифрових сервісів, здатних працювати з неструктурованими даними та підтримувати діалогову взаємодію. Завдяки використанню механізмів глибокого навчання такі моделі можуть аналізувати контекст повідомлень, визначати наміри користувача та формувати більш релевантні результати пошуку. Більше того, великі мовні

моделі дозволяють автоматизувати процес обробки інформації та забезпечувати більш природну взаємодію між користувачем і системою (рис. 1.1).

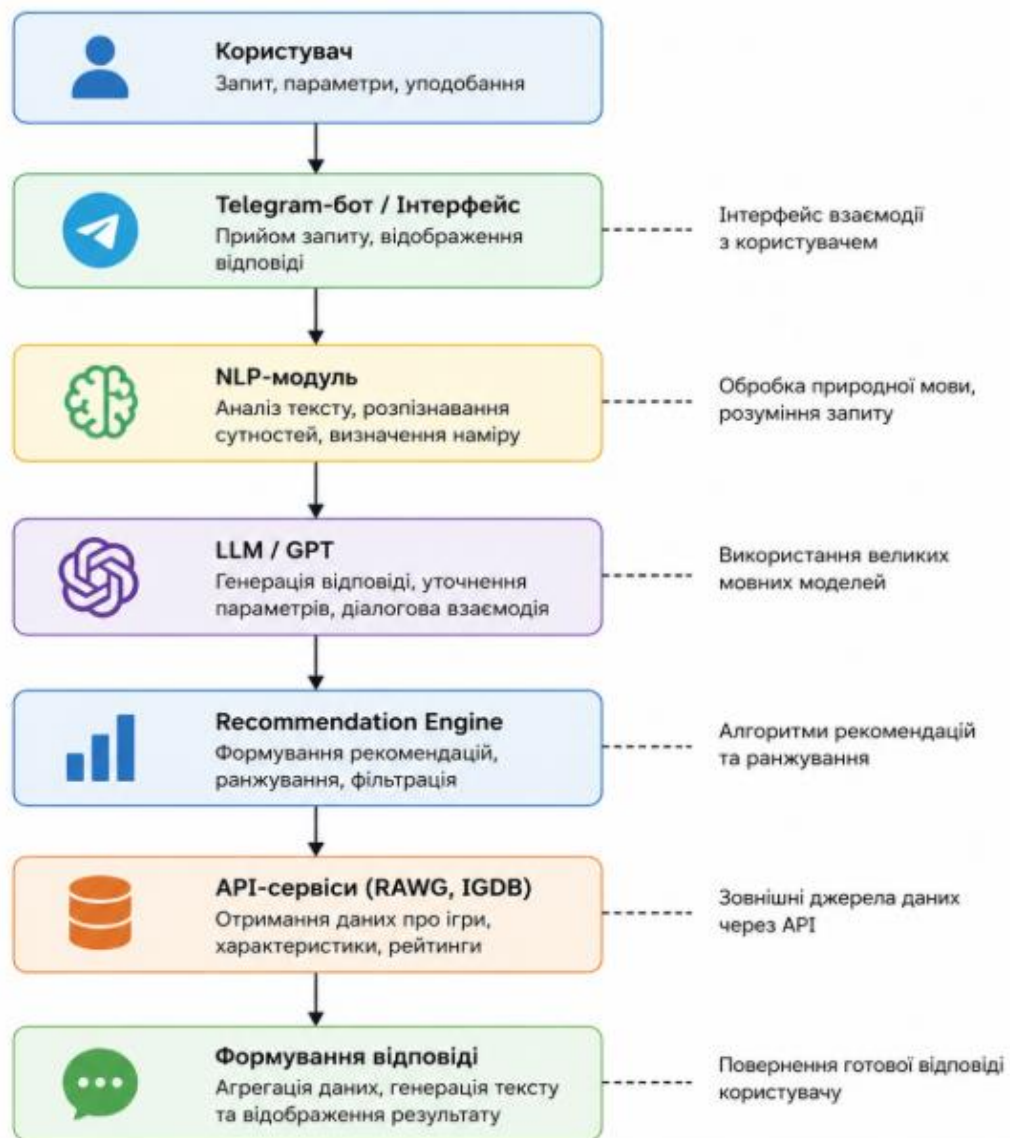


Рисунок 1.1 – Архітектура сучасної інтелектуальної системи рекомендацій відеоігор

Рисунок 1.1 складено автором на основі [3, 4]. У сучасних інформаційних системах важливу роль відіграє використання API-орієнтованого підходу та хмарних технологій. Використання інтерфейсу програмування застосунків (Application Programming Interface, API) забезпечує інтеграцію із зовнішніми сервісами та отримання актуальних даних у режимі реального часу. Крім того, застосування моделей машинного навчання, механізмів асинхронної обробки

запитів та хмарної інфраструктури дозволяє підвищити продуктивність і масштабованість програмного забезпечення [5].

Значного поширення також набувають рекомендаційні системи, які використовують алгоритми аналізу поведінки користувачів та методи машинного навчання для формування персоналізованого контенту. Такі системи дозволяють визначати вподобання користувачів, аналізувати історію взаємодії та формувати релевантні рекомендації відповідно до заданих параметрів. У сучасних цифрових сервісах рекомендаційні алгоритми дедалі частіше поєднуються з NLP-моделями та технологіями штучного інтелекту, що дозволяє враховувати не лише структуровані параметри, а й зміст текстових запитів користувача.

На основі цього, сучасний стан розвитку цифрових сервісів у сфері відеоігор характеризується активним використанням технологій штучного інтелекту, систем аналізу даних та рекомендаційних алгоритмів. Постійне збільшення обсягів інформації формує необхідність створення інтелектуальних інформаційних систем, здатних автоматизувати процес пошуку, аналізу та персоналізації цифрового контенту.

1.2 Інформаційне перевантаження та складність вибору відеоігор

Однією з характерних особливостей сучасного цифрового середовища є постійне збільшення обсягів інформації, доступної користувачеві. У сфері відеоігор це проявляється у стрімкому зростанні кількості ігрових продуктів, інформаційних платформ та мультимедійного контенту. Користувач отримує доступ до великої кількості даних, серед яких описи ігор, рейтинги, відгуки та технічні характеристики. У сучасних дослідженнях з інтелектуального аналізу даних [6] підкреслюється, що ефективне опрацювання великих обсягів інформації потребує використання адаптивних методів аналізу та кластеризації даних.

Розвиток цифрових платформ значно спростив доступ до ігрового контенту, однак одночасно ускладнив процес вибору. Сучасний користувач змушений працювати з великою кількістю альтернатив, що відрізняються за жанром, платформою та іншими характеристиками. Для прийняття рішення користувач повинен враховувати технічні вимоги, оцінки інших користувачів та власні вподобання. Сучасні рекомендаційні системи використовують алгоритми машинного навчання та персоналізації контенту [7], що дозволяє автоматизувати процес аналізу користувацьких вподобань і формувати релевантні рекомендації. Проте важливою проблемою залишається нечіткість користувацьких запитів. У багатьох випадках користувач не може точно сформулювати критерії пошуку та використовує описові або емоційні характеристики, що ускладнює застосування традиційних пошукових механізмів.

Трансформерні архітектури та NLP-моделі [8] дозволяють ефективніше працювати з неструктурованими текстовими запитам та враховувати контекст взаємодії користувача із системою. Використання механізмів уваги та контекстного аналізу забезпечує можливість визначення ключових характеристик запиту навіть у випадках нечіткого або неповного формулювання. До того ж, великі мовні моделі [9] забезпечують можливість аналізу поведінкових характеристик користувача та формування персоналізованого контенту відповідно до його вподобань. Такі моделі здатні адаптувати результати пошуку до індивідуальних потреб користувача, що дозволяє підвищити релевантність рекомендацій та покращити якість взаємодії з цифровими сервісами. У результаті використання сучасних NLP-технологій та великих мовних моделей сприяє підвищенню ефективності інформаційних систем і забезпечує більш природну взаємодію між користувачем та цифровим сервісом. У той же час, інтеграція таких технологій у сучасні рекомендаційні системи дозволяє автоматизувати процес аналізу запитів, зменшити інформаційне навантаження на користувача та забезпечити швидший доступ до релевантного цифрового контенту. Це особливо важливо в умовах постійного збільшення обсягів інформації (рис. 1.2).

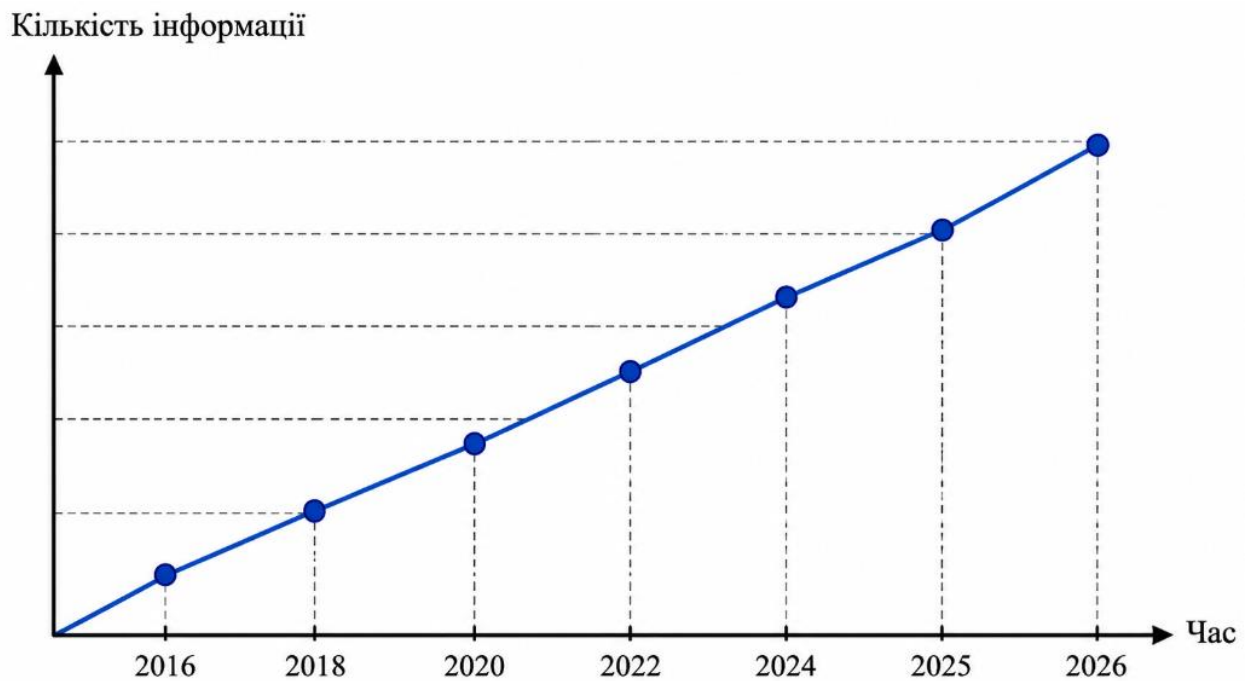


Рисунок 1.2 – Зростання обсягів цифрового контенту та інформаційного навантаження у сфері відеоігор

Окремої уваги потребує проблема достовірності інформації. Дані про відеоігри можуть надходити з великої кількості незалежних джерел, які відрізняються рівнем актуальності та надійності. У результаті користувач змушений додатково аналізувати отриману інформацію, порівнювати різні джерела та перевіряти їх достовірність.

Окрім технічних аспектів, інформаційне перевантаження має також психологічний характер. Надмірна кількість доступних варіантів ускладнює процес прийняття рішення та може негативно впливати на користувацький досвід, особливо в умовах постійного оновлення контенту.

Отже, проблема інформаційного перевантаження у сфері відеоігор охоплює технічні та поведінкові аспекти. Використання рекомендаційних алгоритмів, методів машинного навчання та систем обробки природної мови дозволяє підвищити ефективність пошуку інформації та забезпечити персоналізацію цифрового контенту.

1.3 Особливості взаємодії користувача з інформаційними системами у сфері відеоігор

У сучасному цифровому середовищі взаємодія користувача з інформаційними системами відіграє важливу роль у процесі отримання, оброблення та використання даних. У сфері відеоігор ця взаємодія набуває особливого значення через значний обсяг інформації, який потребує швидкого та ефективного аналізу. Від якості організації взаємодії залежить не лише швидкість отримання результату, а й загальний рівень задоволеності користувача. У сучасних дослідженнях цифрових сервісів підкреслюється важливість побудови адаптивних та орієнтованих на користувача інформаційних систем [2].

Важливою характеристикою є різноманітність користувацьких сценаріїв. В одному випадку користувач може шукати конкретну гру за назвою, а в іншому — отримувати рекомендації на основі загального опису або власних вподобань. Це означає, що система повинна підтримувати як структуровані, так і неструктуровані запити. Сучасні моделі штучного інтелекту (ШІ) та системи аналізу даних дозволяють ефективніше працювати з різними типами користувацьких сценаріїв [10]. Особливу увагу слід приділити неструктурованим запитам, які формулюються у довільній формі та можуть містити описові або емоційні характеристики. Подібні запити значно ускладнюють оброблення інформації традиційними системами пошуку. У зв'язку з цим особливого значення набувають технології обробки природної мови та трансформерні архітектури [11], які забезпечують можливість аналізу контексту запиту та формування більш релевантних результатів. Ще однією важливою особливістю є очікування швидкої відповіді. Сучасні користувачі звикли до високої швидкості роботи цифрових сервісів, тому інформаційна система повинна забезпечувати ефективну обробку даних та мінімальний час відгуку. Ба більше, важливим фактором є персоналізація. Користувачі очікують, що система буде враховувати їхні вподобання та пропонувати релевантний

контент. У сучасних роботах, присвячених великим мовним моделям [12], розглядаються механізми адаптації цифрових сервісів до поведінки користувача та формування персоналізованих рекомендацій.

Важливим аспектом є інтерактивність системи. Користувач очікує не лише отримання інформації, але й можливості взаємодії із системою у процесі пошуку. Це передбачає використання діалогової моделі взаємодії та механізмів уточнення запитів. У сучасних цифрових сервісах дедалі частіше використовуються чат-орієнтовані системи та ШІ-асистенти [4], які забезпечують більш природну форму взаємодії між користувачем і системою.

Крім того, важливою є зрозумілість відповіді системи. Вона повинна бути структурованою, лаконічною та легкою для сприйняття. Для цього сучасні інформаційні системи використовують алгоритми автоматизованого структурування та фільтрації даних [7].

Зважаючи на це, ефективна інформаційна система має бути орієнтованою на користувача та його поведінкові особливості. Врахування цих чинників спрощує пошук інформації, удосконалює механіку взаємодії та забезпечує високий рівень задоволеності користувача.

1.4 Використання чат-ботів як сучасного інструменту доступу до інформації

Глобальне збільшення обсягів інформації та ускладнення програмної архітектури сучасних систем створюють серйозний виклик для користувачів. Саме тому пріоритетним завданням стає спрощення їхньої взаємодії з цифровими сервісами через впровадження ергономічних інтерфейсів, що відкривають легкий доступ до можливостей системи. Одним із найбільш перспективних рішень у цьому напрямку є використання чат-ботів, які дозволяють організувати взаємодію у формі природного діалогу. У сучасних

дослідженнях інформаційних систем [7] підкреслюється, що використання інтелектуальних механізмів взаємодії підвищує ефективність цифрових сервісів.

Чат-боти являють собою програмні системи, здатні приймати, аналізувати та обробляти текстові або голосові запити користувача, а також формувати відповідь у зручному форматі. Використання природної мови значно спрощує взаємодію та робить її більш інтуїтивною. Сучасні системи обробки природної мови та великі мовні моделі [13] дозволяють підвищити точність аналізу запитів і забезпечити більш природну форму комунікації між користувачем та системою.

На відміну від традиційних графічних інтерфейсів, чат-боти дозволяють формулювати запити у довільній формі. Це особливо важливо у випадках, коли користувач не має чітко визначеної мети або використовує описові характеристики. Використання NLP-технологій та трансформерних моделей [14] дозволяє системі аналізувати контекст повідомлення та точніше визначати наміри користувача. Однією з ключових переваг чат-ботів є інтеграція з популярними месенджерами. Особливу популярність отримав месенджер Telegram, який надає широкі можливості для розробки інтерактивних систем завдяки відкритому API та підтримці інтеграції із зовнішніми сервісами (табл. 1.2).

Таблиця 1.2 – Основні переваги чат-ботів

Перевага	Характеристика
Інтерактивність	Підтримка діалогової взаємодії
Персоналізація	Формування рекомендацій відповідно до вподобань користувача
Доступність	Робота на різних пристроях
Масштабованість	Одночасна обробка великої кількості запитів

У сфері відеоігор чат-боти можуть виконувати інформаційні та рекомендаційні функції, надаючи відомості про ігри, жанри, платформи та рейтинги. Аналіз поведінкових характеристик користувача та використання алгоритмів машинного навчання [12] дозволяють формувати персоналізовані рекомендації та підвищувати релевантність результатів. Важливою особливістю чат-ботів є підтримка діалогової моделі взаємодії. Система може уточнювати запити користувача, аналізувати контекст та формувати більш точні результати. Цей підхід, використання великих мовних моделей [15] забезпечує можливість генерації відповідей природною мовою та підтримки більш гнучкого діалогу.

Тоді, чат-боти є ефективним інструментом організації доступу до інформації у сфері відеоігор. Використання Telegram-ботів, NLP-технологій та моделей штучного інтелекту дозволяє забезпечити зручну взаємодію користувача із системою, підвищити швидкість пошуку інформації та реалізувати персоналізований підхід до формування рекомендацій.

1.5 Використання штучного інтелекту та мовних моделей для формування рекомендацій у сфері відеоігор

Сучасні рекомендаційні системи активно використовують технології штучного інтелекту для автоматизації процесу пошуку та фільтрації інформації. У сучасних роботах з аналізу даних [6] підкреслюється важливість використання інтелектуальних підходів для підвищення ефективності інформаційних систем.

Традиційні рекомендаційні алгоритми базуються на аналізі поведінки користувача, історії переглядів або попередніх виборів. Проте такі підходи мають ряд обмежень, зокрема проблему «холодного старту» та недостатню ефективність при роботі з неструктурованими даними.

Сучасним рішенням цієї проблеми є використання моделей обробки природної мови та великих мовних моделей [13], які дозволяють аналізувати текстові запити, враховувати контекст взаємодії та формувати більш релевантні

рекомендації. Трансформерні архітектури [14] забезпечують можливість обробки складних текстових конструкцій і визначення семантичних зв'язків між словами та фразами.

Важливою перевагою ІІІ-моделей є можливість роботи з комбінованими запитами. Користувач може одночасно враховувати жанр гри, платформу, стиль проходження або рівень складності, а система аналізує всі критерії комплексно [12]. На відміну від традиційних алгоритмів пошуку, сучасні моделі здатні враховувати взаємозв'язки між різними параметрами запиту та визначати найбільш релевантні результати. Це дозволяє формувати більш точні та персоналізовані рекомендації навіть у випадках складних або нечітко сформульованих запитів.

До всього, сучасні мовні моделі дозволяють генерувати пояснення до рекомендацій, що підвищує довіру користувача до системи. ІІІ-модель може не лише запропонувати певну відеогру, але й пояснити, які саме характеристики стали причиною вибору. Наприклад, система може враховувати схожість жанру, наявність кооперативного режиму, стиль проходження або сюжетну складову гри. Це робить взаємодію із системою більш зрозумілою та прозорою для користувача.

Завдяки використанню механізмів машинного навчання та контекстного аналізу ІІІ-системи можуть адаптувати рекомендації відповідно до поведінкових характеристик користувача та поступово покращувати якість взаємодії із цифровим сервісом. Аналіз попередніх запитів, історії взаємодії та вподобань користувача дозволяє системі формувати більш релевантний та персоналізований контент. Крім того, сучасні ІІІ-моделі здатні прогнозувати можливі інтереси користувача та пропонувати контент, який найбільше відповідає його потребам, що сприяє підвищенню точності рекомендацій та покращенню користувацького досвіду.

На рисунку 1.3 представлено загальну схему роботи ІІІ-рекомендаційної системи. На першому етапі користувач формує текстовий запит, який надходить до системи у вигляді `prompt`-повідомлення. Далі виконується NLP-аналіз, під час

якого система обробляє текст, визначає ключові характеристики запиту та аналізує його контекст. Після цього запит передається до Transformer/LLM-моделі, яка виконує семантичний аналіз інформації та формує структуроване представлення запиту (рис. 1.3).

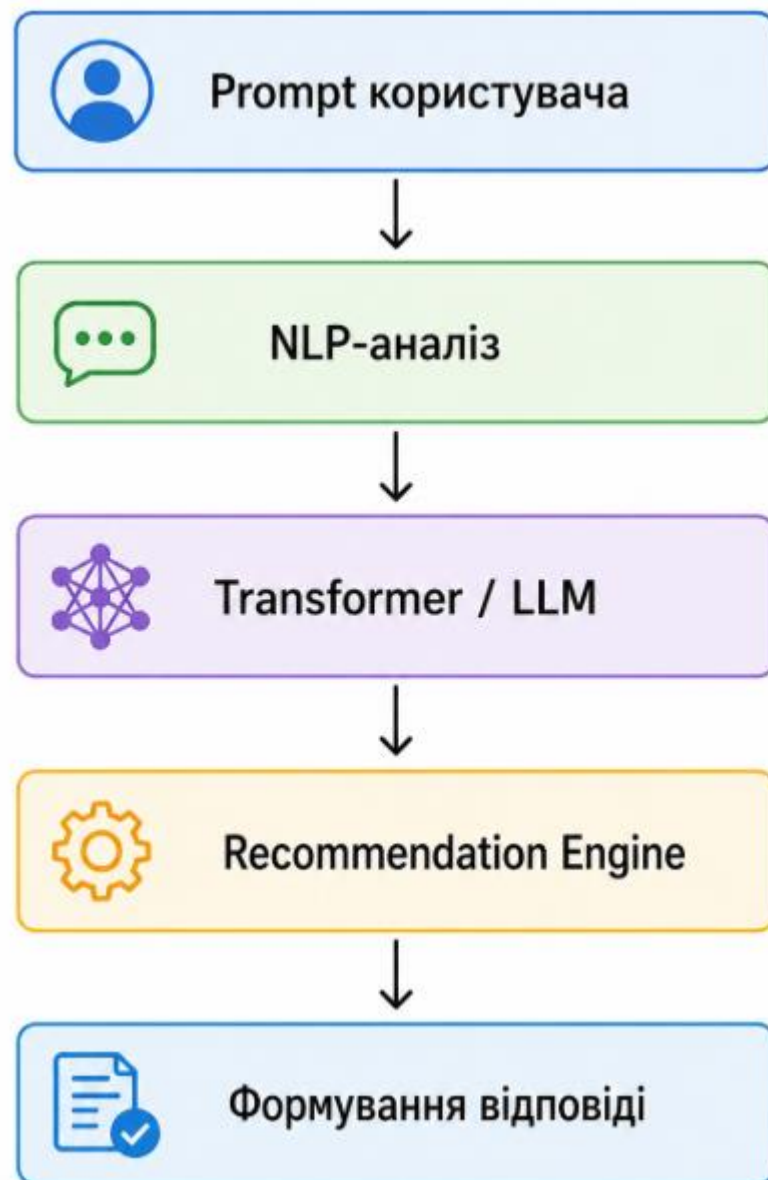


Рисунок 1.3 – Схема роботи ІІІ-рекомендаційної системи

На основі отриманих даних Recommendation Engine здійснює підбір найбільш релевантних рекомендацій відповідно до вподобань користувача та параметрів пошуку. Завершальним етапом є формування текстової відповіді, яка повертається користувачу у зручному та зрозумілому форматі. Ще однією

важливою особливістю є адаптивність системи. Моделі штучного інтелекту можуть аналізувати поведінкові характеристики користувача та поступово покращувати якість рекомендацій відповідно до його вподобань. Використання великих мовних моделей [15] також дозволяє реалізувати природний діалоговий інтерфейс та підтримувати більш гнучку взаємодію між користувачем і системою. Водночас слід враховувати і певні обмеження використання штучного інтелекту, зокрема можливі помилки у інтерпретації запитів, залежність від якості навчальних даних та необхідність регулярного оновлення моделей.

