

Галицький фаховий коледж імені В'ячеслава Чорновола  
відділення комп'ютерних технологій  
циклова комісія інформатики та комп'ютерних дисциплін

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач відділення

комп'ютерних технологій

Наталія СТЕФУРАК / \_\_\_\_\_ /

підпис

«\_\_\_» \_\_\_\_\_ 2025 р.

## ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи

освітньо-професійного ступеня «фаховий молодший бакалавр»

зі спеціальності 122 «Комп'ютерні науки»

на тему: «Чат бот інтерактивної гри вікторина у групових чатах»

Студент групи КН-41      Станіслав БАЧИНСЬКИЙ      \_\_\_\_\_

(підпис)

Керівник роботи      Наталя КУЛЬЧИНСЬКА      \_\_\_\_\_

(підпис)

Консультанти:

з техніко-економічного

обґрунтування      Любов МЕЛЕНЧУК      \_\_\_\_\_

(підпис)

нормоконтролер      Надія ГАВРИШКІВ      \_\_\_\_\_

(підпис)

Тернопіль – 2025



Розділ	Консультанти	Підпис, дата	
		Завдання видано	Завдання прийнято
З техніко-економічного обґрунтування	Любов МЕЛЕНЧУК		

**КАЛЕНДАРНИЙ ПЛАН**  
виконання кваліфікаційної роботи

№ п/п	Найменування етапу	Терміни	
		початку	завершення
1	Вибір теми роботи, вивчення вимог до кваліфікаційної роботи	16.11.2024	25.11.2024
2	Дослідження наявних рішень та написання відповідного розділу	26.11.2024	06.12.2024
3	Дослідження технологій реалізації чат ботів	07.12.2024	10.12.2024
4	Проектування архітектури бота та визначення його основного функціоналу	21.12.2024	07.01.2025
5	Підготовка середовища розробки	08.01.2025	01.02.2025
6	Розробка архітектури чат бота та написання відповідного розділу	02.02.2025	25.04.2025
7	Реалізація основних функцій бота: запуск гри, приєднання, голосування, питання	26.04.2025	05.05.2025
8	Оптимізація коду, покращення UX-елементів і логіки взаємодії	14.04.2025	06.05.2025
9	Тестування чат бота та усунення помилок	06.05.2025	09.05.2025
10	Проведення техніко-економічного обґрунтування проєкту	10.05.2025	15.05.2025
11	Оформлення пояснювальної записки	16.05.2025	11.06.2025
12	Попередній захист кваліфікаційної роботи	13.06.2025	13.06.2025
13	Підготовка до захисту роботи	14.06.2025	23.06.2025
14	Захист кваліфікаційної роботи	24.06.2025	24.06.2025

Дата видачі “ \_\_\_ ” \_\_\_\_\_ 202\_ р. Керівник \_\_\_\_\_ / Кульчинська Н.З.

Завдання прийняв до виконання \_\_\_\_\_ / Бачинський С.П.

## Реферат

Чат бот інтерактивної гри вікторина у групових чатах.

Кваліфікаційна робота. Бачинський Станіслав. Галицький фаховий коледж імені В'ячеслава Чорновола, відділення комп'ютерних технологій. Спеціальність 122 «Комп'ютерні науки», 2025. Сторінок – 53, таблиць – 12, додатків – 2, джерел – 7.

Об'єкт кваліфікаційної роботи: процеси створення та використання чат-ботів для організації інтерактивних ігор у месенджерах.

Мета роботи: розробка Telegram-бота для проведення групових вікторин із підтримкою голосування за теми та складність, автоматизованим підрахунком балів, таймерами та збереженням результатів у базі даних.

Завдання роботи: реалізація функціоналу проведення вікторин, облік результатів, формування рейтингу гравців, гнучка архітектура з використанням принципів MVC.

Для реалізації проєкту використано Python, бібліотеку python-telegram-bot, MySQL, PyCharm. Інтерфейс бота реалізовано через інтерактивні елементи Telegram, а дані зберігаються у реляційній базі з чіткою структурою.

Результатом є завершений Telegram-бот, адаптований для українських користувачів, готовий до використання в освітніх і розважальних цілях у групових чатах.