З урахуванням наведеного, використання технологій штучного інтелекту є перспективним напрямком розвитку рекомендаційних систем у сфері відеоігор. Поєднання NLP, великих мовних моделей та рекомендаційних алгоритмів дозволяє підвищити точність рекомендацій, забезпечити персоналізацію та покращити якість взаємодії користувача із цифровим сервісом.

1.6 Порівняльний аналіз існуючих інформаційних та рекомендаційних систем у сфері відеоігор

Індустрія відеоігор пропонує значну кількість інформаційних та рекомендаційних платформ, які відрізняються функціональними можливостями, рівнем персоналізації та способами взаємодії з користувачем. Аналіз сучасних рішень дозволяє визначити їх сильні сторони та виявити основні обмеження, що є важливим етапом при розробці інтелектуальних інформаційних систем. Однією з найбільш поширених категорій є спеціалізовані інформаційні платформи, які надають структуровані відомості про відеоігри. Такі системи містять інформацію про жанри, розробників, дати релізу, рейтинги та технічні характеристики [10]. Їх основною перевагою є повнота та структурованість даних, що дозволяє користувачу отримати детальну інформацію про конкретний ігровий продукт.

Іншою важливою категорією є цифрові магазини, які поєднують функції продажу ігор та рекомендаційних сервісів. Вони використовують алгоритми

аналізу поведінки користувача для формування персоналізованих рекомендацій. У сучасних роботах, присвячених великим мовним моделям та рекомендаційним системам [16], зазначається, що використання ШІ-технологій дозволяє підвищити релевантність рекомендацій та покращити якість взаємодії з користувачем. Окрему категорію становлять оглядові ресурси та інформаційні портали, які надають аналітичні матеріали, рецензії та рейтинги.

Сучасні рекомендаційні системи дедалі активніше використовують технології штучного інтелекту, алгоритми машинного навчання та NLP-моделі [17]. Це дозволяє аналізувати не лише структуровані параметри, а й текстові запити користувача, враховувати контекст взаємодії та формувати більш персоналізовані результати.

Важливим напрямом розвитку сучасних інформаційних систем є використання чат-ботів та ШІ-асистентів. Такі системи забезпечують діалогову модель взаємодії та дозволяють формулювати запити у довільній формі. Використання великих мовних моделей [18] забезпечує можливість аналізу складних текстових конструкцій та генерації відповідей природною мовою. Сучасні ШІ-асистенти здатні враховувати контекст взаємодії користувача із системою, аналізувати попередні запити та адаптувати результати відповідно до індивідуальних потреб користувача. Це дозволяє підвищити точність пошуку інформації та забезпечити більш персоналізовану взаємодію із цифровим сервісом. Важливою перевагою таких систем є можливість підтримки інтерактивного діалогу, під час якого користувач може уточнювати критерії пошуку, змінювати параметри запиту або отримувати додаткові рекомендації. У результаті використання ШІ-асистентів сприяє покращенню користувацького досвіду, підвищенню ефективності інформаційних систем та спрощенню доступу до необхідної інформації.

На рисунку 1.4 (розроблено автором на основі [16–19]) представлено порівняння сучасних платформ у сфері відеоігор за рівнем функціональності та можливостями взаємодії з користувачем. Інформаційні платформи забезпечують доступ до структурованих даних про відеоігри, однак мають обмежений рівень

персоналізації. Цифрові магазини поєднують функції дистрибуції контенту та рекомендаційних систем, що дозволяє частково адаптувати результати відповідно до вподобань користувача. Оглядові ресурси надають аналітичні матеріали та рецензії, проте здебільшого не підтримують інтерактивну взаємодію (рис. 1.4).

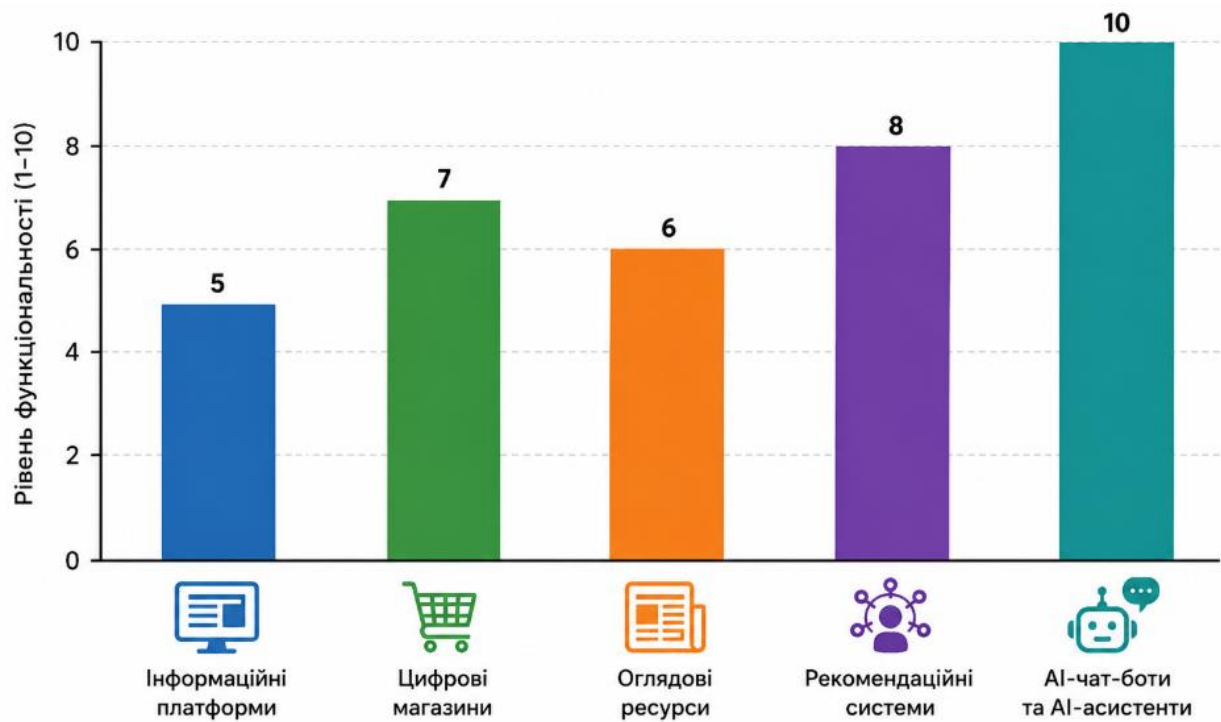


Рисунок 1.4 – Порівняння сучасних платформ у сфері відеоігор

Рекомендаційні системи демонструють вищий рівень функціональності завдяки використанню алгоритмів машинного навчання та аналізу поведінкових характеристик користувача. Найвищий рівень функціональності мають ШІ-чат-боти та ШІ-асистенти, оскільки вони поєднують можливості NLP, великих мовних моделей та діалогової взаємодії.

У зв'язку з цим перспективним напрямом є створення гібридних інформаційних систем, які поєднують рекомендаційні алгоритми, NLP-технології, великі мовні моделі та чат-орієнтовані інтерфейси. Інтеграція таких технологій дозволяє підвищити ефективність обробки інформації, забезпечити

персоналізацію та покращити якість взаємодії користувача із цифровими сервісами [19].

1.7 Постановка задачі

У результаті проведеного аналізу предметної області, а також дослідження існуючих інформаційних і рекомендаційних систем у сфері відеоігор було встановлено, що сучасні користувачі стикаються з проблемою інформаційного перевантаження, складністю формулювання запитів і недостатнім рівнем персоналізації доступних сервісів. Існуючі рішення не забезпечують комплексного підходу до вирішення цих проблем, що зумовлює необхідність розробки нових систем, орієнтованих на зручність, гнучкість і інтелектуальну обробку даних.

У зв'язку з цим виникає задача створення програмного засобу, який забезпечує ефективний доступ до інформації про відеоігри та дозволяє формувати персоналізовані рекомендації на основі запитів користувача.

Об'єктом роботи є розроблення Telegram-бота для надання інформації про відеоігри.

Метою роботи є створення Telegram-бота, який забезпечує отримання інформації про ігри, перегляд майбутніх релізів та формування персоналізованих рекомендацій на основі текстових запитів користувача з використанням моделі GPT.

Для досягнення поставленої мети необхідно вирішити такі задачі:

- провести аналіз предметної області та визначити основні проблеми, пов'язані з пошуком і вибором відеоігор;
- дослідити існуючі інформаційні та рекомендаційні системи і визначити їхні переваги та недоліки;
- обґрунтувати доцільність використання чат-ботів як інтерфейсу взаємодії з користувачем;

- проаналізувати можливості застосування технологій штучного інтелекту та мовних моделей для обробки запитів;
- розробити структуру програмної системи, яка поєднує функції інформаційного сервісу та рекомендаційної системи;
- реалізувати Telegram-бота з інтеграцією зовнішніх джерел даних;
- забезпечити обробку запитів користувача у довільній формі з використанням мовної моделі;
- провести тестування системи та оцінити її ефективність.

Таким чином, постановка задачі визначає основні напрямки дослідження та розробки, а також формує основу для подальшого проєктування і реалізації програмної системи.

2 МАТЕМАТИЧНЕ ОБҐРУНТУВАННЯ ВИБРАНИХ МЕТОДІВ ТА ОСОБЛИВОСТІ РОЗРОБЛЕННЯ МЕТОДУ ДЛЯ РЕКОМЕНДАЦІЙ ВІДЕОІГОР

2.1 Аналіз методів рекомендаційних систем

На теренах сучасного інтернет-простору інтелектуальні рекомендаційні сервіси посідають чільне місце, допомагаючи споживачам максимально швидко знаходити необхідні відомості серед безлічі наявних ресурсів. Подібна тенденція є надзвичайно вагомим для відеоігрового ринку, специфіка якого характеризується розгалуженістю жанрів, ігрових консолей та апаратних параметрів. У теперішніх публікаціях щодо оптимізації даних [20] окремо зазначається, що розгортання штучного інтелекту для структурування інформаційних потоків забезпечує вищий рівень ергономічності цифрових застосунків.

Відповідно до наукових праць з автоматизації систем [7], розгортання рекомендаційного інструментарію покликане знизити психологічний тиск інформаційного потоку та оптимізувати аналітичну діяльність споживача. Дана вимога принципова для вебсайтів, що містять колосальну кількість динамічного відеоконтенту.

На практиці сформувалося кілька основних підходів до побудови рекомендаційних систем, серед яких найбільш поширеними є контентно-орієнтовані методи, колаборативна фільтрація та їх комбіновані моделі.

Контентно-орієнтований підхід базується на аналізі характеристик об'єктів. У випадку відеоігор такими характеристиками можуть бути жанр, тип геймплею, платформа, стиль графіки або підтримка мережових режимів. Система співставляє ці параметри з попередніми вподобаннями користувача та формує рекомендації. Перевагою такого підходу є незалежність від даних інших користувачів, що дозволяє ефективно працювати навіть при невеликій кількості поведінкової інформації.

Разом із тим, ефективність контентно-орієнтованого підходу значною мірою залежить від якості та повноти опису об'єктів. Якщо дані є неповними або занадто узагальненими, система може формувати однотипні рекомендації, що обмежує різноманітність запропонованого контенту. Подібні системи не завжди враховують приховані інтереси користувача або контекст взаємодії. Іншим поширеним підходом є колаборативна фільтрація, яка ґрунтується на аналізі поведінки користувачів. У цьому випадку система визначає подібність між користувачами або між об'єктами та формує рекомендації на основі спільних інтересів. Такий підхід дозволяє виявляти приховані залежності між об'єктами та забезпечує більш гнучкий механізм рекомендацій.

Проте колаборативні методи також мають певні обмеження. Найбільш відомою є проблема «холодного старту», яка виникає у випадках, коли система не має достатньо інформації про нового користувача або новий об'єкт. У подібних ситуаціях точність рекомендацій суттєво знижується. У сучасних роботах, присвячених аналізу даних [3], підкреслюється важливість використання адаптивних алгоритмів для підвищення стабільності рекомендаційних систем.

Для підвищення ефективності часто використовуються гібридні підходи, які поєднують різні методи рекомендацій. Вони дозволяють компенсувати недоліки окремих моделей та отримувати більш точні результати. Комбінування алгоритмів аналізу поведінки користувача з контентно-орієнтованими підходами забезпечує більш стабільну якість рекомендацій у складних інформаційних системах [15].

Окремої уваги заслуговують сучасні підходи, пов'язані з обробкою природної мови. Вони дозволяють працювати із запитамі, сформульованими у довільній формі, що є типовим для реальних користувачів. Особливо важливо це у випадках, коли користувач використовує описові або емоційні характеристики замість чітких параметрів пошуку.

Традиційні рекомендаційні алгоритми не завжди здатні ефективно обробляти подібні запити, оскільки переважно орієнтуються на структуровані

дані або ключові слова. У зв'язку з цим сучасні інформаційні системи дедалі частіше використовують технології NLP та великі мовні моделі [21], які забезпечують можливість аналізу змісту запиту та визначення його контексту.

Застосування моделей штучного інтелекту дозволяє враховувати не лише окремі характеристики об'єктів, але й загальний зміст повідомлення користувача. Великі мовні моделі можуть аналізувати текстові описи, уточнювати запити та формувати більш персоналізовані рекомендації. У сучасних дослідженнях ШІ-систем [22] підкреслюється, що використання мультимодальних та мовних моделей значно розширює функціональні можливості рекомендаційних платформ.

Водночас актуальні рекомендаційні платформи все частіше поєднуються з діалоговими інтерфейсами та популярними месенджерами. Такий підхід забезпечує нативну форму комунікації між людиною та сервісом, полегшуючи процес навігації. Впровадження Telegram-ботів на базі інструментів штучного інтелекту формує надійний фундамент для проєктування адаптивних інформаційних комплексів нового покоління.

Додатково слід зазначити, що перспективним напрямком розвитку рекомендаційних систем є використання контекстно-адаптивних моделей, які враховують не лише історію взаємодії користувача, але й поточні умови використання системи. До таких факторів можуть належати час взаємодії, тип пристрою, актуальні тенденції або зміна інтересів користувача. Подібний підхід дозволяє формувати більш гнучкі та персоналізовані рекомендації, що особливо важливо для динамічної сфери відеоігор.

Розробка сучасних рекомендаційних систем передбачає використання різних підходів до обробки інформації та взаємодії з аудиторією. Поєднання гібридних моделей, технологій розпізнавання мовлення та ШІ виступає головним драйвером персоналізації контенту.

Як показано на рисунку 2.1, різні підходи до побудови рекомендаційних систем відрізняються за принципами роботи, що визначає їх переваги та обмеження.



Рисунок 2.1 – Основні типи рекомендаційних систем

Для узагальнення характеристик доцільно представити типи рекомендаційних систем у вигляді таблиці 2.1.

Таблиця 2.1 – Порівняння методів рекомендаційних систем

Метод	Принцип роботи	Переваги	Недоліки
Контентний	Аналіз характеристик	Простота	Обмежена різноманітність
Колаборативний	Аналіз поведінки	Висока точність	Cold start
Гібридний	Комбінація методів	Найкращий результат	Складність

Для формалізації задачі рекомендацій розглянемо множину користувачів

$$U = \{u_1, u_2, \dots, u_n\}, \tag{2.1}$$

та множину об'єктів:

$$I = \{i_1, i_2, \dots, i_n\}. \quad (2.2)$$

Основною задачею є визначення функції корисності:

$$r : U \times I \rightarrow \mathbb{R}, \quad (2.3)$$

яка відображає ступінь зацікавленості користувача у конкретному об'єкті.

У контентно-орієнтованих підходах оцінка рекомендації може базуватися на косинусній мірі подібності:

$$\text{sim}(i, j) = \frac{\vec{i} \times \vec{j}}{\|\vec{i}\| \times \|\vec{j}\|}, \quad (2.4)$$

де $\vec{i}, \vec{j}$ – векторні представлення ігор.

У випадку колаборативної фільтрації прогнозована оцінка визначається як:

$$r(u, i) = \bar{r}_u + \frac{\sum_{v \in N(u)} \text{sim}(u, v) \times (r(v, i) - \bar{r}_u)}{\sum_{v \in N(u)} |\text{sim}(u, v)|}, \quad (2.5)$$

де $N(u)$ – множина схожих користувачів;

$\text{sim}(u, v)$ – коефіцієнт подібності;

$\bar{r}_u$ – середнє значення оцінок користувача.

Резюмуючи, детальний огляд свідчить про те, що базові моделі мають архітектурні недоліки через жорстку прив'язку до структурованої інформації або чужих поведінкових патернів. Водночас передові інструменти генеративного штучного інтелекту забезпечують якісну обробку гнучких людських звернень та створення дійсно корисного рекомендаційного контенту.

2.2 Використання моделей обробки природної мови для формування рекомендацій

Розпізнавання неструктурованого тексту – це база для створення зручного користувацького інтерфейсу. У сфері геймінгу люди звикли формулювати свої бажання мовою емоцій та метафор, шукаючи «гру під настрій» або «щось схоже на улюблений тайтл, але з іншим сетингом». Традиційний пошук за ключовими словами не вміє працювати з таким гнучким контекстом, оскільки очікує на чіткі фільтри, що суттєво погіршує UX ігрових платформ.

Класичні алгоритми рекомендацій здебільшого орієнтовані на структуровані дані, тому вони не здатні ефективно працювати із запитамі природною мовою. У сучасних дослідженнях з аналізу даних [1] підкреслюється необхідність використання адаптивних методів обробки інформації, здатних враховувати контекст та семантичний зміст текстових повідомлень. Це зумовлює активне використання методів обробки природної мови (Natural Language Processing, NLP), які дозволяють аналізувати текст, виділяти ключові ознаки та формувати точніші рекомендації.

Основною метою NLP є перетворення текстової інформації у форму, придатну для машинної обробки. Це досягається шляхом виконання послідовності етапів, які включають токенізацію, нормалізацію, векторизацію та семантичний аналіз. Послідовне виконання цих процедур дозволяє трансформувати складну й гнучку людську мову в структуровані матриці даних. Це дає змогу алгоритмам порівнювати опис користувача з характеристиками об'єктів у базі знань. Саме тому кожен із цих етапів відіграє важливу роль у формуванні точних рекомендацій (рис. 2.2).

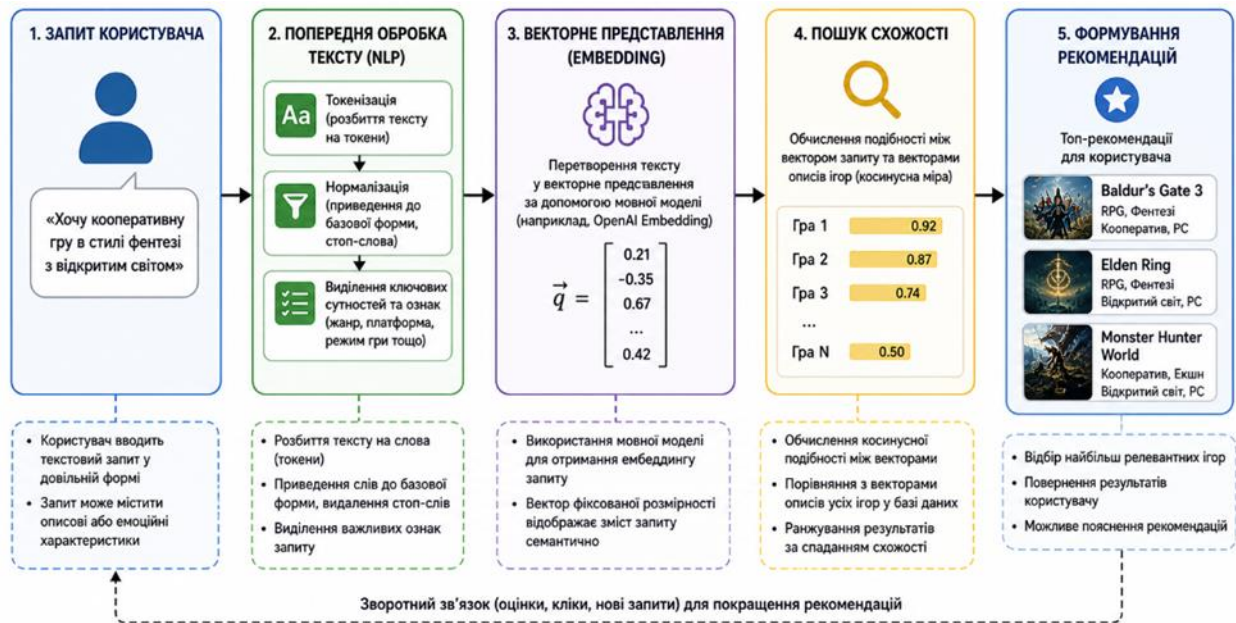


Рисунок 2.2 – Схема обробки текстового запиту користувача

Як показано на рисунку 2.2, обробка запиту починається з введення тексту користувачем. Далі система виконує попередню обробку, яка включає видалення зайвих символів, приведення слів до початкової форми та розбиття тексту на окремі елементи. Після цього формується векторне представлення тексту, яке використовується для визначення подібності між запитом і доступними об'єктами.

Першим етапом є токенизація, під час якої текст розбивається на окремі елементи – токени. Це можуть бути слова, словосполучення або окремі символи. Токенизація дозволяє системі виділити базові компоненти тексту та підготувати його до подальшого аналізу.

Наступним етапом є нормалізація тексту, яка передбачає приведення слів до єдиної форми. На цьому етапі можуть виконуватися видалення службових слів, приведення слів до нижнього регістру, стемінг або лематизація. Подібна обробка дозволяє зменшити вплив різних граматичних форм та підвищити ефективність аналізу текстових запитів. Після цього виконується векторизація тексту. Її основною метою є перетворення текстових даних у числові представлення, придатні для машинного аналізу. У сучасних інформаційних системах активно використовуються методи Word Embedding, TF-IDF та

трансформерні моделі, які дозволяють формувати векторні представлення слів і текстів. Сучасні NLP-підходи [23] забезпечують можливість врахування семантичних зв'язків між словами та контексту їх використання.

Особливе значення у сучасних рекомендаційних системах має семантичний аналіз. Його метою є визначення змісту тексту та контексту взаємодії користувача із системою. Це дозволяє враховувати не лише окремі ключові слова, але й загальний сенс повідомлення. Наприклад, система може визначити, що запити «гра для спокійного проходження» та «щось розслабляюче» мають подібне значення, навіть якщо містять різні слова.

Сучасні системи обробки природної мови дедалі частіше базуються на трансформерних архітектурах та великих мовних моделях. Використання LLM-моделей [21] дозволяє аналізувати складні текстові конструкції, підтримувати діалогову взаємодію та формувати гнучкіші рекомендації відповідно до потреб користувача.

Важливою перевагою NLP є можливість роботи з контекстом запиту. У традиційних системах пошуку результати часто формуються виключно на основі окремих ключових слів, тоді як сучасні мовні моделі здатні враховувати зміст усього повідомлення. Це особливо важливо для рекомендаційних систем у сфері відеоігор, де користувачі часто використовують неформальні або емоційні формулювання.

Методи NLP дозволяють реалізувати діалогову модель взаємодії. Система може уточнювати параметри пошуку, аналізувати попередні повідомлення користувача та поступово покращувати якість рекомендацій. Подібний підхід забезпечує природнішу взаємодію між користувачем і системою та підвищує рівень персоналізації. Разом із перевагами використання NLP існують і певні обмеження. До них належать складність обробки неоднозначних формулювань, залежність якості результатів від навчальних даних та значні обчислювальні ресурси, необхідні для роботи великих мовних моделей. Проте сучасний розвиток ШІ-технологій дозволяє поступово зменшувати ці недоліки та підвищувати ефективність обробки природної мови.

Додатково слід зазначити, що інтеграція NLP-моделей із рекомендаційними системами дозволяє створювати більш адаптивні та інтелектуальні інформаційні сервіси. Поєднання технологій машинного навчання, аналізу тексту та чат-орієнтованих інтерфейсів створює передумови для розробки сучасних систем підтримки користувача, здатних забезпечувати персоналізовану взаємодію у режимі реального часу [13].

З математичної точки зору текстовий запит можна представити як послідовність:

$$X = (x_1, x_2, \dots, x_n), \quad (2.6)$$

де x_i – окремі слова або токени.

Кожен токен перетворюється у вектор за допомогою функції *embedding*:

$$e_i = Emb(x_i). \quad (2.7)$$

Узагальнене представлення запиту визначається як:

$$E = \frac{1}{n} \sum_{i=1}^n e_i. \quad (2.8)$$

Отриманий вектор E використовується для обчислення подібності між запитом та описами ігор. Для цього застосовується косинусна міра подібності:

$$sim(q, i) = \frac{E_q \times E_i}{\|E_q\| \times \|E_i\|}. \quad (2.9)$$

Цей підхід дозволяє знаходити найбільш релевантні об'єкти навіть у випадках, коли запит користувача сформульований неточно або містить неявні характеристики. Завдяки семантичному аналізу система здатна визначати

приховані зв'язки між словами та інтерпретувати загальний зміст повідомлення, а не лише окремі ключові фрази.

Сучасним етапом розвитку NLP є використання великих мовних моделей, таких як GPT. На відміну від класичних підходів, ці моделі здатні враховувати контекст запиту, аналізувати зв'язки між словами та інтерпретувати складні мовні конструкції. У сучасних дослідженнях великих мовних моделей [18] підкреслюється, що використання LLM-технологій дозволяє значно підвищити точність обробки текстових даних та покращити якість взаємодії із користувачем.

Важливою особливістю мовних моделей є їх здатність працювати із багатокomпонентними запитами. Наприклад, запит «відкрита RPG з хорошим сюжетом і невисокими вимогами» містить декілька критеріїв, які необхідно врахувати одночасно. Модель аналізує кожен компонент запиту та формує рекомендації, що відповідають усім заданим умовам. Подібний підхід забезпечує більш гнучку систему пошуку та дозволяє враховувати індивідуальні вподобання користувача.

Понад те, великі мовні моделі мають здатність розпізнавати імпліцитні (приховані) параметри, зокрема емоційний стан користувача або специфіку поточного контексту. Наприклад, формулювання «гра для відпочинку після роботи» система інтерпретує як запит на казуальний або заспокійливий ігровий процес. Реалізація такої семантичної дедукції є занадто складною для традиційних евристичних алгоритмів, тоді як сучасні NLP-моделі успішно ідентифікують подібні контекстуальні маркери завдяки глибинному аналізу текстової семантики [23].

Ще однією важливою перевагою є можливість генерації пояснень. Система може не лише запропонувати гру, але й пояснити, чому саме вона була обрана. Це підвищує довіру користувача до системи та покращує загальний досвід взаємодії. Можливість формування структурованих та зрозумілих відповідей є однією з ключових переваг сучасних чат-орієнтованих ІІІ-систем. Додатково важливою особливістю великих мовних моделей є їх здатність підтримувати

діалогову взаємодію з користувачем. Система може уточнювати окремі параметри пошуку, ставити додаткові запитання та поступово формувати більш точні рекомендації. Подібний підхід дозволяє реалізувати адаптивну модель взаємодії, у якій результати пошуку змінюються відповідно до поточного контексту діалогу та поведінки користувача [22].

Для порівняння різних підходів до обробки запитів доцільно використати таблицю 2.2.

Таблиця 2.2 – Порівняння підходів до обробки запитів

Підхід	Тип запитів	Гнучкість	Точність
Класичний пошук	Структуровані	Низька	Середня
Фільтрація	Поведінкові	Середня	Висока
NLP/GPT	Текстові	Висока	Висока

Як видно з таблиці, використання NLP дозволяє працювати з більш складними типами запитів та забезпечує вищу точність рекомендацій.

Крім базових підходів, сучасні системи активно використовують трансформерні моделі, які базуються на механізмі самоуваги (attention). Основною ідеєю таких моделей є визначення важливості кожного слова у контексті всього запиту.

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \quad (2.10)$$

де Q , K , V – матриці запиту, ключів та значень відповідно.

Механізм уваги (attention mechanism) дозволяє моделі визначати, які слова є найбільш значущими у межах запиту. Наприклад, у повідомленні «гра з відкритим світом і кооперативом» система приділяє більшу увагу словам «відкритий світ» та «кооператив», оскільки саме вони визначають основні

характеристики бажаного контенту. Використання механізмів уваги є однією з ключових особливостей сучасних трансформерних моделей та забезпечує більш точний аналіз текстових даних [24].

До того ж сучасні NLP-архітектури використовують динамічні контекстуальні ембединги, які адаптують значення кожного токена до навколишнього тексту. Це нівелює труднощі, пов'язані з омонімією та мовною неоднозначністю. Зважаючи на це, алгоритм набагато коректніше розшифровує інтенції користувача та формує точну й адаптивну пошукову видачу.

Ще одним важливим аспектом є можливість адаптації моделей до конкретної предметної області. У випадку відеоігор це означає врахування спеціалізованої термінології, назв жанрів, типів ігрового процесу, характеристик геймплею та інших особливостей ігрового середовища. Подібна адаптація дозволяє суттєво підвищити точність рекомендацій та забезпечити більш якісний аналіз користувацьких запитів [11].

Важливо також зазначити, що використання NLP дозволяє реалізувати діалогову взаємодію з користувачем. Система може уточнювати параметри пошуку, пропонувати альтернативні варіанти та адаптувати результати відповідно до змін у запитах користувача. Подібний механізм взаємодії є особливо ефективним у чат-орієнтованих системах та Telegram-ботах, де процес пошуку реалізується у формі природного діалогу.

Сучасні мовні моделі також дозволяють поєднувати текстовий аналіз із зовнішніми джерелами даних та API-сервісами. Це забезпечує можливість отримання актуальної інформації про відеоігри у режимі реального часу та підвищує практичну ефективність рекомендаційної системи. Інтеграція ШІ-моделей із зовнішніми інформаційними платформами [25] є одним із перспективних напрямків розвитку сучасних інтелектуальних сервісів.

На основі проведеного аналізу можна констатувати, що використання методів обробки природної мови є ключовим для побудови інтелектуальних рекомендаційних систем. Інтеграція NLP дозволяє ефективно вирішити три стратегічні завдання:

- глибинна семантика: забезпечення концептуального розуміння неструктурованих запитів користувача;
- контекстуалізація: динамічна фіксація та аналіз умов поточного сеансу взаємодії;
- персоналізація: формування високоточних індивідуальних рекомендацій під конкретного споживача.

2.3 Архітектура системи рекомендацій відеоігор

Під час проєктування сучасних інформаційних комплексів фундаментальне значення має формування їхньої архітектурної моделі, яка регламентує взаємодію системних компонентів та логіку обробки інформаційних потоків. Архітектурні рішення безпосередньо визначають такі системні метрики, як швидкодія, масштабованість, відмовостійкість та якість користувацького інтерфейсу. Стосовно рекомендаційного сервісу у сфері відеоігор, ключовими архітектурними вимогами постають підтримка аналізу текстових промптів, інтеграція із зовнішніми інформаційними базами (API) та розгортання моделей штучного інтелекту.

У сучасних дослідженнях архітектури інформаційних систем [5] підкреслюється, що модульний підхід дозволяє значно спростити процес розроблення та забезпечити гнучкість програмного забезпечення. Модульна архітектура дозволяє розділити систему на незалежні компоненти, кожен з яких відповідає за окрему функціональність. Це спрощує тестування, підтримку та подальше масштабування системи.

У межах даної роботи запропоновано архітектуру, яка поєднує чат-інтерфейс, серверну логіку, модулі обробки природної мови та зовнішні API. Загальна структура системи представлена на рисунку 2.3.

Архітектура системи рекомендацій відеоігор

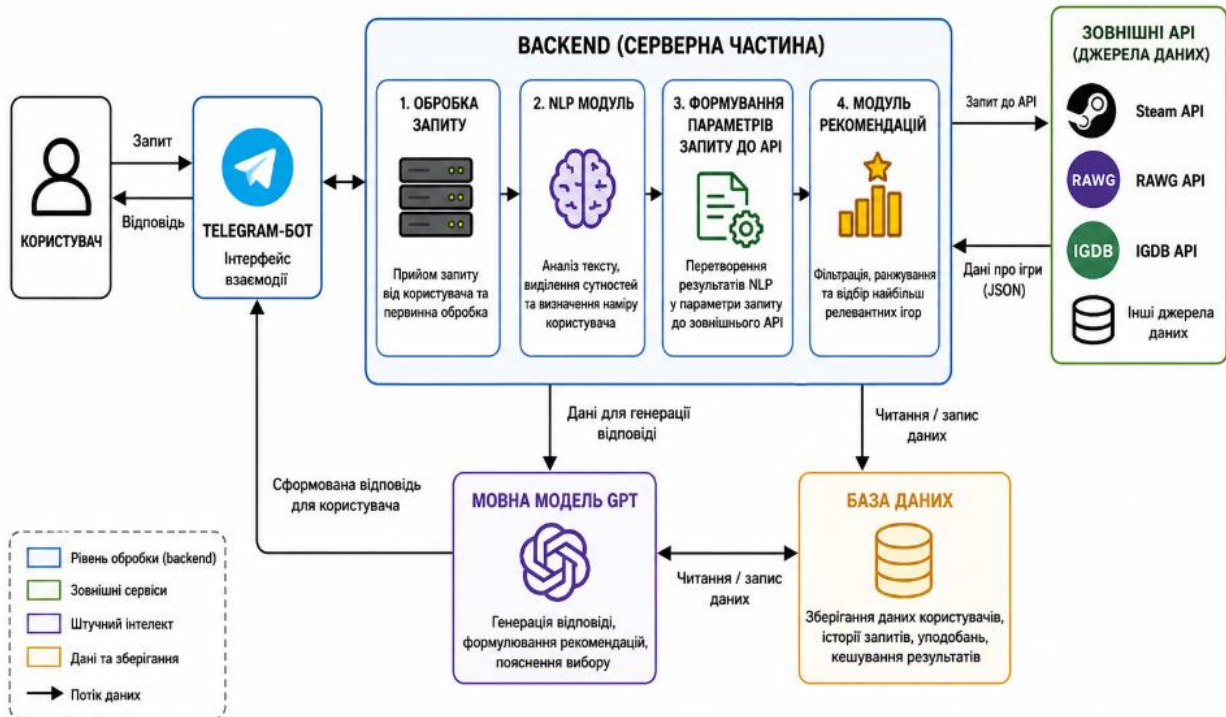


Рисунок 2.3 – Архітектура системи рекомендацій відеоігор

Рисунок 2.3 розроблено автором на основі [5, 20–24]. Основним елементом взаємодії з користувачем є Telegram-бот, який виступає у ролі інтерфейсу системи. Використання месенджера як клієнтської платформи обумовлене його доступністю, зручністю та підтримкою різних типів пристроїв. Користувач формулює запит у текстовій формі, що забезпечує просту та інтуїтивну взаємодію без необхідності використання складного графічного інтерфейсу. У сучасних дослідженнях чат-орієнтованих систем підкреслюється, що використання месенджерів дозволяє значно спростити доступ користувача до інформаційних сервісів [26].

Після отримання запиту він передається до серверної частини системи (backend), яка виконує роль центрального елемента архітектури. Backend відповідає за обробку запитів, управління логікою системи та координацію взаємодії між окремими модулями. Саме серверна частина забезпечує інтеграцію NLP-модуля, рекомендаційної системи та зовнішніх API-сервісів.

Наступним етапом є обробка запиту за допомогою NLP-модуля, який виконує аналіз тексту, виділення ключових характеристик і визначення намірів користувача. У результаті формується структуроване представлення запиту, що використовується для подальшої обробки. Сучасні методи обробки природної мови [27] дозволяють враховувати контекст повідомлення та аналізувати складні текстові конструкції. Після цього система звертається до зовнішніх API, які надають інформацію про відеоігри, їх характеристики, рейтинги та інші параметри. Використання API дозволяє отримувати актуальні дані без необхідності створення власної великої бази даних, що значно спрощує архітектуру системи та підвищує гнучкість її розширення.

Отримані дані передаються до модуля формування рекомендацій, де виконується їх фільтрація, аналіз і ранжування відповідно до параметрів запиту. На цьому етапі враховуються характеристики ігор, контекст запиту користувача та результати роботи мовної моделі. Подібний підхід дозволяє формувати більш релевантні рекомендації та покращувати якість взаємодії із системою.

Окрему роль у системі відіграє мовна модель GPT, яка використовується для інтерпретації складних запитів і генерації відповіді. Вона дозволяє враховувати контекст повідомлення, аналізувати приховані характеристики запиту та формувати зрозумілі пояснення рекомендацій. Використання великих мовних моделей [18] забезпечує більш природну форму взаємодії та підвищує рівень персоналізації системи.

GPT-модель дозволяє реалізувати механізм діалогової взаємодії. У випадках, коли запит користувача є недостатньо точним або містить неоднозначні характеристики, система може уточнювати окремі параметри та поступово формувати більш точні результати. Це забезпечує адаптивність системи та покращує користувацький досвід.

Важливою особливістю запропонованої архітектури є розподіл функціональності між окремими модулями. Подібний підхід дозволяє незалежно оновлювати або модифікувати компоненти системи без необхідності зміни всієї

архітектури. Наприклад, у майбутньому можлива інтеграція нових API-сервісів, заміна мовної моделі або розширення механізмів аналізу даних [5].

Додатково слід зазначити, що використання модульної та API-орієнтованої архітектури створює передумови для подальшого масштабування системи. Це дозволяє інтегрувати додаткові функції, реалізовувати підтримку інших платформ або розширювати можливості рекомендаційного механізму без суттєвих змін базової структури програмного забезпечення.

Для узагальнення функцій компонентів наведено таблицю 2.3.

Таблиця 2.3 – Основні компоненти системи

Компонент	Призначення
Telegram-бот	Інтерфейс користувача
Backend	Логіка системи
NLP	Аналіз тексту
API	Джерело даних
GPT	Генерація відповідей

З формальної точки зору архітектуру системи можна описати наступним чином:

$$Y = G(F(API(NLP(X)))), \quad (2.11)$$

де X – запит користувача;

а Y – результат обробки.

У цьому контексті процес функціонування системи можна розглядати як послідовність перетворень даних:

$$D_{out} = T\left(G\left(F\left(API(NLP(D_{in}))\right)\right)\right). \quad (2.12)$$

Такий підхід дозволяє формалізувати логіку роботи системи та спрощує її аналіз. Розподіл функціональності між окремими компонентами забезпечує більш гнучке керування процесами обробки даних та полегшує подальшу модифікацію системи.

Також варто вказати, що система має багаторівневу структуру, яка включає рівень представлення, рівень обробки, аналітичний рівень та рівень даних. Це відповідає сучасним принципам побудови інформаційних систем і забезпечує чітке розмежування функцій між компонентами. Подібна архітектура дозволяє незалежно оновлювати окремі модулі та забезпечує більш ефективну підтримку програмного забезпечення [28].

Рівень представлення відповідає за взаємодію користувача із системою через Telegram-бота. На цьому рівні забезпечується прийом повідомлень, відображення результатів пошуку та організація діалогової взаємодії. Рівень обробки виконує маршрутизацію запитів, управління бізнес-логікою та координацію роботи окремих модулів системи.

Аналітичний рівень включає модулі NLP, рекомендаційний механізм та інтеграцію з великою мовною моделлю GPT. Саме на цьому рівні виконується аналіз текстових запитів, формування рекомендацій та обробка результатів. Рівень даних відповідає за взаємодію із зовнішніми API, кешування інформації та тимчасове збереження результатів запитів.

Персоналізація рекомендацій досягається за рахунок урахування попередніх дій користувача, історії взаємодії та параметрів пошуку. Це дозволяє адаптувати систему до індивідуальних потреб користувача та підвищувати релевантність результатів. У сучасних роботах, присвячених ШІ-системам [13], підкреслюється важливість використання адаптивних механізмів персоналізації для покращення якості цифрових сервісів.

Для забезпечення ефективної роботи системи застосовуються сучасні технології, зокрема хмарні сервіси, контейнеризація та балансування навантаження. Це дозволяє забезпечити масштабованість і стабільність роботи навіть при зростанні кількості користувачів. Використання контейнеризації

також спрощує розгортання системи та забезпечує ізольоване виконання окремих компонентів.

Окрім цього, важливу роль відіграє оптимізація взаємодії із зовнішніми API. Для цього застосовуються кешування результатів, обмеження кількості запитів та асинхронна обробка, що дозволяє зменшити затримки та підвищити продуктивність системи. Подібний підхід є особливо важливим для чат-орієнтованих систем, які повинні забезпечувати швидку відповідь користувачу.

Надійність системи забезпечується за рахунок реалізації механізмів обробки помилок, повторних запитів та резервних сценаріїв. Це дозволяє підтримувати стабільну роботу навіть у випадку часткових відмов окремих компонентів або зовнішніх сервісів.

Важливим аспектом є також забезпечення безпеки системи. Це включає захист API-ключів, перевірку введених даних, контроль доступу та обмеження несанкціонованих запитів. Реалізація подібних механізмів дозволяє підвищити рівень захисту інформації та забезпечити безпечну взаємодію користувача із системою.

Додатково слід зазначити, що модульна архітектура створює можливість подальшого розвитку програмного забезпечення. У майбутньому система може бути розширена шляхом інтеграції нових ШІ-моделей, підтримки інших месенджерів або впровадження додаткових механізмів рекомендацій. Подібний підхід забезпечує гнучкість системи та її адаптацію до змін сучасного цифрового середовища [29].

У підсумку, спроектована архітектура платформи підбору відеоігор ефективно відповідає викликам сучасного ринку. Поєднання алгоритмів ШІ, швидкої обробки інформації та інтеграцій із зовнішніми базами знань дозволяє розгорнути кастомізовану екосистему, де безшовний досвід користувача та стабільність роботи системи є головними пріоритетами.

2.4 Реалізація системи та аналіз результатів

Практична реалізація системи рекомендацій відеоігор виконана у вигляді Telegram-бота, який забезпечує взаємодію користувача із системою у діалоговому режимі. Такий підхід дозволяє реалізувати природний спосіб формування запитів і спрощує використання системи без необхідності встановлення додаткового програмного забезпечення. У сучасних дослідженнях чат-орієнтованих систем [30] підкреслюється, що використання месенджерів є ефективним рішенням для сучасних інформаційних сервісів.

У якості основної мови програмування було обрано Python, що обумовлено його широким застосуванням у сфері розробки систем штучного інтелекту та наявністю великої кількості бібліотек для роботи з API, обробки природної мови та побудови серверної логіки. Python забезпечує швидке прототипування, зручність інтеграції різних компонентів системи та підтримку сучасних ІІТ-технологій.

Система реалізована відповідно до клієнт-серверної архітектури, де Telegram-бот виступає клієнтською частиною, а сервер (backend) відповідає за обробку запитів, взаємодію з API та формування рекомендацій. Подібний підхід дозволяє розділити функціональність системи, підвищити її гнучкість та забезпечити можливість подальшого масштабування [28].

Основний алгоритм роботи системи включає декілька послідовних етапів. На першому етапі користувач формує текстовий запит у довільній формі, який надсилається до Telegram-бота. Далі цей запит передається до серверної частини, де виконується його попередня обробка та підготовка до подальшого аналізу.

На етапі обробки тексту застосовуються методи обробки природної мови, що дозволяють виділити ключові характеристики запиту. Сучасні NLP-моделі здатні ефективно працювати з неструктурованими даними, аналізувати контекст повідомлення та визначати основні параметри пошуку. Використання мовних моделей [21] дозволяє системі більш точно інтерпретувати текстові запити користувачів.

Після аналізу запиту формується набір параметрів, які використовуються для звернення до зовнішніх API. У рамках реалізації системи можуть використовуватися такі джерела даних, як RAWG API, IGDB або інші сервіси, що надають інформацію про відеоігри. Використання REST API забезпечує стандартизований спосіб отримання даних та їх подальшої обробки [23].

Отримані дані представляються у форматі JSON і містять інформацію про ігри, включаючи назву, жанр, рейтинг, платформу, дату релізу та інші характеристики. Далі ці дані передаються до модуля формування рекомендацій, де виконується фільтрація та аналіз результатів відповідно до параметрів запиту.

Важливою частиною практичної реалізації є інтеграція із мовною моделлю GPT, яка використовується для аналізу складних запитів та генерації текстових відповідей. Модель дозволяє враховувати контекст взаємодії, аналізувати багатокomпонентні запити та формувати більш природні пояснення рекомендацій [16].

Для взаємодії з Telegram API можуть використовуватися спеціалізовані Python-бібліотеки, зокрема aiogram або python-telegram-bot, які спрощують реалізацію обробки повідомлень, команд та кнопок інтерфейсу. Використання подібних бібліотек дозволяє значно скоротити час розроблення та забезпечує зручну організацію логіки Telegram-бота.

Додаткова інженерна перевага рішення полягає у використанні асинхронних патернів проєктування для керування внутрішніми процесами системи. Завдяки цьому паралельні API-виклики та багатокористувацькі сесії можуть оброблятися незалежно одна від одної, що дозволяє мінімізувати затримки та підвищити швидкодію сервісу. Це є особливо важливим для чат-орієнтованих систем, які повинні забезпечувати оперативне реагування на дії користувача.

Для підвищення продуктивності системи можуть використовуватися механізми кешування результатів API-запитів. Це дозволяє зменшити навантаження на зовнішні сервіси та прискорити повторне отримання

популярних даних. Додатково реалізуються механізми обробки помилок та повторних запитів, що забезпечує більш стабільну роботу системи.

Важливим аспектом практичної реалізації є забезпечення безпеки системи. З цією метою використовується приховане зберігання API-ключів, перевірка вхідних даних та контроль доступу до окремих функцій сервісу. Подібні механізми дозволяють мінімізувати ризики несанкціонованого доступу та підвищити надійність роботи програмного забезпечення.

Додатково слід зазначити, що модульна структура програмного забезпечення забезпечує можливість подальшого розвитку системи. У майбутньому можуть бути реалізовані додаткові функції, зокрема розширені механізми персоналізації, підтримка нових джерел даних або інтеграція з іншими ШІ-моделями [13].

На етапі формування рекомендацій виконується фільтрація та ранжування отриманих результатів. Для цього використовується оцінка відповідності гри параметрам запиту користувача. У загальному вигляді функцію оцінки можна представити як:

$$S_{core}(i) = \sum_{k=1}^n w_k \times f_k(i), \quad (2.13)$$

де $f_k(i)$ – значення k -го параметра гри;

w_k – ваговий коефіцієнт.