Ключові слова: TELEGRAM-БОТ, ВІКТОРИНА, ГРУПОВИЙ ЧАТ, ІНТЕРАКТИВНИЙ ІНТЕРФЕЙС, PYTHON, MYSQL, ГЕЙМІФІКАЦІЯ, АВТОМАТИЗАЦІЯ.

## Abstract

Chatbot for a Quiz Game in Group Chats.

Qualification work. Bachynskyi Stanislav. Halytskyi Professional College named after Vyacheslav Chornovil, Department of Computer Technologies. Specialty 122 “Computer Science”, 2025. Pages – 53, tables – 12, appendices – 2, references – 7.

The research object is the processes of creating and using chatbots for organizing interactive games in messengers.

The goal is to develop a Telegram chatbot for conducting group quizzes with support for voting on topics and difficulty levels, automated score calculation, timers, and result storage in a database.

The tasks of the work include implementing the functionality for conducting quizzes, tracking results, generating player rankings, and creating a flexible architecture based on MVC principles.

The project was implemented using Python, the python-telegram-bot library, MySQL, and PyCharm. The chatbot’s interface is built with Telegram’s interactive elements, and data is stored in a clearly structured relational database.

The result is a completed Telegram chatbot tailored for Ukrainian users, ready for use in educational and entertainment purposes in group chats.

Keywords: TELEGRAM CHATBOT, QUIZ, GROUP CHAT, INTERACTIVE INTERFACE, PYTHON, MYSQL, GAMIFICATION, AUTOMATION.

## ЗМІСТ

Вступ.....	7
1 Аналіз предметної області та постановка завдань.....	9
1.1 Опис предметної області .....	9
1.2 Аналіз наявних рішень .....	11
1.3. Аналіз вимог до програмного засобу та постановка завдання.....	14
2 Проектування системи.....	16
2.1 Проектування структури проєкту.....	16
2.2 Проектування логіки гри .....	18
2.3 Проектування архітектури .....	20
2.5 Опис функціональних модулів .....	24
2.6 Проектування інтерфейсу .....	26
2.7 Вибір засобів розробки.....	28
3 Реалізація та тестування системи .....	30
3.1 Необхідне програмне забезпечення .....	30
3.2 Встановлення та налаштування модулів .....	31
3.3 Створення бази даних.....	32
3.4 Процес розробки програмного продукту .....	33
3.5 Реалізація інтерфейсу користувача .....	36
3.6 Тестування .....	40
4 Техніко-економічне обґрунтування .....	45
4.1 Аналіз ринку .....	45
4.2 Розрахункова частина .....	46
4.3 Обґрунтування необхідності розробки .....	50
Висновки .....	51
Перелік джерел посилання .....	53
Додатки.....	54

					<b>КР.КН 25.582.01.000 ПЗ</b>			
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Чат бот інтерактивної гри вікторина у групових чатах	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
Розроб.		Бачинський С.					5	57
Перевір.		Квльчинська Н.				<b>ГФК. ВКТ. КН-41</b>		
Рецензет.		Сиротюк О.						
Норм.контр.		Гавришків Н.						
Зав. відд.		Стефурак Н.						

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

PY – Python

MVC – Model-View-Controller

API – Application Programming Interface

DB – Database

UID – User ID

VPS – Virtual Private Server

HTML – HyperText Markup Language

ER – Entity-Relationship

HTML – HyperText Markup Language

					КР.КН 25.582.01.000 ПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

## ВСТУП

Цифрові технології сьогодні глибоко інтегровані в повсякденне життя, докорінно змінюючи підходи до комунікації, освіти та дозвілля. Месенджери, зокрема Telegram, еволюціонували далеко за межі простого обміну повідомленнями, перетворившись на багатофункціональні платформи для інтерактивних сервісів різного призначення. Серед розмаїття інтерактивного контенту особливу популярність здобули вікторини, які ефективно стимулюють розвиток логіки, збагачують знання та створюють атмосферу розваги у групових чатах.

Актуальність теми дослідження зумовлена кількома факторами. По-перше, зростаючим інтересом до інтерактивних розваг у месенджерах, які дозволяють користувачам взаємодіяти в реальному часі. По-друге, потребою в автоматизованих рішеннях, що забезпечують зручне проведення вікторин без необхідності залучення ведучого або модератора. По-третє, важливістю розробки гнучкого та функціонального інструменту, який легко інтегрується у групові чати та може використовуватися як для розваг, так і для освітніх цілей.