Такий підхід дозволяє враховувати різні характеристики гри та визначати її релевантність відносно запиту користувача. Під час формування рекомендацій система аналізує жанр, стиль гри, підтримку мережевих режимів, платформу, рейтинг та інші параметри, що дозволяє забезпечити більш точний підбір контенту.

Після визначення найбільш релевантних варіантів результати передаються до мовної моделі GPT, яка формує текстову відповідь. Модель генерує рекомендації у зрозумілій формі та може додатково пояснювати причини вибору певної гри. Використання великих мовних моделей [16] дозволяє значно

підвищити якість взаємодії з користувачем та забезпечити більш природну форму діалогу.

Важливою особливістю реалізованої системи є підтримка діалогового режиму. Користувач може уточнювати запити, змінювати критерії пошуку або отримувати додаткову інформацію про рекомендовані ігри. Це дозволяє адаптувати систему до індивідуальних потреб користувача та підвищити точність рекомендацій. Подібний підхід забезпечує більш гнучку взаємодію між користувачем і системою та покращує загальний користувацький досвід [15].

Для оцінки ефективності системи було проведено тестування з використанням різних типів запитів:

- прості запити (наприклад, «RPG ігри»);
- комбіновані запити («RPG з відкритим світом і кооперативом»);
- неструктуровані запити («щось атмосферне для вечора»).

Під час тестування оцінювалися точність рекомендацій, швидкість обробки запитів та коректність інтерпретації текстових повідомлень. Окрему увагу було приділено роботі системи з неструктурованими та багатокomпонентними запитами, оскільки саме вони є найбільш складними для традиційних рекомендаційних алгоритмів.

Результати тестування показали, що система успішно обробляє всі типи запитів, проте найбільш ефективно працює саме з неструктурованими запитами. Це пояснюється використанням NLP та мовної моделі, які дозволяють враховувати семантику тексту та аналізувати контекст повідомлення [24].

Водночас аналіз підтвердив, що інтеграція GPT уможливорює створення набагато живіших та простіших інтеракцій на відміну від базових рекомендаційних комплексів. Користувач отримує не звичайний набір програм, а ємні пояснення поточного вибору, що полегшує оцінку контенту і збільшує рівень довіри до платформи.

У процесі тестування також було перевірено стабільність роботи системи при одночасній обробці декількох запитів. Використання асинхронної

архітектури та механізмів оптимізації API-запитів дозволило забезпечити стабільний час відповіді навіть при збільшенні навантаження на систему.

Важливим результатом практичної реалізації стало підтвердження ефективності використання Telegram-бота як інтерфейсу взаємодії. Чат-орієнтований підхід забезпечує простоту використання, доступність сервісу на різних пристроях та більш природний формат роботи з інформаційною системою [27].

Разом з тим варто підкреслити, що створений програмний комплекс володіє значними можливостями для майбутньої модернізації. Надалі планується масштабування інструментарію через підключення сторонніх сервісів, залучення прогресивних ШІ-алгоритмів чи інтеграцію інструментів аналітики користувацького досвіду. Такий крок забезпечить глибшу індивідуалізацію та оптимізує точність рекомендаційних рішень стратегічно.

Для наочності результати тестування можна представити у вигляді таблиці 2.4.

Таблиця 2.4 – Результати тестування системи

Тип запиту	Точність	Час відповіді
Простий	Висока	Низький
Комбінований	Середня	Середній
Неструктурований	Висока	Середній

Як видно з таблиці 2.4, система демонструє стабільні результати для різних типів запитів.

Крім точності, важливим показником ефективності системи є швидкодія. У ході тестування було встановлено, що середній час формування відповіді становить приблизно 2 – 4 секунди залежно від складності запиту та швидкості роботи зовнішніх API-сервісів. Подібний результат відповідає сучасним вимогам до інформаційних сервісів, оскільки значні затримки можуть негативно впливати на користувацький досвід та знижувати комфорт взаємодії із системою [14].

Для забезпечення стабільної швидкодії у системі застосовуються механізми асинхронної обробки запитів, кешування результатів та оптимізації взаємодії із зовнішніми API. Це дозволяє зменшити навантаження на серверну частину та скоротити час очікування відповіді користувачем.

У процесі тестування також було виявлено певні обмеження системи. Зокрема, у випадках дуже загальних або неоднозначних запитів система може формувати менш точні рекомендації. Наприклад, короткі повідомлення без конкретних характеристик або надто абстрактні формулювання ускладнюють процес інтерпретації запиту навіть при використанні сучасних NLP-моделей.

Ключовим фактором ризику тут є якість вхідного потоку даних із зовнішніх джерел. Якщо API передає урізану, неповну або застарілу інформацію про гру, нейромережа опиняється в умовах браку контексту. Це безпосередньо б'є по релевантності підбору, змушуючи алгоритм видавати розмиті або частково помилкові рекомендації.

Ще одним важливим обмеженням є залежність системи від зовнішніх сервісів. У разі недоступності API, перевищення лімітів запитів або виникнення мережових помилок система може працювати некоректно або тимчасово втрачати доступ до окремих функцій. Для зменшення впливу подібних ситуацій доцільно використовувати механізми кешування, резервні джерела даних та повторні запити [23].

Окремої уваги потребує і питання масштабованості системи. При значному збільшенні кількості користувачів може виникати додаткове навантаження на серверну частину та API-сервіси. Для вирішення цієї проблеми можуть використовуватися хмарні технології, балансування навантаження та контейнеризація окремих компонентів системи.

Разом із тим результати проведеного дослідження підтверджують ефективність запропонованого підходу. Поєднання чат-інтерфейсу, NLP, зовнішніх API та мовної моделі дозволяє створити систему, яка забезпечує зручний доступ до інформації та формує релевантні рекомендації відповідно до потреб користувача.

Використання великих мовних моделей забезпечує більш гнучку взаємодію із системою, дозволяє працювати з неструктурованими запитами та формувати зрозумілі текстові відповіді. Це суттєво покращує користувацький досвід порівняно з традиційними рекомендаційними системами, які базуються виключно на структурованих параметрах пошуку [12].

Додатково слід зазначити, що реалізована система має перспективи подальшого розвитку. У майбутньому можливе впровадження механізмів навчання на основі поведінки користувача, розширення підтримки мультимодальних моделей та інтеграція нових джерел даних. Це дозволить підвищити рівень персоналізації, покращити точність рекомендацій та розширити функціональні можливості інформаційної системи.

У підсумку, представлена система засвідчує успішне подолання викликів інформаційного пресингу під час вибору розважального медіаконтенту. Завдяки використанню інструментів штучного інтелекту, технологій обробки тексту та чат-моделей забезпечується точна персоналізація пропозицій, оптимізується алгоритм пошуку й вибудовується якісно новий рівень взаємодії із юзером.

3 РЕАЛІЗАЦІЯ ЗАДАЧІ, ТЕСТУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

3.1 Вибір засобів програмної реалізації

Для реалізації Telegram-бота рекомендації відеоігор було обрано мову програмування Python. Вибір даної мови обумовлений її широким використанням у сфері веброзробки, автоматизації та створення систем штучного інтелекту. Python має значну кількість готових бібліотек для роботи з мережевими сервісами, базами даних та зовнішніми API, що дозволяє скоротити час розроблення програмного забезпечення та підвищити його надійність.

Як середовище розробки використовувалась Visual Studio 2022. Дане середовище забезпечує зручні засоби написання, налагодження та тестування програмного коду. Наявність вбудованих інструментів аналізу коду дозволяє оперативно виявляти помилки та контролювати якість програмного продукту.

Для реалізації функціоналу Telegram-бота було використано бібліотеку aiogram версії 3.27.0. Дана бібліотека є асинхронним фреймворком для роботи з Telegram Bot API та дозволяє ефективно обробляти велику кількість одночасних запитів користувачів. Використання асинхронного підходу забезпечує швидке виконання операцій взаємодії із зовнішніми сервісами без блокування роботи бота.

Для отримання інформації про відеоігри використано зовнішній сервіс RAWG API. Даний сервіс надає доступ до великої бази даних відеоігор, яка містить інформацію про назви ігор, жанри, рейтинги, платформи, дати релізів, описи та графічні матеріали. Використання RAWG API дозволило реалізувати пошук ігор, формування тематичних добірок та відображення актуальної інформації про ігрові продукти.

Для реалізації функцій пошуку акційних пропозицій було використано сервіс CheapShark API. За допомогою даного API бот отримує інформацію про поточні знижки на відеоігри у різних цифрових магазинах. Отримані дані

включають назву гри, початкову та поточну ціну, відсоток знижки та магазин, у якому доступна пропозиція.

Для реалізації функціоналу інтелектуальних рекомендацій було використано OpenAI API. Даний сервіс забезпечує доступ до сучасних великих мовних моделей, здатних аналізувати текстові запити користувачів природною мовою. Завдяки використанню OpenAI API користувач може вказати свої вподобання щодо жанру, тематики або особливостей ігрового процесу, після чого система формує персоналізовані рекомендації відеоігор. Застосування технологій штучного інтелекту дозволило підвищити якість рекомендацій та зробити взаємодію з ботом більш природною і зручною для користувача.

Для зберігання даних користувачів було використано вбудовану систему керування базами даних SQLite. Дана СКБД не потребує встановлення окремого серверного програмного забезпечення та добре підходить для невеликих інформаційних систем. У базі даних зберігається історія переглянутих рекомендацій, оцінки користувачів та список бажаних ігор.

Використання зазначених програмних засобів дозволило створити функціональний Telegram-бот для рекомендації відеоігор, який забезпечує отримання інформації про ігри, пошук знижок, ведення персональної статистики користувача та формування рекомендацій за допомогою технологій штучного інтелекту.

3.2 Створення та налаштування Telegram-бота

Після вибору технологій та інструментальних засобів було розпочато реалізацію програмного продукту у вигляді Telegram-бота для рекомендації відеоігор. Основною платформою взаємодії з користувачем було обрано месенджер Telegram, оскільки він забезпечує швидкий доступ до функціоналу бота без необхідності встановлення додаткового програмного забезпечення.

Першим етапом розроблення стало створення нового Telegram-бота за допомогою офіційного сервісу BotFather. Для цього було використано команду `/newbot`, після чого було задано назву бота та його унікальне ім'я. Після завершення процедури реєстрації сервіс згенерував спеціальний токен доступу, який використовується для взаємодії між програмним кодом та Telegram Bot API (рис. 3.1).

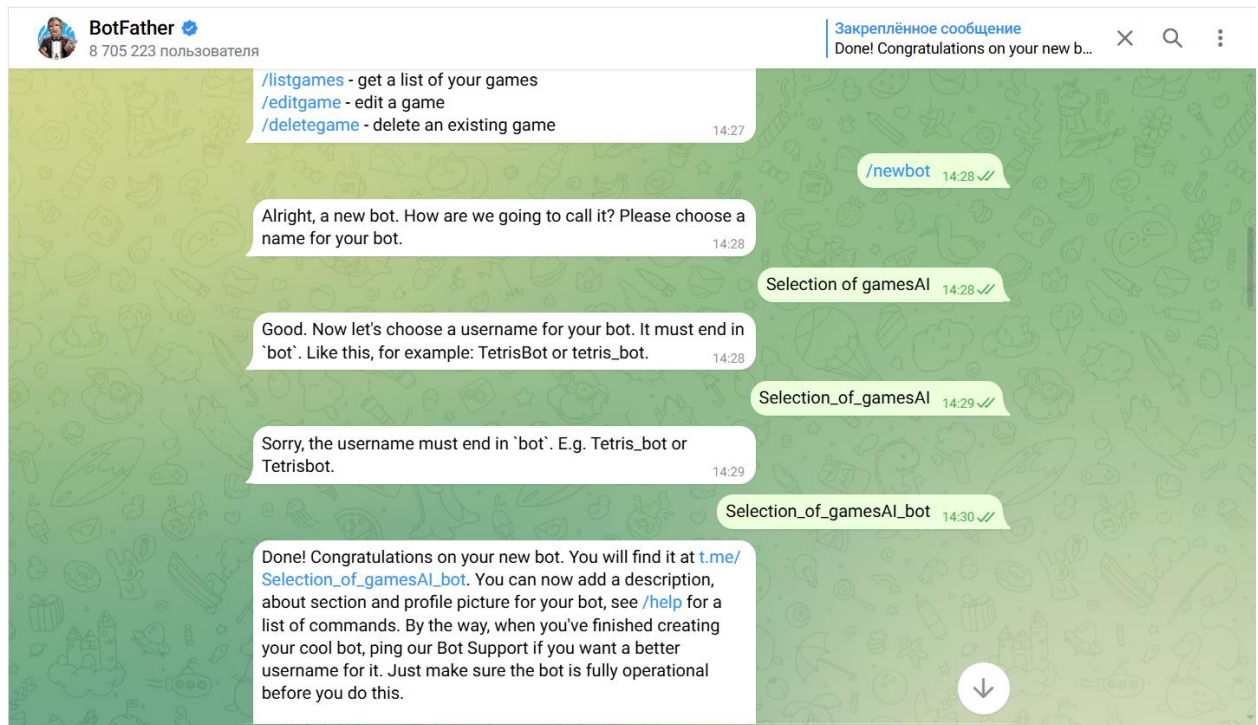


Рисунок 3.1 – Створення бота через BotFather

З метою підвищення безпеки токен не зберігався безпосередньо у вихідному коді програми. Для цього було використано окремий конфігураційний файл `.env`, у якому розміщуються всі секретні параметри доступу до зовнішніх сервісів. Такий підхід дозволяє запобігти випадковому розголошенню конфіденційної інформації під час публікації або передачі програмного коду.

Окрім токена Telegram-бота, у конфігураційному файлі було збережено ключі доступу до зовнішніх сервісів RAWG API, CheapShark API та OpenAI API, які використовуються для отримання інформації про відеоігри, пошуку

актуальних знижок та формування рекомендацій на основі технологій штучного інтелекту.

Після налаштування середовища було реалізовано початкову ініціалізацію Telegram-бота за допомогою бібліотеки `aiogram`. На цьому етапі створюються об'єкти бота та диспетчера повідомлень, які забезпечують приймання та обробку запитів користувачів (рис. 3.2).

```

56      # =====
57      # Bot / Dispatcher
58      # =====
59      bot = Bot(token=BOT_TOKEN)

```

Рисунок 3.2 – Ініціалізація Telegram-бота в Visual Studio 2022

Після запуску програми бот починає отримувати повідомлення від користувачів та передавати їх відповідним обробникам подій. Для реалізації взаємодії з користувачем було створено систему команд та інтерактивне меню з використанням кнопок Telegram (рис. 3.3).

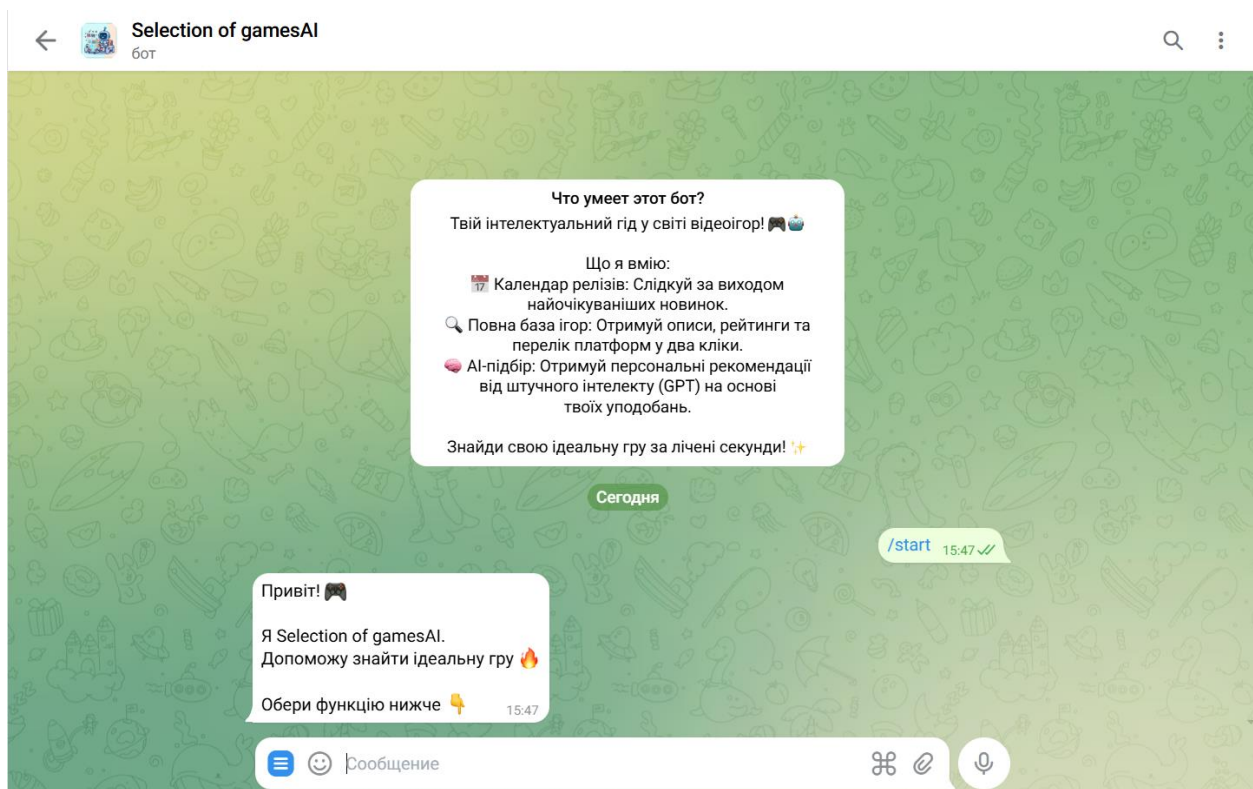


Рисунок 3.3 – Привітальне повідомлення Telegram-бота після запуску

Головне меню бота містить набір функцій, які забезпечують доступ до основних можливостей системи. Серед них пошук інформації про гру, отримання випадкової гри, перегляд актуальних знижок, пошук гри зі знижкою, рекомендації на основі штучного інтелекту, календар майбутніх релізів, перегляд дати виходу гри, топи ігор за жанрами та інші функції.

Для створення меню було використано клавіатуру типу ReplyKeyboardMarkup, яка дозволяє відображати набір кнопок безпосередньо в інтерфейсі Telegram. Такий підхід підвищує зручність використання системи та спрощує доступ до основного функціоналу.

Для реалізації багатокрокових сценаріїв взаємодії з користувачем було використано механізм скінченних автоматів станів (FSM), який підтримується бібліотекою aiogram. Застосування FSM дозволяє зберігати поточний стан користувача та коректно обробляти послідовність його дій під час пошуку ігор, отримання рекомендацій та роботи зі списком бажаного (рис. 3.4).

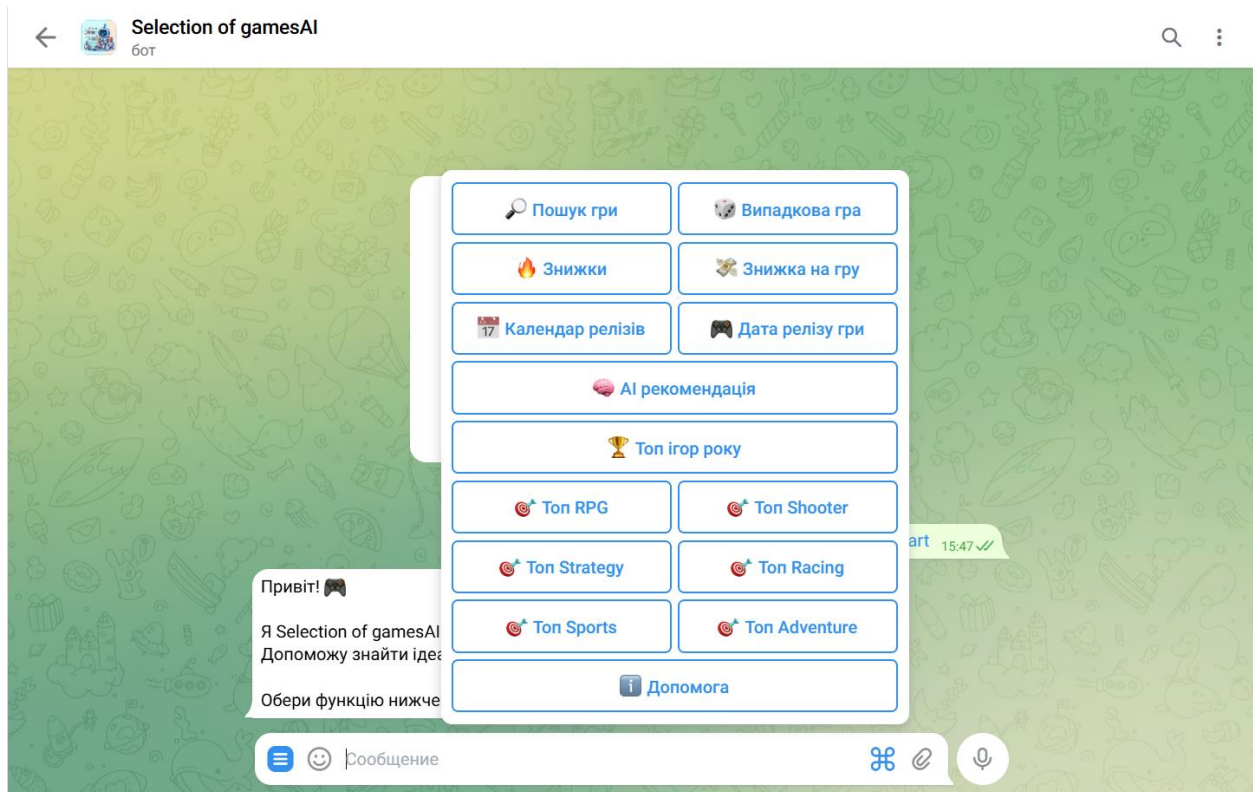


Рисунок 3.4 – Інтерфейс головного меню Telegram-бота для рекомендації відеоігор

Після завершення налаштування було виконано тестовий запуск системи та перевірено коректність підключення до Telegram Bot API. Результати перевірки підтвердили успішне отримання повідомлень від користувачів та коректне функціонування головного меню бота.

3.3 Реалізація пошуку та отримання інформації про відеоігри за допомогою RAWG API

Однією з основних функцій розробленого Telegram-бота є пошук інформації про відеоігри та відображення детальних відомостей про них. Для реалізації даного функціоналу було використано зовнішній сервіс RAWG API, який надає доступ до великої бази даних відеоігор різних жанрів та платформ (рис. 3.5).

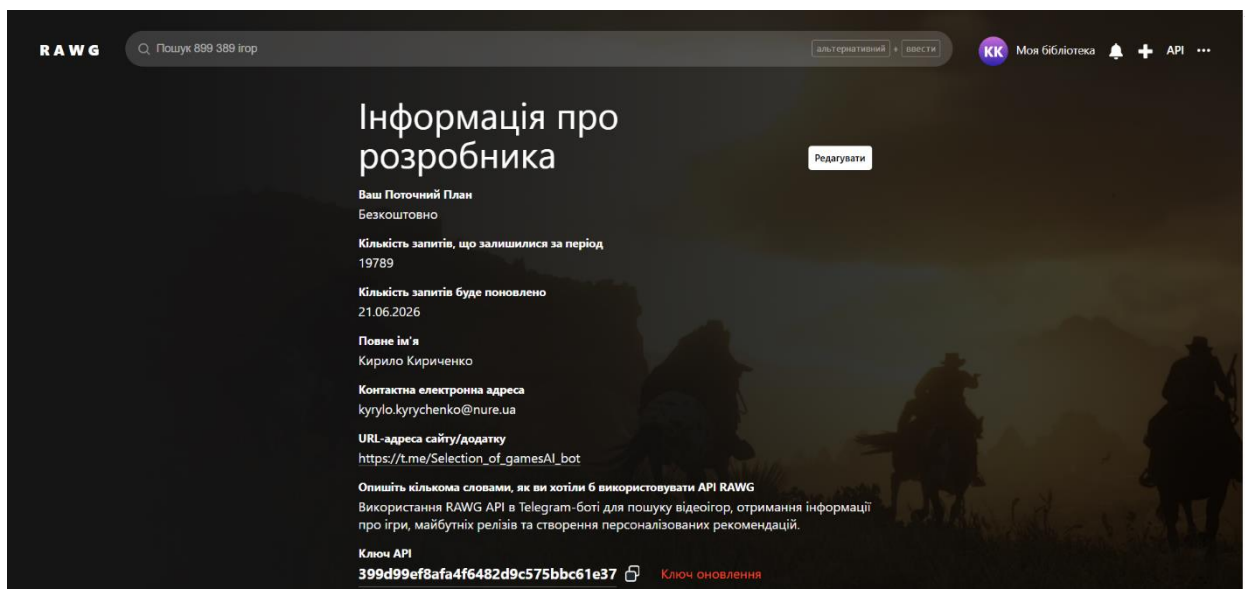
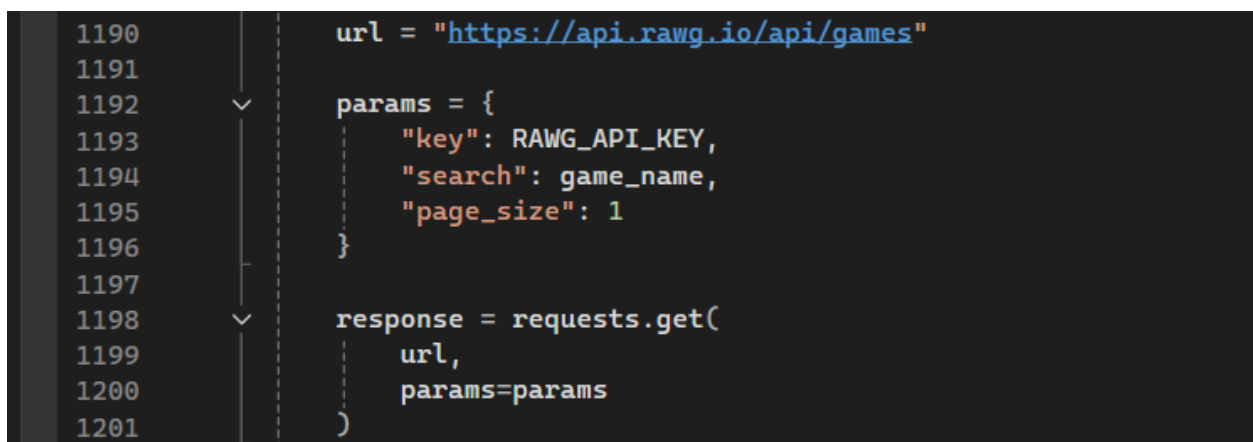


Рисунок 3.5 – Інформація про розробника на сайті RAWG API

Взаємодія із сервісом здійснюється шляхом надсилання HTTP-запитів до API з використанням бібліотеки `requests`. Користувач може скористатися функцією «Пошук гри», після чого бот переходить у режим очікування введення назви гри. Отримана від користувача назва використовується для формування запиту до сервісу RAWG API.

Після надсилання запиту сервіс повертає інформацію про знайдені ігри у форматі JSON. Отримані дані аналізуються програмою, після чого обирається найбільш релевантний результат пошуку (рис. 3.6).



```
1190 url = "https://api.rawg.io/api/games"
1191
1192     params = {
1193         "key": RAWG_API_KEY,
1194         "search": game_name,
1195         "page_size": 1
1196     }
1197
1198     response = requests.get(
1199         url,
1200         params=params
1201     )
```

Рисунок 3.6 – Реалізація пошуку гри за назвою через RAWG API

Після успішного пошуку бот формує інформаційну картку гри. До картки включаються такі характеристики:

- назва гри;
- рейтинг користувачів;
- оцінка Metacritic;
- дата виходу;
- жанри;
- підтримувані платформи;
- поточна ціна та інформація про магазин;
- короткий опис гри;
- зображення обкладинки.

Для підвищення зручності користувача короткий опис гри автоматично перекладається українською мовою та адаптується для відображення у форматі повідомлення Telegram (рис. 3.7).

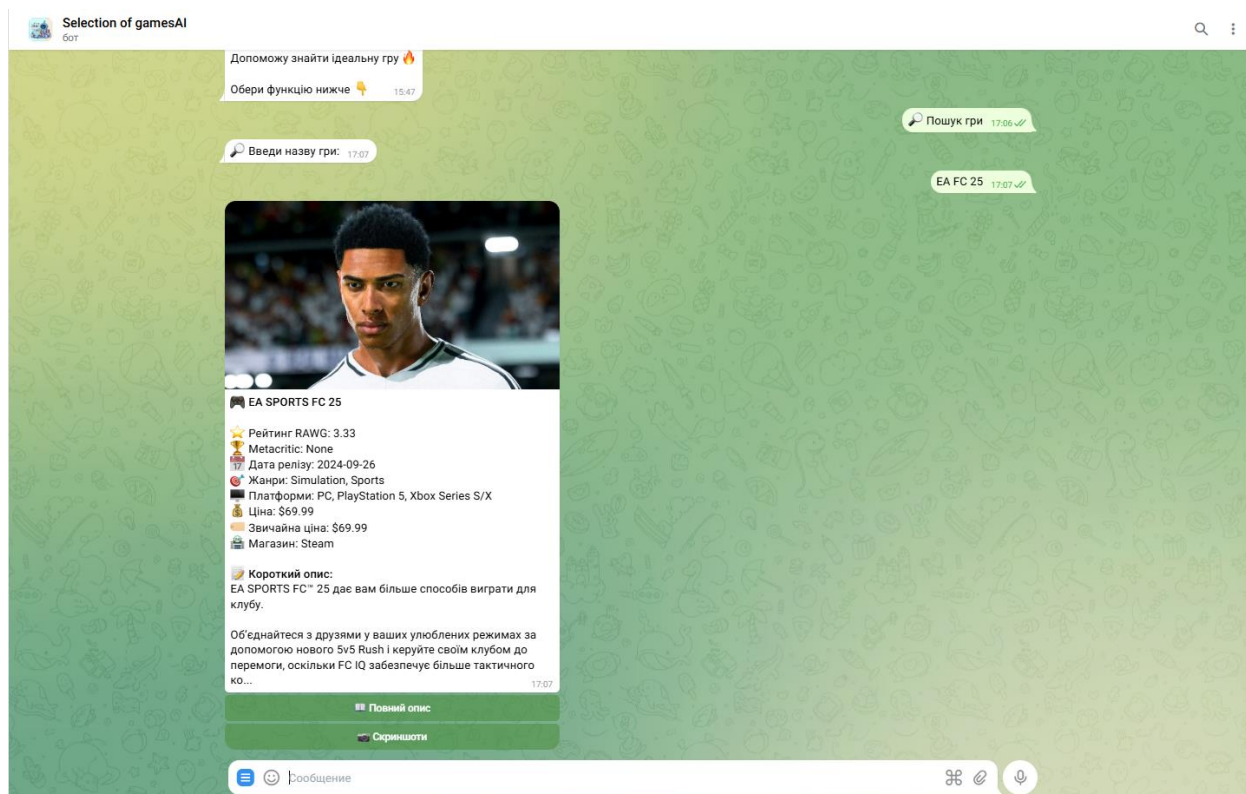


Рисунок 3.7 – Приклад відображення інформації про відеогру

Крім основної інформації про гру, реалізовано можливість перегляду повного опису. Для цього після формування інформаційної картки користувачу відображається спеціальна кнопка «Повний опис». При натисканні на дану кнопку бот отримує раніше збережений текст опису гри та надсилає його окремим повідомленням у чаті. Реалізація даної функції дозволяє розділити коротку та детальну інформацію про відеогру, що покращує зручність користування системою. Основна картка містить лише найважливіші характеристики, такі як назва, рейтинг, жанри, дата виходу та короткий опис, тоді як повний опис надає більш детальну інформацію про сюжет, особливості ігрового процесу, механіки та інші характеристики гри. Для забезпечення комфортного сприйняття тексту виконується попередня обробка отриманого

опису. Зокрема, видаляються зайві службові символи та елементи форматування, після чого текст перекладається українською мовою. Це дозволяє користувачам отримувати повну інформацію про гру рідною мовою незалежно від мови оригінального опису. Реалізований механізм перегляду повного опису дає змогу уникнути перевантаження основної картки гри надлишковою інформацією та забезпечує більш зручну взаємодію користувача з Telegram-ботом. Приклад роботи функції перегляду повного опису представлено на рисунку 3.8.

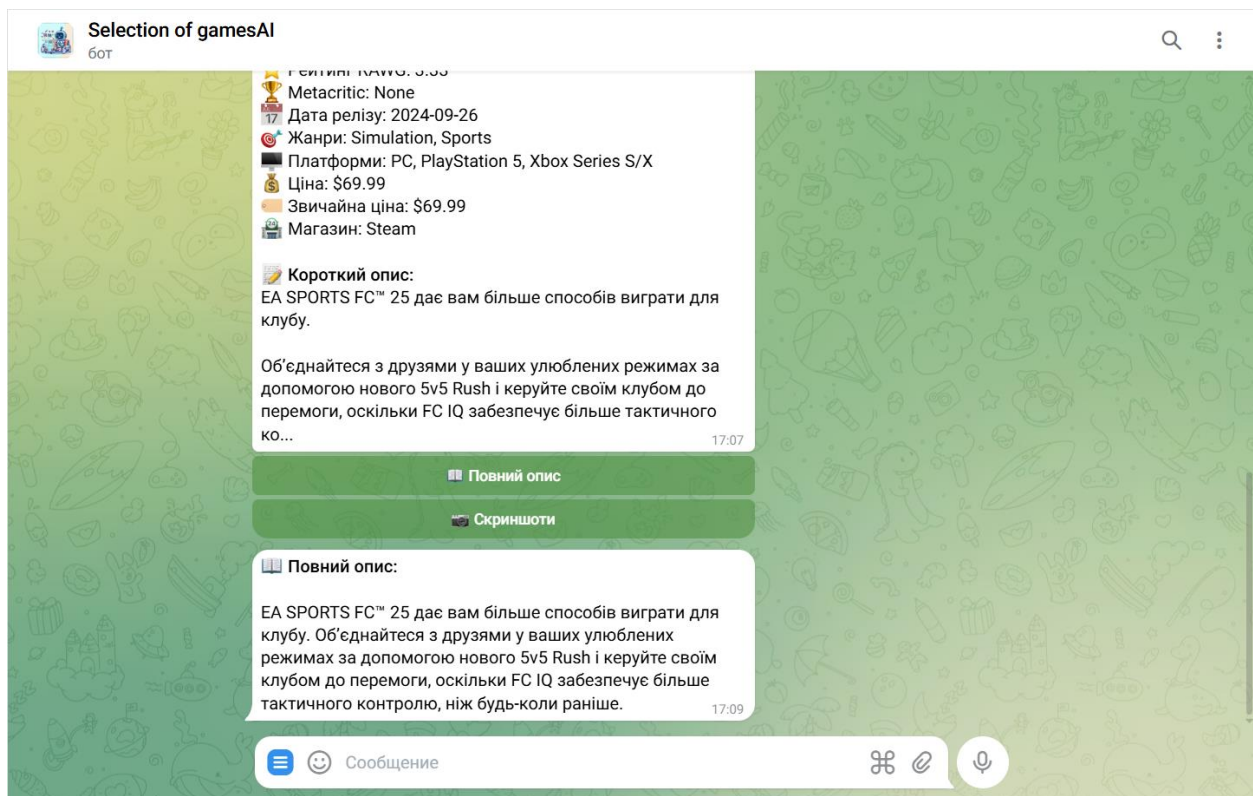


Рисунок 3.8 – Перегляд повного опису гри

Для підвищення інформативності системи також було реалізовано можливість перегляду скріншотів гри. Після вибору відповідної кнопки бот виконує додатковий запит до RAWG API та отримує перелік доступних зображень, пов'язаних із вибраною відеогрою. У відповіді сервісу містяться посилання на офіційні скріншоти, які демонструють ігровий процес, графічне оформлення та особливості віртуального середовища. Отримані дані обробляються програмою, після чого скріншоти надсилаються користувачу у

вигляді галереї зображень у чаті Telegram. Такий підхід дозволяє користувачу швидко ознайомитися з візуальними особливостями гри без необхідності переходу на сторонні вебресурси або пошуку додаткової інформації в Інтернеті. Використання скриншотів є важливим елементом інформаційної підтримки користувача, оскільки графічне представлення гри часто суттєво впливає на рішення щодо її вибору. Завдяки перегляду зображень користувач може оцінити рівень графіки, художній стиль, інтерфейс та загальну атмосферу гри ще до її придбання або встановлення. Ба більше, автоматичне отримання скриншотів через RAWG API забезпечує актуальність та достовірність відображуваних матеріалів. Усі зображення отримуються безпосередньо із зовнішньої бази даних сервісу, що виключає необхідність ручного оновлення графічного контенту (рис. 3.9).

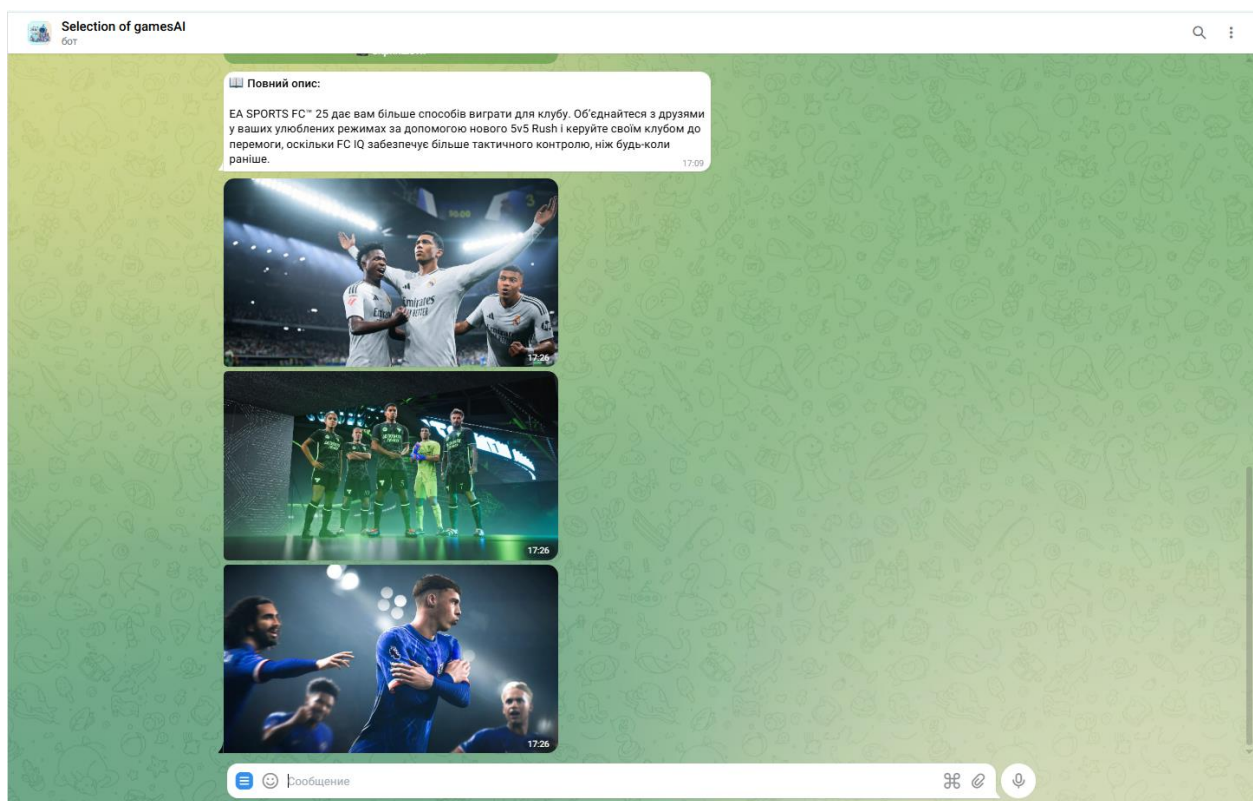


Рисунок 3.9 – Відображення скриншотів гри

Окрім пошуку окремих ігор, використання RAWG API дозволило реалізувати додаткові функції системи. До них належать отримання випадкової

гри, перегляд календаря майбутніх релізів, визначення дати виходу конкретної гри, а також формування тематичних рейтингів за жанрами.

Завдяки використанню RAWG API було забезпечено автоматичне отримання актуальної інформації про відеоігри без необхідності створення та підтримки власної бази даних. Це дозволило значно розширити функціональні можливості Telegram-бота та забезпечити користувачів повною і достовірною інформацією про сучасні відеоігри.

3.4 Реалізація системи рекомендацій на основі OpenAI API

Однією з основних особливостей розробленого Telegram-бота є наявність підсистеми інтелектуальних рекомендацій, яка дозволяє користувачам отримувати персоналізовані рекомендації відеоігор відповідно до їхніх вподобань. На відміну від звичайного пошуку, де користувач повинен самостійно визначитися з конкретною назвою гри, дана функція дозволяє отримувати рекомендації навіть у випадках, коли користувач не знає, у що саме хоче пограти.

Для реалізації даного функціоналу було використано OpenAI API, який надає доступ до сучасних великих мовних моделей. Використання технологій штучного інтелекту дозволяє аналізувати текстові запити природною мовою та визначати основні характеристики бажаної гри. Це значно розширює можливості взаємодії користувача із системою та робить процес пошуку ігор більш гнучким і зручним (рис. 3.10).

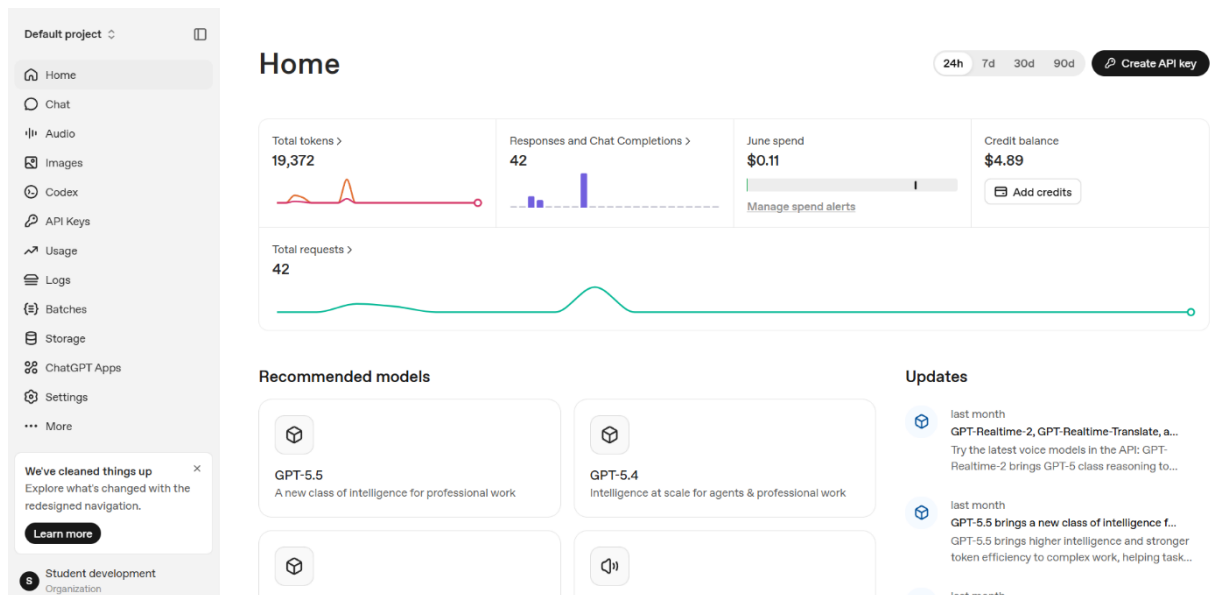


Рисунок 3.10 – Отримання API ключа від OpenAI

На відміну від класичних рекомендаційних систем, які працюють на основі фільтрів або жорстко заданих категорій, реалізована система дозволяє користувачу формулювати свої побажання у довільній формі. Наприклад, користувач може вказати бажаний жанр, тематику гри, особливості сюжету, тип ігрового процесу або навіть загальну атмосферу, яку він хоче отримати від гри. Завдяки цьому рекомендації формуються більш природним способом та максимально наближені до реальних потреб користувача.

Для запуску процесу формування рекомендацій користувач обирає пункт «AI рекомендація» у головному меню бота. Після цього система переводить користувача у спеціальний режим введення запиту та пропонує описати свої побажання щодо майбутньої гри. Для реалізації даного механізму використовується система станів FSM, що підтримується бібліотекою aiogram (рис. 3.11).

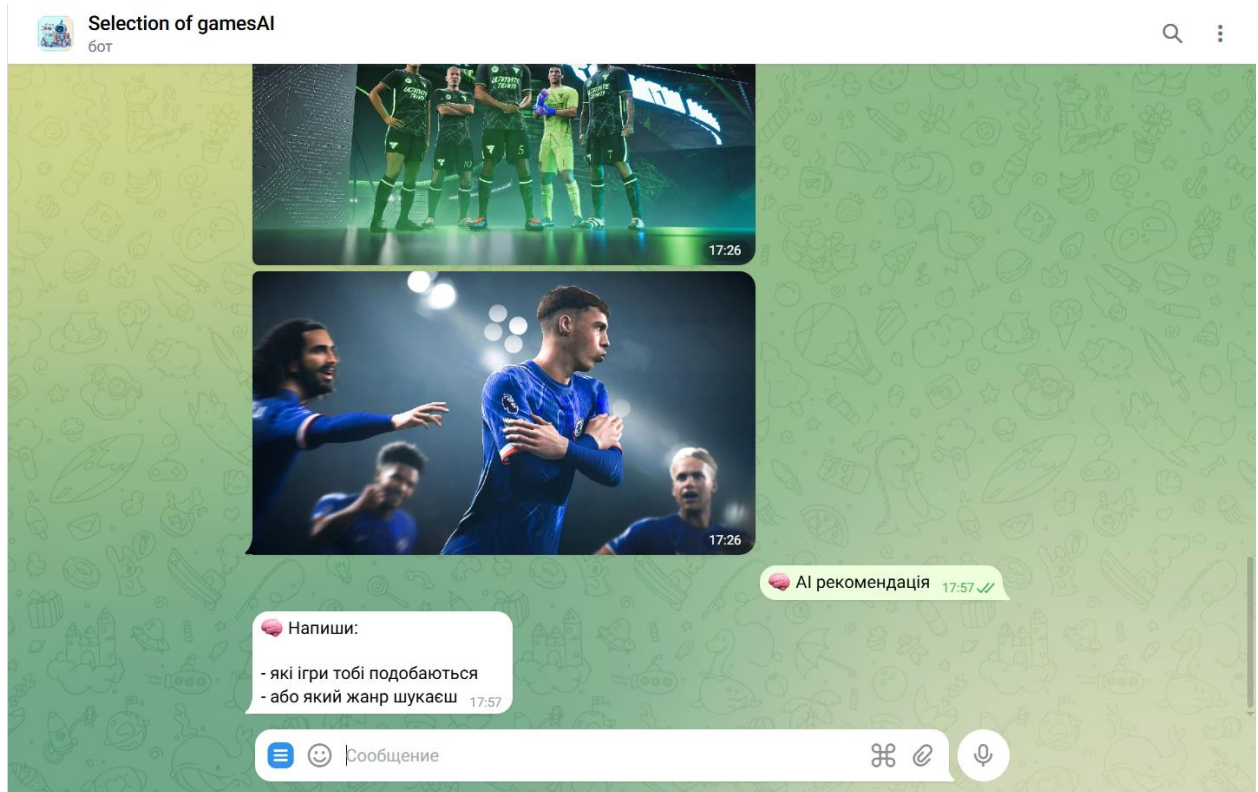


Рисунок 3.11 – Перехід користувача до режиму отримання рекомендацій

Після отримання текстового повідомлення від користувача система виконує його попередню обробку та формує спеціальний запит для великої мовної моделі. У сформованому промті визначаються вимоги до структури відповіді, кількості рекомендованих ігор та формату їх представлення. Завдяки цьому забезпечується отримання впорядкованого та зрозумілого результату.