Метою даної роботи є розробка чат-бота для організації вікторин у групових чатах, який дозволить учасникам проходити ігрові тести, отримувати бали за правильні відповіді та змагатися один з одним у режимі реального часу. Для досягнення цієї мети необхідно вирішити наступні завдання:

- Аналіз існуючих рішень та визначення їх сильних і слабких сторін.
- Розробка архітектури чат-бота та його функціональних можливостей.
- Реалізація механізму створення, збереження та проведення вікторин.
- Впровадження системи підрахунку балів та рейтингу гравців.
- Забезпечення зручного та інтуїтивного інтерфейсу для користувачів.

					КР.КН 25.582.01.000 ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

– Тестування та оптимізація бота для стабільної роботи у групових чатах.

Об'єктом дослідження є процес створення інтерактивних вікторин у месенджерах. Предметом дослідження є методи та інструменти розробки Telegram-ботів для інтерактивних ігор у групових чатах.

Результати цієї роботи можуть бути застосовані в наступних сферах:

– Організація інтерактивних розваг та освітніх заходів у групових чатах.

– Використання ботів для підвищення залученості користувачів у спільнотах.

– Розробка подібних ігрових рішень для месенджерів.

– Дослідження користувацького досвіду у сфері інтерактивних чат-ботів.

Практичне значення роботи полягає у створенні чат-бота, який дозволяє легко та ефективно організовувати вікторини у групових чатах, що сприятиме активному залученню учасників, розвитку командної взаємодії та підвищенню інтересу до освітніх та розважальних заходів.

					КР.КН 25.582.01.000 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАНЬ

## 1.1 Опис предметної області

У сучасному цифровому світі месенджери стали ключовим інструментом комунікації, що об'єднує людей незалежно від їхнього географічного розташування. З появою таких платформ, як Telegram, WhatsApp, Viber та інших, користувачі отримали можливість не лише обмінюватися текстовими повідомленнями, але й передавати мультимедійний контент (фото, відео, аудіо), здійснювати голосові та відеодзвінки, а також брати участь у групових чатах для спільної роботи чи розваг. Завдяки зручності, швидкості та доступності месенджери міцно увійшли в повсякденне життя мільйонів людей по всьому світу, ставши невід'ємною частиною як особистого, так і професійного спілкування.

З розвитком технологій і масовим поширенням мобільних пристроїв месенджери почали виконувати значно ширший спектр функцій, ніж просто засоби обміну повідомленнями. Одним із перспективних напрямків їхнього використання є автоматизація процесів за допомогою чат-ботів – програмних інструментів, які здатні взаємодіяти з користувачами в реальному часі, виконувати команди та надавати персоналізовані послуги. Чат-боти вже успішно застосовуються в різних сферах: від автоматизації клієнтської підтримки в бізнесі до організації інтерактивних розваг і навчання. У цьому контексті створення Telegram-бота для проведення вікторин у групових чатах є актуальним і перспективним завданням, що поєднує в собі як розважальні, так і освітні аспекти.

Групові чати в месенджерах, зокрема в Telegram, активно використовуються для координації спільної діяльності, обміну інформацією та неформального спілкування. Вони популярні серед студентів, які обговорюють навчальні проекти, серед колег, які планують робочі завдання, а також серед друзів, які шукають способи спільного дозвілля.

					КР.КН 25.582.01.000 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

Проведення вікторин у таких чатах може слугувати різноманітним цілям: від простої розваги до підвищення залученості учасників у навчальний процес чи командну роботу. Наприклад, у навчальних закладах вікторини допомагають перевіряти знання студентів у цікавій ігровій формі, у корпоративному середовищі – підвищувати мотивацію працівників через гейміфікацію, а в особистих групах – створювати дружню атмосферу та стимулювати взаємодію.

Основна ідея створення чат-бота полягає в автоматизації всіх етапів проведення вікторин: включаючи створення питань, до обробки відповідей користувачів, підрахунку балів і формування рейтингу учасників. Такий бот може включати додаткові функції, наприклад, голосування за теми та складність вікторин та таймери для обмеження часу на відповідь. На відміну від традиційних методів організації ігор, де потрібен ведучий або значна підготовка, автоматизований бот значно спрощує процес, роблячи його доступним навіть для великих груп користувачів.