На наступному етапі сформований запит передається до OpenAI API. Після виконання аналізу модель генерує список рекомендованих відеоігор, які найбільше відповідають зазначеним користувачем вимогам. Отримана відповідь містить назви ігор та короткий опис причин, через які саме ці ігри були рекомендовані (рис. 3.12).

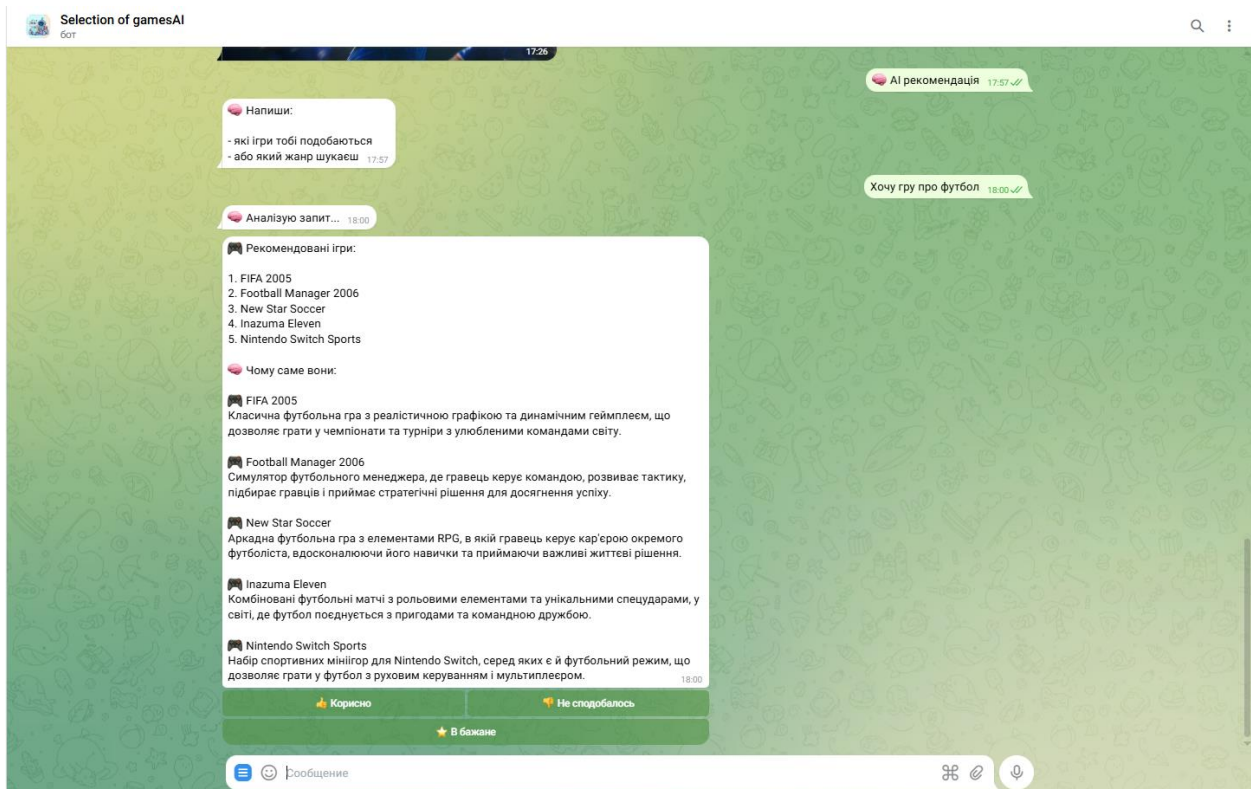


Рисунок 3.12 – Формування запиту до OpenAI API

Отримані результати проходять додаткову обробку програмою та відображаються користувачу у вигляді структурованого повідомлення. Для покращення читабельності між окремими рекомендаціями автоматично додаються відступи, що дозволяє чітко розділити інформацію про кожну запропоновану гру.

Важливою перевагою використання великих мовних моделей є здатність враховувати складні текстові описи. Наприклад, користувач може вказати, що його цікавлять пригодницькі ігри з відкритим світом, сюжетною складовою та можливістю дослідження локацій. У такому випадку система аналізує всі зазначені характеристики одночасно та формує рекомендації на основі їх сукупності.

Крім формування рекомендацій, реалізовано механізм оцінювання результатів роботи системи. Після отримання рекомендації користувач може поставити позитивну або негативну оцінку за допомогою відповідних кнопок.

Інформація про оцінки зберігається у базі даних та використовується для ведення статистики користувача (рис. 3.13).

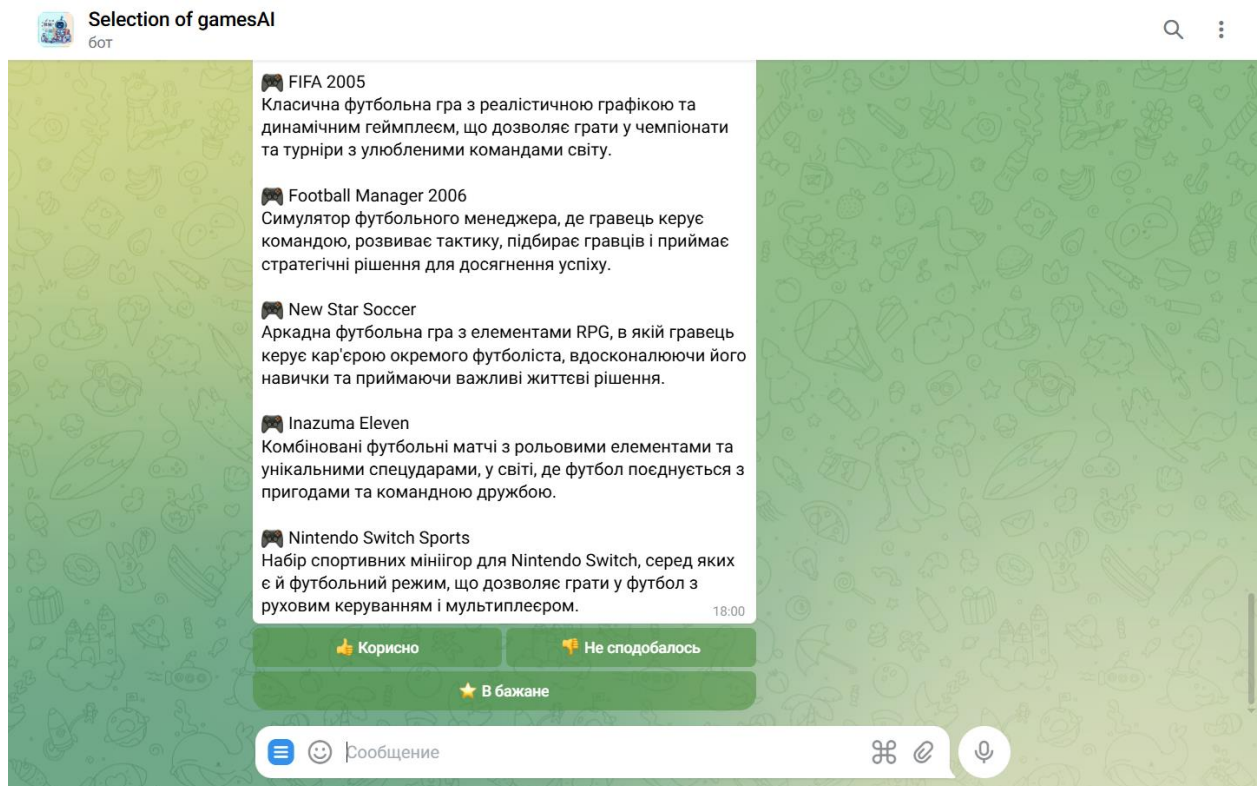


Рисунок 3.13 – Оцінювання рекомендацій користувачем

Використання механізму оцінювання дозволяє аналізувати рівень задоволеності користувачів отриманими рекомендаціями та здійснювати моніторинг якості роботи системи. Інформація про позитивні та негативні оцінки використовується для формування профілю користувача та відображення статистичних показників у відповідному розділі бота.

Реалізована підсистема рекомендацій є одним із ключових компонентів розробленого Telegram-бота. Її використання дозволяє значно спростити процес пошуку нових відеоігор, забезпечити індивідуальний підхід до кожного користувача та підвищити загальну зручність використання програмного продукту. Завдяки інтеграції OpenAI API було реалізовано сучасний механізм взаємодії з користувачем, який забезпечує формування рекомендацій на основі

природної мови та наближає систему до інтелектуальних рекомендаційних сервісів нового покоління.

3.5 Реалізація підсистеми пошуку знижок на відеоігри за допомогою CheapShark API

Однією з додаткових функцій розробленого Telegram-бота є пошук актуальних знижок на відеоігри. Даний функціонал дозволяє користувачам оперативно отримувати інформацію про акційні пропозиції та знижені ціни на ігри в цифрових магазинах. Для реалізації цієї можливості було використано сервіс CheapShark API, який надає інформацію про поточні акції та спеціальні пропозиції на різних ігрових майданчиках.

Використання CheapShark API дозволяє отримувати дані про назву гри, початкову ціну, поточну акційну ціну, відсоток знижки та магазин, у якому доступна відповідна пропозиція. Отримана інформація автоматично обробляється ботом та відображається користувачу у зручному для сприйняття вигляді.

Для доступу до функціоналу перегляду акційних пропозицій у головному меню було реалізовано кнопку «Знижки». Після її натискання бот виконує запит до CheapShark API та отримує перелік актуальних пропозицій. З отриманого списку випадковим чином обирається декілька ігор, що дозволяє урізноманітнити результати при кожному новому зверненні до системи.

Реалізація механізму випадкового вибору дозволяє уникнути постійного відображення одних і тих самих ігор та робить використання функції більш цікавим для користувачів. При кожному натисканні кнопки формується нова добірка акційних пропозицій із загального списку доступних знижок. Завдяки цьому користувач отримує можливість ознайомитися з більшою кількістю ігор та акційних пропозицій, що підвищує практичну цінність функції та стимулює повторне використання бота.

Використання випадкового відбору дозволяє рівномірно демонструвати різні категорії ігор, серед яких можуть бути проєкти різних жанрів, платформ та цінових категорій. Такий підхід забезпечує різноманітність контенту та збільшує ймовірність того, що користувач знайде цікаву для себе пропозицію.

Після отримання відповіді від сервісу CheapShark API виконується аналіз отриманих даних та формування інформаційних карток. На даному етапі система обробляє отримані параметри, перевіряє коректність значень та виконує форматування інформації для подальшого відображення в інтерфейсі Telegram.

Для кожної гри відображаються основні відомості про акційну пропозицію, включаючи назву гри, відсоток знижки, початкову ціну, поточну акційну ціну та назву магазину. Наявність інформації про стару та нову ціну дозволяє користувачу швидко оцінити вигідність пропозиції та визначити розмір економії. Додатково відображається магазин, у якому доступна відповідна знижка, що спрощує подальший перехід до придбання гри.

Результати пошуку подаються у вигляді окремих повідомлень із чіткою структурою, що забезпечує зручність сприйняття інформації навіть при перегляді великої кількості акційних пропозицій. Приклад відображення інформації про акційні пропозиції представлено на рис. 3.14–3.15.

Крім перегляду випадкових акційних пропозицій, у системі реалізовано можливість пошуку знижки для конкретної гри. Для цього користувач обирає пункт меню «Знижка на гру», після чого бот переходить у режим очікування введення назви гри (рис. 3.16).

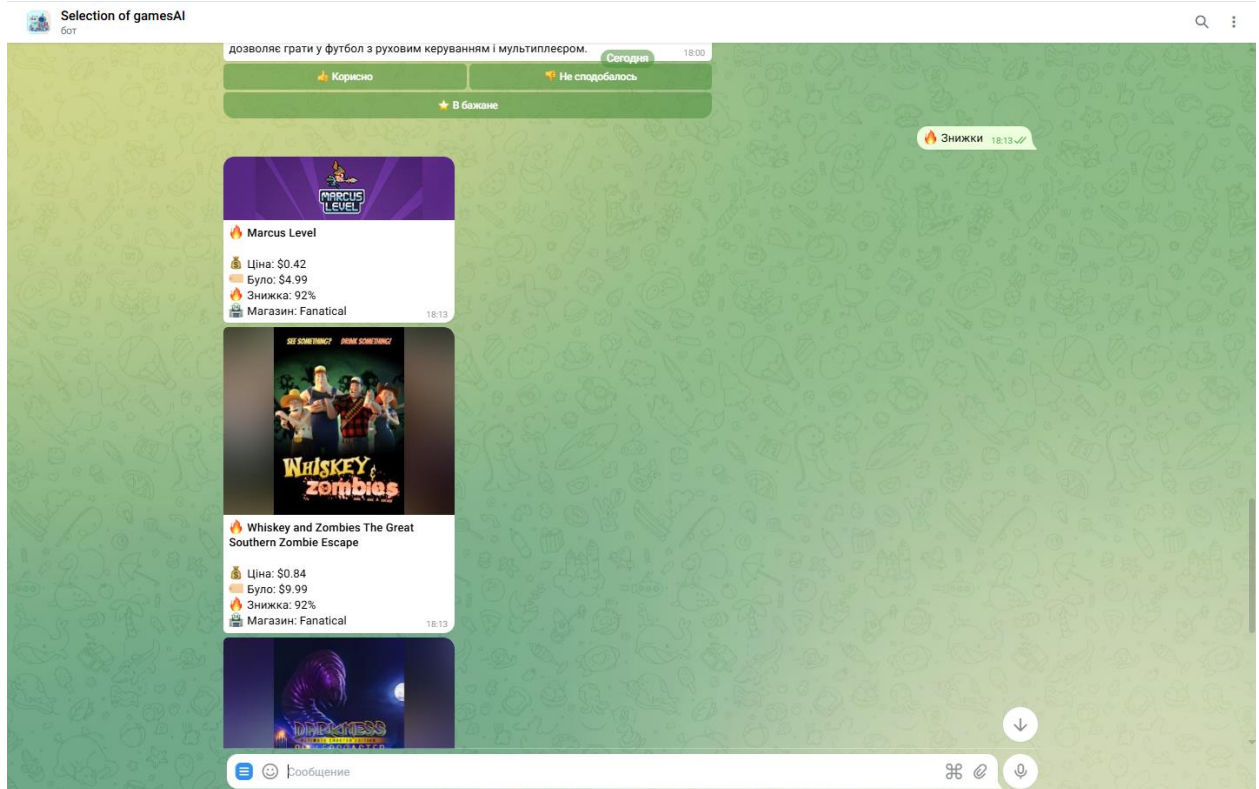


Рисунок 3.14 – Отримання випадкових акційних пропозицій

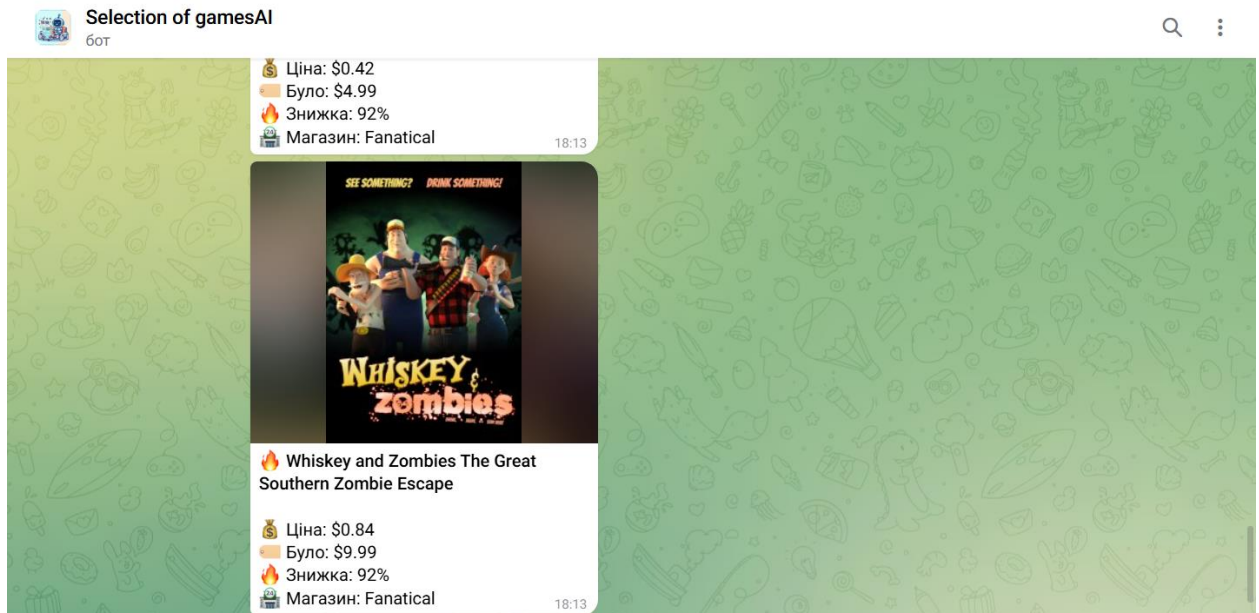


Рисунок 3.15 – Відображення інформації про знижку на гру



Рисунок 3.16 – Перехід до режиму пошуку знижки на гру

Після введення назви гри формується запит до CheapShark API, який здійснює пошук відповідних пропозицій серед доступних магазинів. Якщо знижка на вказану гру знайдена, користувач отримує детальну інформацію про акційну пропозицію. У випадку відсутності знижок система повідомляє користувача про те, що на даний момент відповідних пропозицій не знайдено (рис.3.17).

Для підвищення інформативності реалізовано механізм відображення назви магазину, у якому доступна акційна пропозиція. У випадку відсутності назви магазину в отриманих даних система відображає його ідентифікатор, що дозволяє уникнути появи порожніх або некоректних значень у повідомленнях бота.

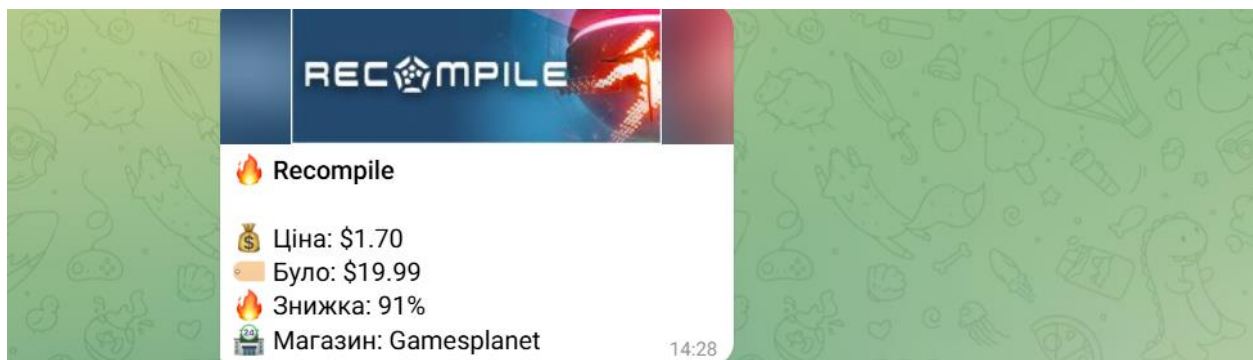


Рисунок 3.17 – Результат пошуку знижки для конкретної гри

Використання CheapShark API дозволило реалізувати ефективний механізм моніторингу цін на відеоігри та надати користувачам додатковий інструмент для пошуку вигідних пропозицій. Завдяки інтеграції із зовнішнім сервісом інформація про знижки постійно оновлюється, що забезпечує

актуальність отримуваних даних та підвищує практичну цінність розробленого Telegram-бота.

3.6 Реалізація бази даних користувачів на основі SQLite

Для забезпечення збереження даних користувачів у розробленому Telegram-боті було використано систему керування базами даних SQLite. Вибір даної СКБД обумовлений її простотою використання, високою швидкістю роботи та відсутністю необхідності встановлення окремого серверного програмного забезпечення. SQLite є оптимальним рішенням для невеликих програмних систем, що працюють із відносно невеликими обсягами даних та не потребують складної серверної інфраструктури.

У межах розробленого програмного продукту база даних використовується для збереження інформації про дії користувачів, їхні оцінки отриманих рекомендацій та список бажаних ігор. Використання бази даних дозволяє забезпечити персоналізацію роботи бота та зберігати інформацію між окремими сеансами взаємодії користувача із системою.

Для роботи з базою даних було створено окремий програмний модуль `database.py`, який містить набір функцій для створення таблиць, запису інформації та отримання збережених даних. Такий підхід дозволяє відокремити логіку роботи з базою даних від основної логіки Telegram-бота та спрощує подальшу підтримку програмного коду. Завдяки модульній структурі програми всі операції з базою даних виконуються через окремі функції, що підвищує читабельність коду та спрощує його подальшу модернізацію.

У модулі реалізовано функції для ініціалізації структури бази даних, збереження історії взаємодії користувачів із системою, запису оцінок отриманих рекомендацій, роботи зі списком бажаних ігор, а також отримання та видалення раніше збережених записів. Кожна функція виконує окреме завдання та забезпечує взаємодію із відповідними таблицями бази даних.

Використання окремого програмного модуля дозволило уникнути дублювання коду в основному файлі Telegram-бота та забезпечило централізоване керування всіма операціями збереження даних. Крім того, такий підхід відповідає сучасним принципам розроблення програмного забезпечення, згідно з якими різні функціональні компоненти системи повинні бути логічно відокремлені один від одного.

Реалізована структура бази даних забезпечує збереження інформації навіть після завершення роботи програми або перезапуску Telegram-бота. Це дозволяє накопичувати історію взаємодії користувачів із системою, формувати персональні списки бажаних ігор та використовувати збережені дані для подальшої персоналізації рекомендацій (рис. 3.17).

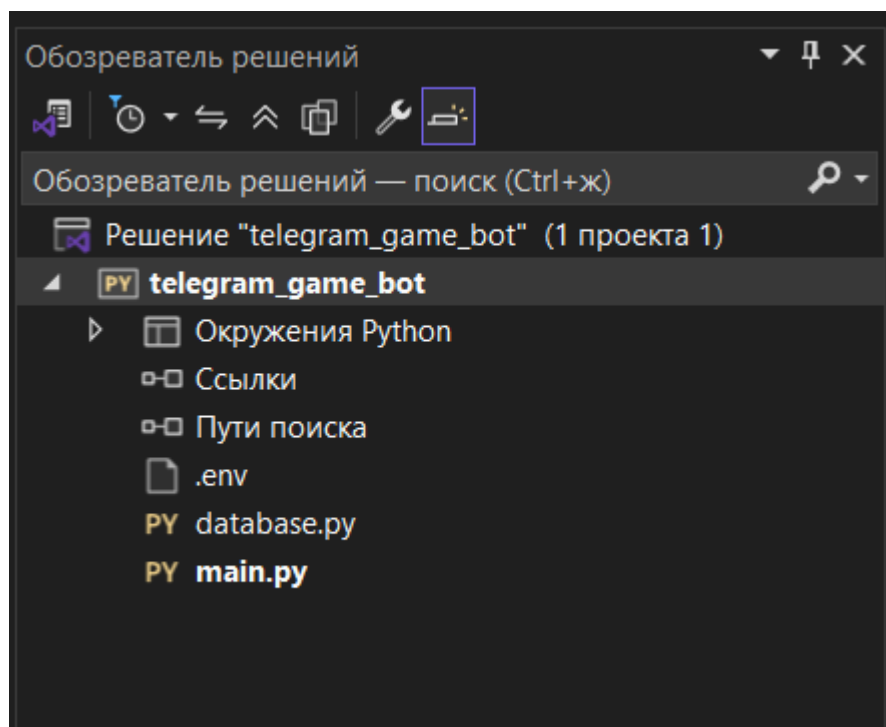


Рисунок 3.17 – Програмний модуль database.py

На початковому етапі роботи системи виконується створення необхідних таблиць бази даних. Для цього використовується функція `init_db()`, яка автоматично перевіряє наявність таблиць та створює їх у випадку відсутності.

Такий підхід дозволяє підготувати базу даних до роботи без додаткових дій з боку користувача.

Під час запуску програми функція створює таблиці для зберігання історії взаємодії користувачів, оцінок рекомендацій та списку бажаних ігор. Ще для таблиці історії створюється індекс, що підвищує швидкість обробки запитів до бази даних.

Використання функції `init_db()` забезпечує автоматичне формування структури бази даних та спрощує розгортання програмного продукту на інших пристроях (рис. 3.18).

```
3
4  def init_db():
5      conn = sqlite3.connect("users.db")
6      cursor = conn.cursor()
7
8      cursor.execute("""
9          CREATE TABLE IF NOT EXISTS user_history (
10             user_id INTEGER,
11             game_name TEXT
12          )
13      """)
14
15      cursor.execute("""
16          CREATE TABLE IF NOT EXISTS user_feedback (
17             user_id INTEGER,
18             game_name TEXT,
19             feedback TEXT
20          )
21      """)
22
23      cursor.execute("""
24          CREATE TABLE IF NOT EXISTS wishlist (
25             user_id INTEGER,
26             game_name TEXT
27          )
28      """)
```

Рисунок 3.18 – Ініціалізація структури бази даних SQLite

У розробленій системі реалізовано три основні таблиці: `user_history`, `user_feedback` та `wishlist`. Кожна з них виконує окрему функцію та забезпечує збереження інформації, необхідної для персоналізації роботи Telegram-бота. Використання декількох взаємопов'язаних таблиць дозволяє структурувати дані

користувачів, спростити їх обробку та забезпечити ефективний доступ до необхідної інформації.

Таблиця `user_history` (табл. 3.1) призначена для збереження історії взаємодії користувача із системою. У ній зберігаються ідентифікатор користувача та назва гри, інформацію про яку було отримано через Telegram-бота. Кожен запис формується автоматично після успішного пошуку або отримання інформації про відеогру. Таким чином, база даних накопичує інформацію про інтереси користувача та його активність у системі.

Наявність історії переглянутих ігор дозволяє не лише зберігати інформацію про попередні запити, а й використовувати її для аналізу вподобань користувача. На основі таких даних у подальшому можуть бути реалізовані більш складні механізми персоналізації рекомендацій та формування індивідуальних добірок відеоігор.

Використання таблиці `user_history` підвищує зручність роботи із системою, оскільки користувач може переглядати інформацію про раніше знайдені ігри без необхідності повторного введення пошукових запитів. Збереження історії також дозволяє оцінювати популярність окремих функцій бота та аналізувати характер взаємодії користувачів із програмним продуктом.

Таблиця 3.1 – Структура таблиці `user_history`

Поле	Тип даних	Призначення
<code>user_id</code>	INTEGER	Ідентифікатор користувача Telegram
<code>game_name</code>	TEXT	Назва гри, інформацію про яку переглядав користувач

Друга таблиця `user_feedback` (табл. 3.2) використовується для збереження оцінок користувача щодо отриманих рекомендацій. У таблиці зберігається інформація про гру та тип оцінки, яку поставив користувач. Як оцінки використовуються значення «like» та «dislike».

Накопичення таких даних дозволяє аналізувати вподобання користувача та визначати його ставлення до запропонованих відеоігор. Отримана інформація може використовуватися для подальшої персоналізації рекомендацій та підвищення якості роботи системи.

Таблиця 3.2 – Структура таблиці user_feedback

Поле	Тип даних	Призначення
user_id	INTEGER	Ідентифікатор користувача Telegram
game_name	TEXT	Назва гри, для якої було залишено оцінку
feedback	TEXT	Тип оцінки користувача (like або dislike)

Збереження оцінок дозволяє формувати статистику користувача та відображати інформацію про кількість позитивних і негативних реакцій у профілі.

Третьою таблицею є wishlist (табл. 3.3), яка використовується для реалізації списку бажаних ігор. Користувач може додавати цікаві для себе ігри до даного списку та переглядати їх у будь-який момент часу.

Таблиця 3.3 – Структура таблиці wishlist

Поле	Тип даних	Призначення
user_id	INTEGER	Ідентифікатор користувача Telegram
game_name	TEXT	Назва гри, доданої до списку бажаного

Під час додавання гри до списку бажаного система виконує перевірку на наявність дублюючих записів. Якщо така гра вже присутня у списку

користувача, новий запис не створюється. Це дозволяє підтримувати цілісність даних та уникати дублювання інформації.

Для підвищення продуктивності роботи системи було створено індекс для поля `user_id` таблиці `user_history`. Використання індексування дозволяє прискорити виконання запитів під час отримання історії переглянутих ігор конкретного користувача.

Реалізована структура бази даних забезпечує ефективне збереження інформації про користувачів та їхню взаємодію із системою. Використання SQLite дозволило створити просту та надійну систему зберігання даних, яка не потребує додаткових серверних ресурсів і повністю задовольняє вимоги розробленого Telegram-бота.

3.7 Тестування програмного продукту та аналіз результатів

Після завершення розроблення Telegram-бота було проведено комплексне тестування реалізованого програмного забезпечення. Основною метою тестування була перевірка коректності роботи всіх функціональних модулів системи, оцінка стабільності взаємодії із зовнішніми сервісами та підтвердження відповідності отриманих результатів поставленим вимогам.

На попередніх етапах розроблення було реалізовано функції пошуку інформації про відеоігри, перегляду повного опису, отримання скріншотів, формування рекомендацій на основі штучного інтелекту, пошуку актуальних знижок та зберігання даних користувачів у базі SQLite. Під час тестування було підтверджено коректну роботу зазначених функцій та успішну інтеграцію із зовнішніми API.

Окрему увагу було приділено перевірці додаткових можливостей системи. Однією з таких функцій є список бажаних ігор (Wishlist), який дозволяє користувачам зберігати цікаві для них ігри для подальшого перегляду. Наявність даного функціоналу підвищує зручність використання Telegram-бота, оскільки

користувач може формувати власну колекцію відеоігор та повертатися до неї у будь-який момент часу.

Під час тестування було перевірено процес додавання гри до списку бажаного, відображення збережених записів, а також механізм запобігання дублюванню даних. Для цього до списку додавалися різні відеоігри, після чого виконувалася перевірка їх наявності у відповідному розділі бота. Додатково було здійснено спробу повторного додавання вже існуючої гри, що дозволило перевірити роботу механізму контролю дублювання записів.

Результати тестування показали, що після додавання гри інформація коректно зберігається у базі даних SQLite та відображається користувачу під час перегляду списку бажаного. У випадку повторного додавання тієї самої гри система успішно запобігала створенню дублюючого запису, що підтвердило правильність роботи реалізованого алгоритму перевірки даних.

Ще було перевірено стабільність роботи функції при додаванні декількох ігор поспіль та під час повторного запуску Telegram-бота. Результати підтвердили, що всі збережені дані залишаються доступними після завершення роботи програми, а механізм взаємодії з базою даних функціонує коректно. Отримані результати підтвердили правильність роботи функції Wishlist та надійність реалізованої системи збереження даних користувачів (рис. 3.19).

Наступним етапом було протестовано функцію профілю користувача (рис. 3.20). Даний модуль призначений для відображення статистичної інформації, сформованої на основі взаємодії користувача із системою. Під час тестування було перевірено коректність відображення кількості позитивних та негативних оцінок, інформації про список бажаних ігор, а також інших показників активності користувача. Результати тестування підтвердили правильність роботи механізмів збору та обробки статистичних даних.

Для перевірки роботи тематичних добірок було проведено тестування категорії RPG. Дана категорія призначена для відображення рольових відеоігор, які є одними з найпопулярніших серед користувачів ігрових платформ. Під час тестування система успішно формувала список рольових ігор на основі даних,

отриманих через RAWG API. До результатів потрапляли популярні представники жанру, які супроводжувалися короткою інформацією про рейтинг, дату виходу, жанри та інші характеристики (рис. 3.21).

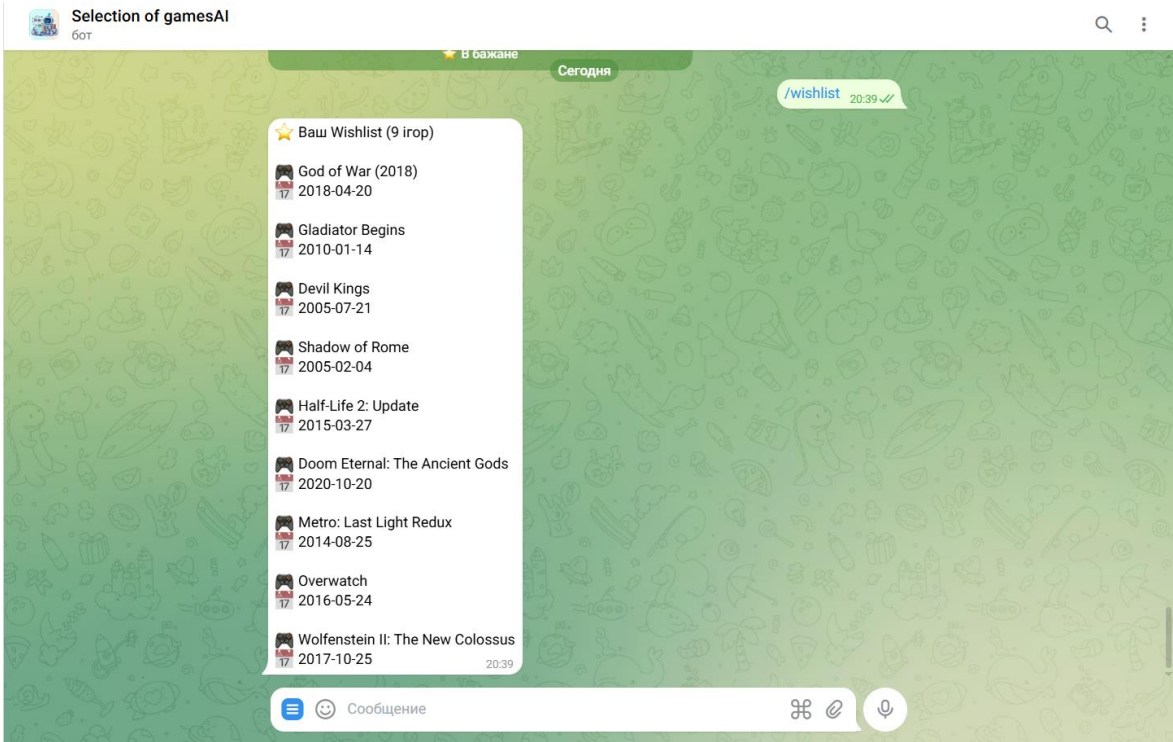


Рисунок 3.19 – Результат тестування функції Wishlist

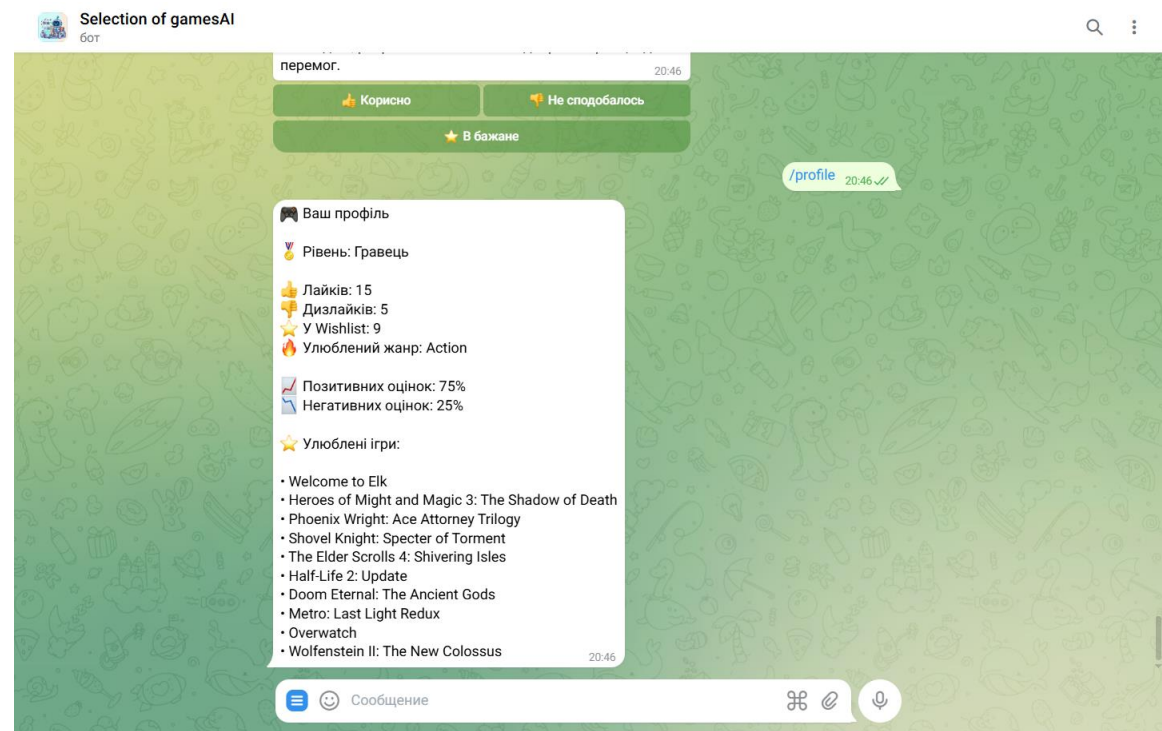


Рисунок 3.20 – Результат тестування профілю користувача

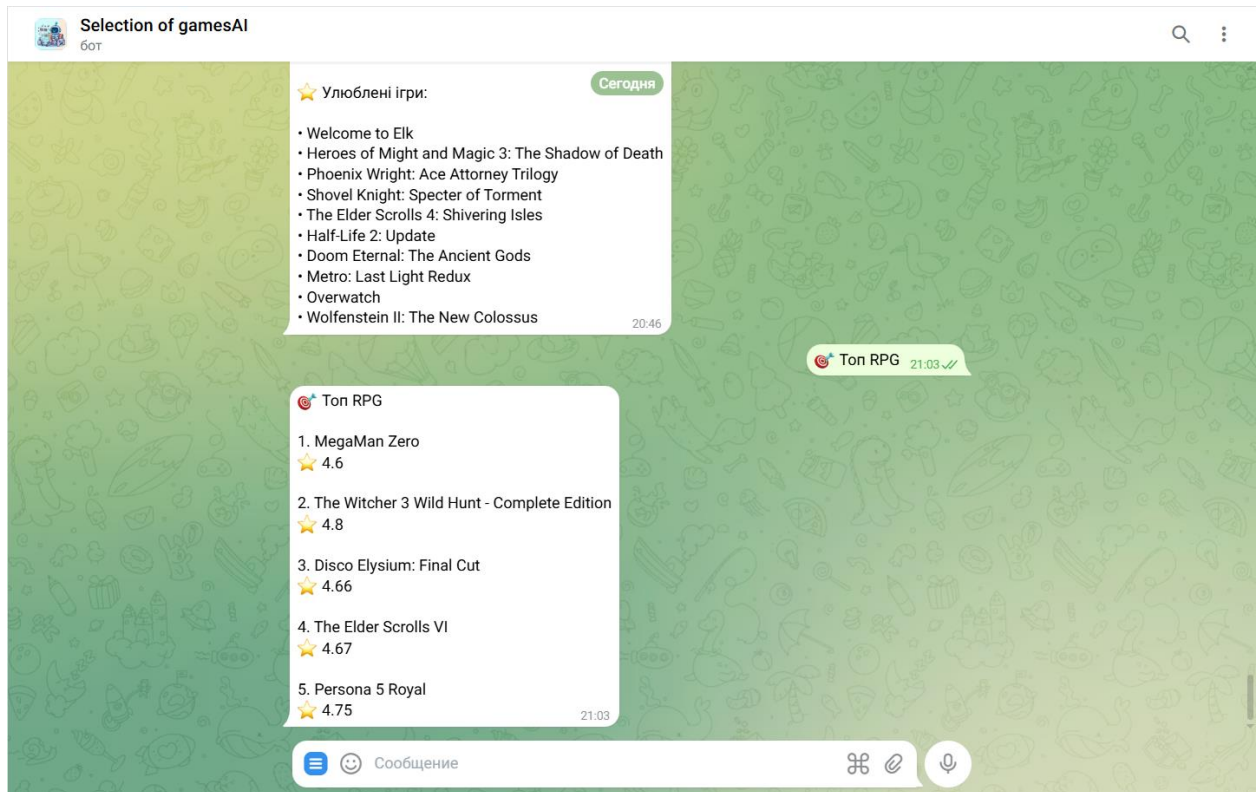


Рисунок 3.21 – Результат тестування добірки Top RPG

Під час перевірки особлива увага приділялася правильності відбору ігор відповідно до обраної категорії. Було підтверджено, що система відображає саме рольові відеоігри та не включає проекти інших жанрів. Було перевірено механізм випадкового формування добірки, який забезпечує різноманітність результатів при повторному використанні функції. Отримані результати підтвердили коректність роботи механізму формування тематичних добірок для рольових відеоігор та відповідність отриманих даних заданим критеріям пошуку.

Окремо було протестовано категорію Shooter, яка призначена для відображення популярних шутерів (рис.3.22). Даний жанр об'єднує велику кількість відеоігор, орієнтованих на динамічний ігровий процес, використання різноманітної зброї та змагальні механіки. Під час перевірки система коректно отримувала та відображала інформацію про ігри даного жанру, використовуючи дані зовнішнього сервісу RAWG API.

У процесі тестування було виконано декілька послідовних запитів до категорії Shooter для перевірки стабільності роботи функції. Результати

показали, що добірка формується відповідно до обраного жанру, а користувач отримує актуальну інформацію про найбільш популярні шутери. Також було підтверджено правильність відображення основних характеристик ігор та коректність роботи механізму випадкового відбору. Завдяки цьому при повторному використанні функції користувач може отримувати нові рекомендації, що підвищує інтерес до використання системи.

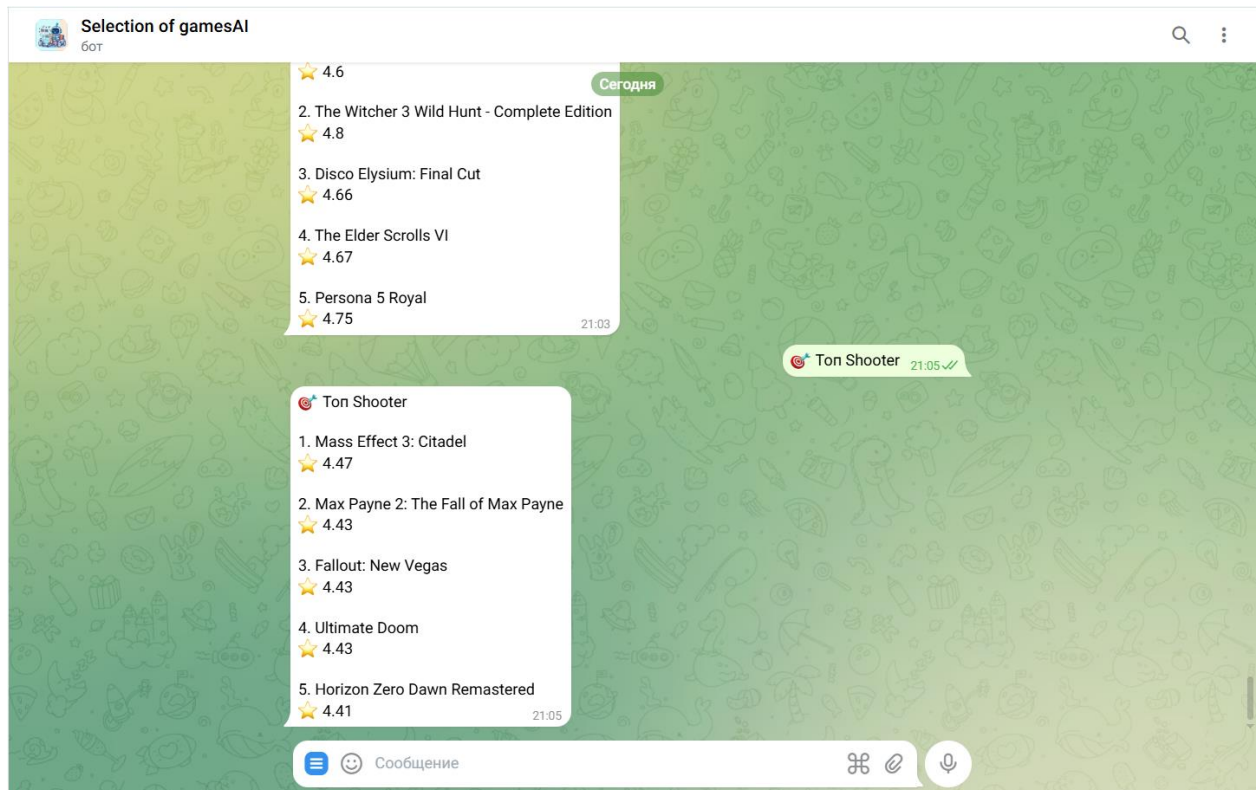


Рисунок 3.22 – Результат тестування добірки Top Shooter

Наступним етапом було протестовано категорію Sports. Даний розділ містить спортивні відеоігри різних напрямів, зокрема футбольні, баскетбольні, хокейні, тенісні та інші спортивні симулятори. Основною метою тестування було підтвердження правильності відбору спортивних ігор та перевірка якості відображення отриманої інформації (рис.3.23).

У ході тестування система успішно формувала добірку ігор відповідного жанру та відображала її користувачу у зручному форматі. Для кожної гри надавалася інформація про назву, рейтинг та інші характеристики, що дозволяло

користувачу швидко ознайомитися із запропонованими варіантами. Також було перевірено коректність взаємодії із зовнішнім API та швидкість отримання результатів. Отримані результати підтвердили правильність роботи алгоритму відбору та відображення спортивних відеоігор, а також стабільність роботи функціонального модуля в цілому.