Сьогодні подібні рішення вже існують на ринку, однак більшість із них мають суттєві обмеження. Наприклад, багато ботів не підтримують локалізацію для різних мов або погано адаптовані до роботи в групових чатах із великою кількістю учасників. Крім того, часто відсутня можливість інтеграції з іншими функціями месенджера, такими як голосування чи сповіщення, що зменшує їхню ефективність. Усе це підкреслює потребу в розробці нового Telegram-бота, який би усунув ці недоліки та запропонував користувачам ширший функціонал.

Ще одним важливим аспектом є підвищення залученості користувачів. Інтерактивні елементи, такі як зворотний відлік часу, динамічні рейтингові таблиці, можливість, можуть перетворити звичайну вікторину на захоплюючий процес. Окрім розважального потенціалу, подібні боти мають значний прикладний характер.

					КР.КН 25.582.01.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

У навчальних закладах їх можна використовувати для оцінки знань студентів або створення інтерактивних уроків, у корпоративних тренінгах – для тестування навичок працівників, у мовних школах – для практики лексики чи граматики, а в маркетингових кампаніях – для залучення аудиторії через гейміфіковані опитування. Таким чином, Telegram-бот для вікторин може стати універсальним інструментом із широким спектром застосувань.

## 1.2 Аналіз наявних рішень

На ринку вже представлено кілька Telegram-ботів для проведення вікторин, які мають різний рівень функціональності та популярності серед користувачів. Щоб оцінити їхні можливості та виявити прогалини, проведемо детальний аналіз найпоширеніших аналогів.

QuizBot, як офіційний продукт Telegram, є одним із найпростіших у використанні рішень. Він дозволяє швидко створювати вікторини з кількома варіантами відповідей, але його функціонал обмежений базовими можливостями. Наприклад, користувачі не можуть налаштовувати рівень складності чи додавати власні питання, а також відсутня підтримка групових рейтингів, що робить його менш придатним для змагань у великих чатах (рис. 1.1).

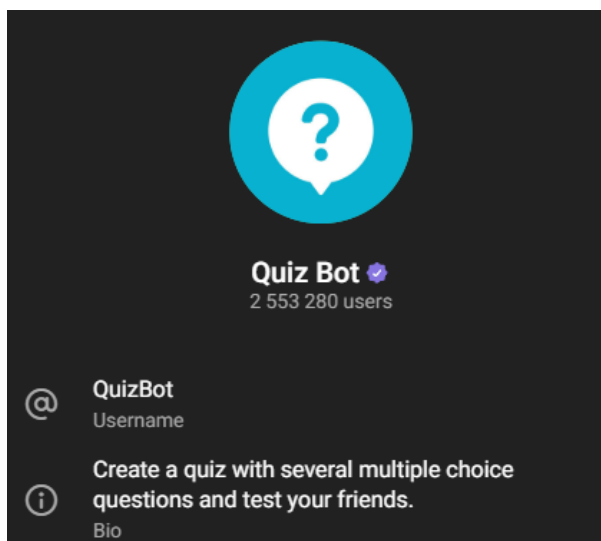


Рисунок 1.1 – Бот Quiz

					КР.КН 25.582.01.000 ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

TriviaBot пропонує ширший вибір категорій питань (наприклад, історія, спорт, наука), що робить його привабливим для любителів різноманітних тем. Він також підтримує гру в групах, але не дозволяє адміністраторам додавати власний контент чи налаштовувати параметри вікторини, такі як таймер або складність. Це суттєво обмежує його гнучкість і потенціал для використання в навчальних цілях (рис. 1.2).

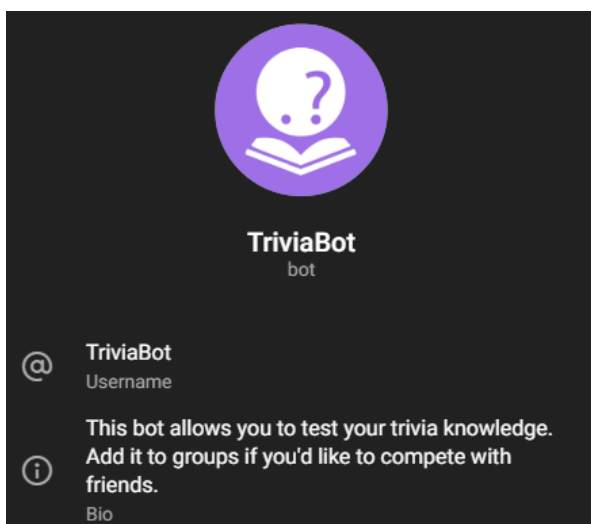


Рисунок 1.2 – Бот TriviaBot

Quizarium орієнтований на командні змагання, що робить його цікавим для групового формату. Проте відсутність розширених аналітичних інструментів, таких як статистика результатів чи історія ігор, а також обмежена адаптація до роботи з різними рівнями складності зменшують його привабливість для більш серйозних застосувань (рис. 1.3).



Рисунок 1.3 – Бот Quizarium

					КР.КН 25.582.01.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

Окрім Telegram-ботів, існують також інші платформи для створення інтерактивних опитувань і тестів, такі як Google Forms, Kahoot і Socrative, які часто використовуються в освітній сфері.

Результати цього аналізу представлені в таблиці 1.1, яка включає опис ботів, їхні переваги та недоліки.

Таблиця 1.1 – Таблиця аналогів

Назва	Опис	Переваги	Недоліки
QuizBot	Офіційний бот Telegram для створення вікторин	Простота використання, інтеграція з Telegram	Обмежена кастомізація, відсутність підтримки групових рейтингів, відсутність таймера, відсутність вибору складностей.
TriviaBot	Багатофункціональний бот для вікторин	Підтримка різних категорій питань, можливість гри в групах	Відсутність можливості додавання власних запитань, відсутність таймера, відсутність вибору тем і рівнів складності.
Quizarium	Інтерактивний бот для командних вікторин	Можливість гри в групах.	Відсутність розширених аналітичних функцій, відсутність вибору складностей.

Google Forms дозволяє створювати детальні тести з можливістю збереження відповідей, але не інтегрується з месенджерами, що ускладнює його використання в групових чатах. Kahoot відомий своєю гейміфікацією та інтерактивністю, однак потребує окремого доступу через браузер або додаток, що робить його менш зручним для швидкого запуску в Telegram. Socrative підходить для оцінки знань у реальному часі, але його інтеграція з месенджерами також обмежена.

					КР.КН 25.582.01.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

Аналіз наявних рішень показує, що більшість ботів і платформ мають спільні недоліки: недостатня гнучкість у налаштуванні контенту, обмежена підтримка групових функцій, відсутність розширеної аналітики чи локалізації. Ці прогалини створюють потребу в розробці нового Telegram-бота, який би врахував згадані недоліки та запропонував користувачам ширший набір можливостей, адаптований до потреб як розважального, так і навчального формату.

### 1.3. Аналіз вимог до програмного засобу та постановка завдання

У процесі аналізу предметної області та вивчення аналогічних рішень сформувався чітке бачення того, яким повинен бути Telegram-бот для вікторин. Основна мета — зробити його простим, зручним і при цьому максимально функціональним як для групових чатів, так і для приватного використання.

До функціональних вимог можна віднести насамперед автоматизацію всього процесу вікторини. Бот повинен сам ставити запитання, приймати відповіді від гравців, рахувати бали та автоматично будувати рейтинг. Учасники повинні мати можливість обирати категорії вікторин та складність запитань, щоб гра була цікавою для всіх, незалежно від рівня підготовки. Крім того, бот має без проблем працювати в групових чатах, навіть якщо там багато людей — важливо, щоб усі відповіді оброблялись чітко й швидко.

Інтерфейс повинен бути інтуїтивно зрозумілий — все має працювати без зайвих команд, із кнопками, таймерами, сповіщеннями про початок чи завершення гри. Також обов'язковою є реалізація рейтингової системи, яка буде вести облік результатів і формувати таблицю лідерів. При цьому важливо, щоб усі персональні дані гравців зберігалися безпечно, без ризику стороннього доступу.