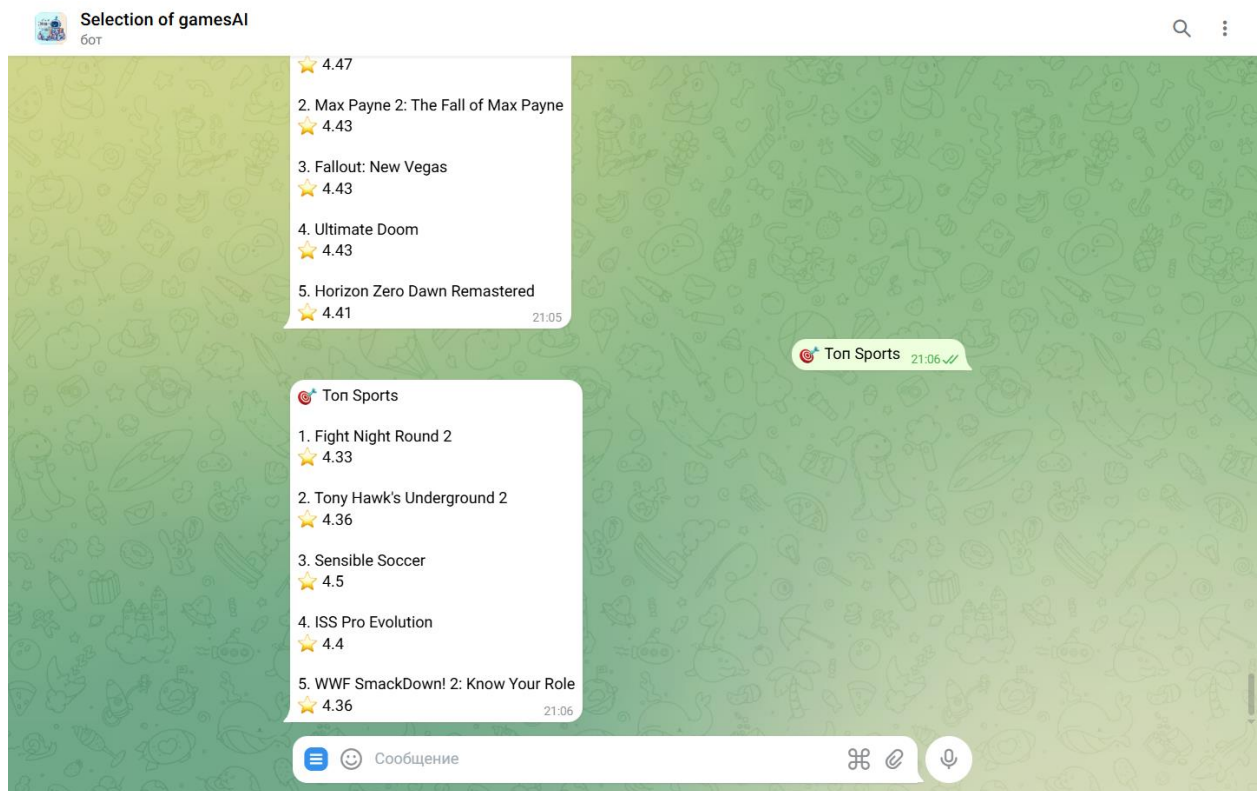


Рисунок 3.23 – Результат тестування добірки Top Sports

Додатково було проведено тестування функцій, пов'язаних із відображенням інформації про майбутні релізи відеоігор. Для цього використовувався календар релізів, який отримує актуальні дані через RAWG API та формує список найближчих ігрових новинок. Реалізація даної функції дозволяє користувачам своєчасно отримувати інформацію про майбутні виходи відеоігор та відстежувати найочікуваніші релізи (рис.3.24).

У ході тестування система успішно відображала інформацію про майбутні релізи, включаючи назви ігор та дати їх виходу. Для перевірки коректності роботи було виконано декілька послідовних запитів до функції календаря релізів,

після чого отримані результати порівнювалися з інформацією, доступною у сервісі RAWG API. У всіх випадках система повертала актуальні дані без помилок та затримок.

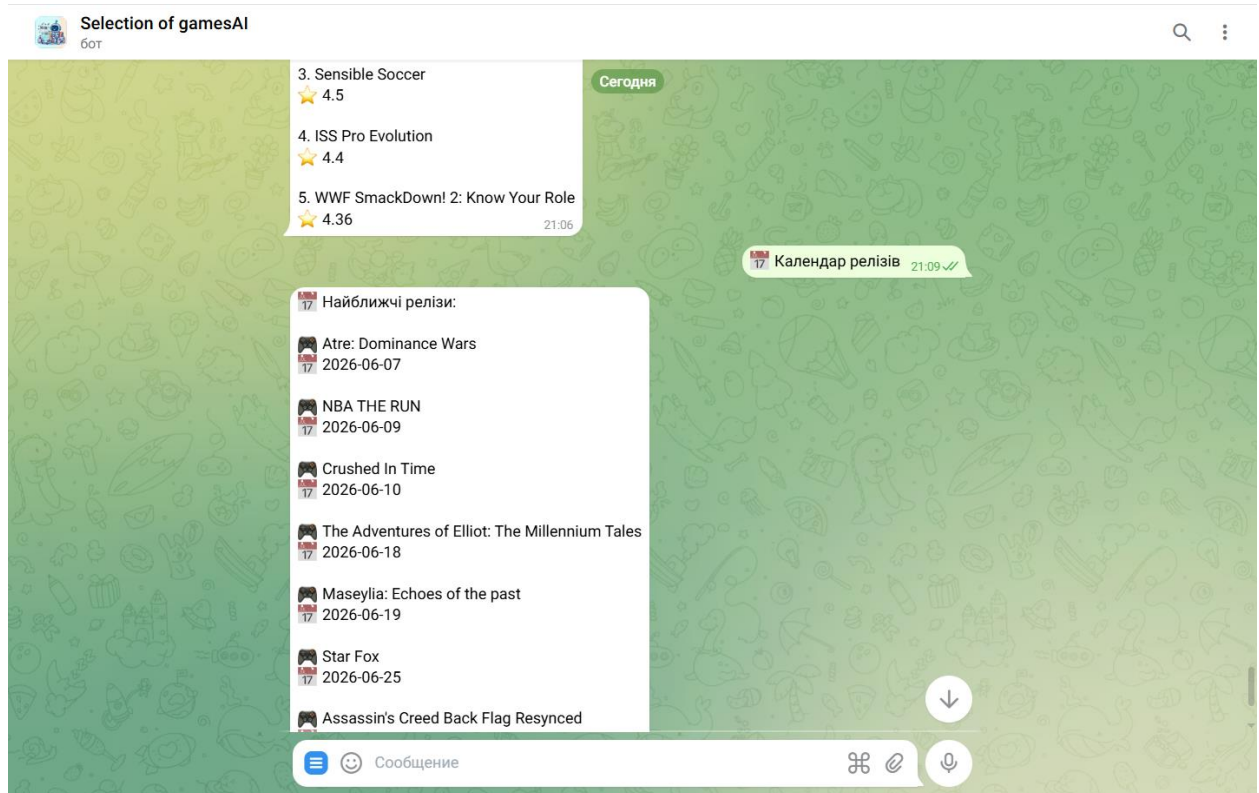


Рисунок 3.24 – Результат тестування календаря релізів

Особлива увага приділялася правильності обробки отриманої інформації та її відображенню в інтерфейсі Telegram. Було підтверджено, що бот коректно формує повідомлення, структурує перелік майбутніх релізів та забезпечує зручне представлення інформації для користувача. Крім того, було перевірено стабільність роботи функції при багаторазовому виконанні запитів, що підтвердило надійність реалізованого механізму взаємодії із зовнішнім API.

Використання календаря релізів суттєво розширює функціональні можливості Telegram-бота, оскільки дозволяє користувачам не лише отримувати інформацію про вже існуючі відеоігри, а й відстежувати появу нових проєктів на ринку. Це робить систему більш корисною для широкого кола користувачів та підвищує її практичну цінність.

Також було перевірено функцію пошуку дати виходу конкретної гри (рис.3.25). Після введення назви гри бот здійснював пошук відповідної інформації через RAWG API та повертав користувачу дату офіційного релізу. Усі тестові запити були успішно оброблені, а отримані результати відповідали інформації, доступній у зовнішній базі даних.

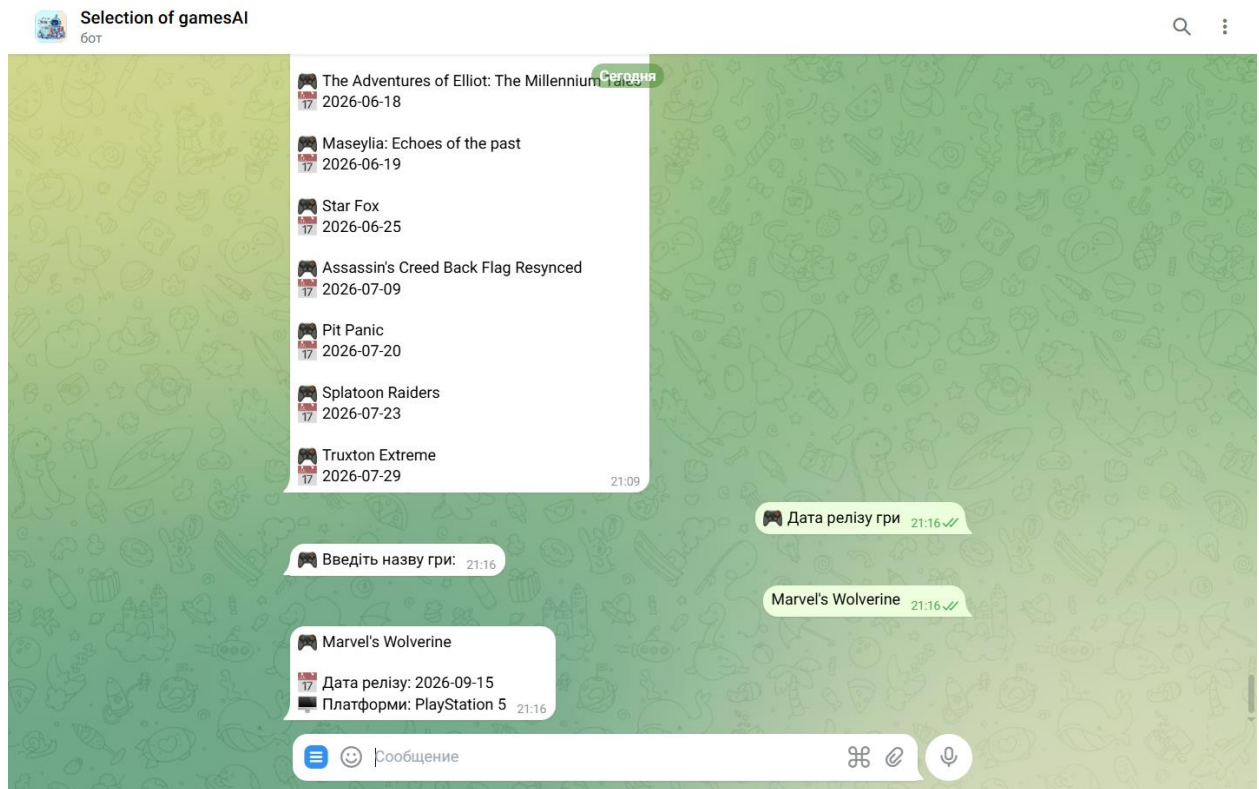


Рисунок 3.25 – Результат тестування функції пошуку дати виходу гри

Проведене тестування підтвердило коректну роботу всіх функціональних модулів розробленого Telegram-бота. Усі реалізовані функції працювали відповідно до поставлених вимог, успішно взаємодіяли із зовнішніми сервісами та забезпечували отримання актуальної інформації про відеоігри. Отримані результати свідчать про готовність програмного продукту до практичного використання та підтверджують ефективність запропонованих рішень.

3.8 Перспективи розвитку системи

Розроблений Telegram-бот для рекомендації відеоігор успішно реалізує основні функції пошуку інформації про ігри, формування рекомендацій за допомогою технологій штучного інтелекту, відображення актуальних знижок та збереження персональних даних користувачів. Проте сучасний розвиток інформаційних технологій та ігрової індустрії створює широкі можливості для подальшого вдосконалення розробленої системи.

Одним із перспективних напрямів розвитку є вдосконалення механізму формування рекомендацій. На даний момент система використовує можливості OpenAI API для аналізу текстових запитів користувачів та формування добірок відеоігор. У майбутньому доцільним є впровадження більш складних алгоритмів персоналізації, які враховуватимуть історію пошуку, оцінки користувача, список бажаних ігор та інші параметри його активності. Це дозволить підвищити точність рекомендацій та забезпечити більш індивідуальний підхід до кожного користувача.

Іншим напрямом розвитку може стати розширення кількості джерел даних. Наразі бот використовує сервіси RAWG API та CheapShark API для отримання інформації про відеоігри та актуальні знижки. У подальшому можливе підключення додаткових сервісів, які надають інформацію про новини ігрової індустрії, огляди, рейтинги користувачів та статистику популярності відеоігор. Це дозволить зробити інформаційне наповнення системи більш повним та різноманітним.

Перспективним також є вдосконалення підсистеми роботи зі знижками. У майбутньому можна реалізувати механізм автоматичного моніторингу цін на обрані користувачем ігри. У такому випадку система зможе автоматично повідомляти користувача про появу нових акційних пропозицій або зниження вартості гри, яка знаходиться у списку бажаного.

Окрему увагу можна приділити розвитку функцій статистичного аналізу. На основі накопичених даних про вподобання користувачів система зможе

автоматично визначати улюблені жанри, аналізувати зміни інтересів та формувати більш детальні рекомендації. Крім того, це дозволить створювати персоналізовані добірки ігор на основі попереднього досвіду користувача.

Ще одним напрямом розвитку є створення вебверсії або мобільного застосунку на основі реалізованого програмного продукту. Це дозволить розширити аудиторію користувачів та забезпечити доступ до функціоналу системи не лише через Telegram, а й через окремий інтерфейс. При цьому основна логіка роботи системи та інтеграція із зовнішніми сервісами можуть бути використані без суттєвих змін.

Таким чином, розроблений Telegram-бот має значний потенціал для подальшого розвитку та вдосконалення. Реалізація запропонованих напрямів модернізації дозволить підвищити функціональність системи, покращити якість рекомендацій та забезпечити більш комфортну взаємодію користувачів із програмним продуктом.

ВИСНОВКИ

У рамках кваліфікаційної роботи було розроблено та реалізовано Telegram-бот для підбору відеоігор із використанням технологій штучного інтелекту. Створена система дозволяє користувачам отримувати актуальну інформацію про відеоігри, переглядати майбутні релізи, знаходити акційні пропозиції та отримувати персоналізовані рекомендації на основі текстових запитів природною мовою.

На початковому етапі виконання роботи було проведено аналіз предметної області та досліджено особливості сучасного ринку відеоігор. Було розглянуто існуючі рекомендаційні системи, підходи до формування персоналізованих рекомендацій, особливості використання чат-ботів у сфері інформаційних сервісів, а також можливості застосування технологій штучного інтелекту для аналізу користувацьких запитів. Проведений аналіз підтвердив актуальність створення інтелектуальної системи рекомендації відеоігор у форматі Telegram-бота.

У другому розділі було виконано проєктування системи та визначено основні принципи її функціонування. Було обґрунтовано вибір архітектури програмного забезпечення, розглянуто методи рекомендаційних систем та підходи до обробки природної мови. Також було сформовано структуру взаємодії між користувачем, Telegram Bot API, зовнішніми сервісами та базою даних. У результаті було створено логічну модель системи, яка стала основою для подальшої реалізації програмного продукту.

У процесі виконання роботи було обрано мову програмування Python та бібліотеку aiogram для реалізації Telegram-бота. Для отримання інформації про відеоігри було використано сервіс RAWG API, який забезпечує доступ до великої бази даних ігрових проєктів. Для пошуку актуальних знижок було інтегровано CheapShark API, а для формування інтелектуальних рекомендацій використано OpenAI API. Зберігання даних користувачів було реалізовано за допомогою системи керування базами даних SQLite.

Під час виконання роботи було реалізовано:

- взаємодію користувачів із системою через Telegram Bot API;
- пошук інформації про відеоігри за назвою;
- відображення детального опису та скриншотів ігор;
- отримання інформації про дату виходу гри;
- перегляд календаря майбутніх релізів;
- формування тематичних добірок відеоігор за жанрами;
- отримання випадкових рекомендацій;
- пошук актуальних знижок на відеоігри;
- пошук акційних пропозицій для конкретної гри;
- реалізацію списку бажаних ігор (Wishlist);
- формування профілю користувача на основі історії взаємодії із системою;
- інтеграцію OpenAI API для формування рекомендацій відповідно до текстових побажань користувача;
- збереження даних користувачів у базі даних SQLite.

У третьому розділі було виконано практичну реалізацію програмного продукту та проведено тестування всіх функціональних модулів системи. Результати тестування підтвердили коректність роботи пошуку інформації про відеоігри, механізмів формування рекомендацій, підсистеми пошуку знижок, календаря релізів, списку бажаних ігор та інших реалізованих функцій. Також було підтверджено стабільність взаємодії із зовнішніми сервісами та правильність роботи механізмів збереження даних користувачів.

Отримані результати демонструють ефективність використання технологій штучного інтелекту для формування рекомендацій відеоігор на основі природномовних запитів. Використання OpenAI API дозволило реалізувати більш гнучкий та інтелектуальний підхід до підбору ігор порівняно з традиційними системами пошуку за жорстко визначеними параметрами.

Таким чином, поставлена мета кваліфікаційної роботи була досягнута, а всі поставлені завдання виконані у повному обсязі. Розроблений Telegram-бот є працездатним програмним продуктом, який може використовуватися для отримання інформації про відеоігри, формування рекомендацій та пошуку вигідних ігрових пропозицій. Подальший розвиток системи може бути пов'язаний із вдосконаленням алгоритмів персоналізації, розширенням джерел даних та впровадженням додаткових сервісів для користувачів.

Результати роботи апробовано у вигляді тез доповідей під час XXX Міжнародного молодіжного форуму «Радіoeлектроніка та молодь у XXI столітті» [31].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Шафроненко, А. Ю. (2023). *Адаптивні методи нечіткої кластеризації потоків даних з використанням еволюційного самонавчання* (Дис. д-ра техн. наук). Харківський національний університет радіоелектроніки, Харків.
2. Гороховатський, В. О., & Творошенко, І. С. (2022). *Аналіз багатовимірних даних за описом у формі множини компонент: монографія*. Харків: ХНУРЕ.
3. Mashtalir, S. V., & Nikolenko, O. V. (2024). Optimizing hierarchical classifiers with parameter tuning and confidence scoring. *Herald of Advanced Information Technology*, 7(3), 231–242.
4. Yakovleva, O., Nebeský, L., & Kirichenko, A. (2023). Using the GPT models for responses based on custom content to develop neural consultant for university applicants. In *Proceedings of the V International Scientific and Practical Conference “The World of Modern Technologies and Inventions”* (pp. 172–178). Madrid, Spain.
5. Pomazan, V., Tvoroshenko, I., & Gorokhovatskyi, V. (2023). Development of an application for recognizing emotions using convolutional neural networks. *International Journal of Academic Information Systems Research*, 7(7), 25–36.
6. Кобилін, О. А., & Творошенко, І. С. (2021). *Методи цифрової обробки зображень: навчальний посібник*. Харків: ХНУРЕ.
7. Ситніков, Д. Е., Ситнікова, П. Е., Тітов, С. В., & Тітова, О. В. (2021). Фільтрація результуючого набору асоціативних правил з точки зору оцінки цікавості. *Системи обробки інформації*, 1(164), 83–88.
8. Bodyanskiy, Y., Shafronenko, A., Rudenko, D., & Tanianskyi, O. (2024). Online stacking credibilistic fuzzy clustering for data stream mining. In *Proceedings of the Seventh International Workshop on Computer Modeling and Intelligent Systems* (pp. 1–10).

9. Руденко, Д. О., Лотвінова, В. В., & Безверха, Є. В. (2023). Навчання нейронної мережі в задачах обробки даних. In *Modernization of today's science: Experience and trends* (pp. 94–95). Singapore.
10. Yakovleva, O., Matúšová, S., Liubchenko, V., & Maksimov, H. (2025). Innovative solutions based on AI in education, on the example of generating multimodal lecture notes. *Verejná správa a regionálny rozvoj*, 140–161.
11. Lyubchenko, V., Veretelnyk, K., Kots, P., & Lyashenko, V. (2024). Digital image segmentation procedure as an example of an NP-problem. *The Multidisciplinary Journal of Science and Technology*, 4(4), 170–177.
12. Suprun, A., Tvoroshenko, I., Gorokhovatskyi, V., & Yakovleva, O. (2025). Development and research of a method for the combined use of large language models for text generation. *International Journal of Academic and Applied Research*, 9(10), 249–263.
13. Yakovleva, O., Matúšová, S., & Koshel, V. (2025). Implementation of AI approaches in current tools for managing image collections to improve the search capabilities. *Grail of Science*, 49, 752–755.
14. Bass, L., Weber, I., & Zhu, L. (2021). *DevOps: A software architect's perspective*. Addison-Wesley.
15. Ricci, F., Rokach, L., & Shapira, B. (2022). *Recommender systems handbook* (3rd ed.). Springer.
16. OpenAI. (2023). *GPT-4 technical report*. arXiv. <https://arxiv.org/abs/2303.08774>
17. Zhang, S., Yao, L., Sun, A., & Tay, Y. (2021). Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys*, 52(1), 1–38.
18. Brown, T., Mann, B., Ryder, N., et al. (2020). Language models are few-shot learners. In *Advances in Neural Information Processing Systems* (Vol. 33, pp. 1877–1901).
19. Bommasani, R., et al. (2021). *On the opportunities and risks of foundation models*. arXiv. <https://arxiv.org/abs/2108.07258>

20. OpenAI. (2023). *Introducing ChatGPT and Whisper APIs*.
<https://openai.com>
21. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT* (pp. 4171–4186).
22. OpenAI. (2024). *GPT-4o system card*. <https://openai.com>
23. Fielding, R. T. (2022). *Architectural styles and the design of network-based software architectures*. University of California.
24. Young, T., Hazarika, D., Poria, S., & Cambria, E. (2020). Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence Magazine*, 13(3), 55–75.
25. Burke, R., & Ramezani, M. (2021). Matching recommendation technologies and domains. In F. Ricci, L. Rokach, & B. Shapira (Eds.), *Recommender Systems Handbook* (pp. 367–386). Springer.
26. Vaswani, A., et al. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems* (Vol. 30).
27. Abdellatif, A., Mohamed, A., & Al-Mutairi, S. (2021). Chatbot applications in smart environments: Challenges and opportunities. *Journal of Information Technology Research*, 14(3), 1–18.
28. Schafer, J. B., Konstan, J., & Riedl, J. (2020). Recommender systems in e-commerce. In *Proceedings of the 1st ACM Conference on Electronic Commerce* (pp. 158–166).
29. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv. <https://arxiv.org/abs/1301.3781>
30. Vaswani, A., et al. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems* (Vol. 30).
31. Кириченко К. Ю. (2026) Створення системи рекомендацій відеоігор у форматі Telegram-бота з використанням ШІ. Радіоелектроніка і молодь у XXI столітті: тези доповідей 30-го Міжнародного молодіжного форуму (Харків, 22–24 квітня 2026 р.). Харків: ХНУРЕ, 2026. Т. 7. С. 74-76.