Що стосується нефункціональних вимог, то на першому місці — стабільність і надійність.

					КР.КН 25.582.01.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

Бот не повинен зависати чи видавати помилки, незалежно від того, скільки людей бере участь у грі. Важлива й швидкодія: усі відповіді й команди мають оброблятися практично миттєво. Також враховується масштабованість — тобто, бот має справлятися з великою кількістю гравців одночасно без втрати продуктивності.

Окрема увага приділяється кросплатформеності: бот повинен однаково добре працювати на різних пристроях і версіях Telegram, щоб жоден користувач не відчував дискомфорту. І ще один ключовий момент — можливість легкого розширення функціоналу. Архітектура має бути такою, щоб нові фішки можна було додавати без глобальних змін у коді.

Таким чином, розроблений Telegram-бот дозволить користувачам ефективно проводити вікторини у групових чатах, підвищуючи рівень взаємодії та залученості учасників. Його гнучкість та розширені функціональні можливості зроблять його зручним інструментом для проведення як розважальних, так і навчальних вікторин.

Було проведено ґрунтовний аналіз предметної області, розглянуто роль месенджерів і чат-ботів у сучасному світі, а також оцінено переваги й недоліки існуючих рішень для проведення вікторин. На основі цього сформульовано потребу в розробці нового Telegram-бота з розширеним функціоналом, включаючи підтримку групових змагань, голосування за вибір теми та складності, таймер для відповідей та результати ігор. Визначено ключові вимоги та поставлено конкретні завдання, які будуть реалізовані в наступних етапах роботи. Успішна реалізація запропонованого рішення дозволить створити ефективний і багатофункціональний інструмент для інтерактивних вікторин у Telegram.

					КР.КН 25.582.01.000 ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

## 2 ПРОЄКТУВАННЯ СИСТЕМИ

### 2.1 Проєктування структури проєкту

Структура Telegram-бота для вікторини проєктуватиметься з урахуванням зручності підтримки, логіки організації коду та майбутнього розширення функціоналу. Основна ідея полягає в тому, щоб побудувати систему, яка буде легкою у розвитку, гнучкою в налаштуванні та стабільною під час роботи навіть із великою кількістю користувачів.

У головному файлі буде розміщено логіку ініціалізації бота, підключення команд, обробників подій, а також запуск основних функцій гри. Тут також зберігатиметься токен, необхідний для авторизації в Telegram.

Для роботи з базою даних буде створений окремий модуль, який відповідатиме за збереження і обробку інформації. Він буде взаємодіяти з базою даних: додавати нових користувачів, зберігати питання, результати вікторин, статистику тощо. Такий підхід дозволить ефективно управляти всією інформацією та забезпечить стабільність у роботі.

Основна геймплейна логіка буде рознесена по окремих функціях, які працюватимуть асинхронно, щоб уникати затримок і не блокувати роботу бота. Наприклад, функція, яка відповідає за відлік часу до старту гри, дасть змогу користувачам встигнути приєднатися. Інші функції будуть відповідати за вибір теми вікторини та рівня складності через голосування, запуск самої гри, надсилання питань учасникам і обробку їхніх відповідей. Після кожного раунду система автоматично перевірятиме відповіді, оновлюватиме рахунки та фіксуватиме ці результати в базі.

Окремий блок буде обробляти відповіді — перевіряти правильність та оновлювати рахунки гравців. Також планується реалізація модуля взаємодії з кнопками, щоб учасники могли легко приєднуватися до гри, голосувати за теми чи складність, і вибирати відповіді без зайвих команд.

					КР.КН 25.582.01.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

Проект буде побудований за модульним принципом, де кожен блок коду чітко відповідатиме за свою частину, але водночас зможе ефективно взаємодіяти з іншими. Такий підхід дасть можливість легко оновлювати окремі частини бота, додавати нові функції або змінювати існуючі без ризику порушити роботу всієї системи. Загальна структура проекту буде побудована так, щоб усі модулі були незалежними, але добре взаємодіяли між собою. Це забезпечить простоту в подальшій розробці, можливість масштабування та гнучкість у додаванні нового функціоналу. На рисунку 2.1 зображено структурну схему, яка ілюструє основні частини системи та їх взаємодію.

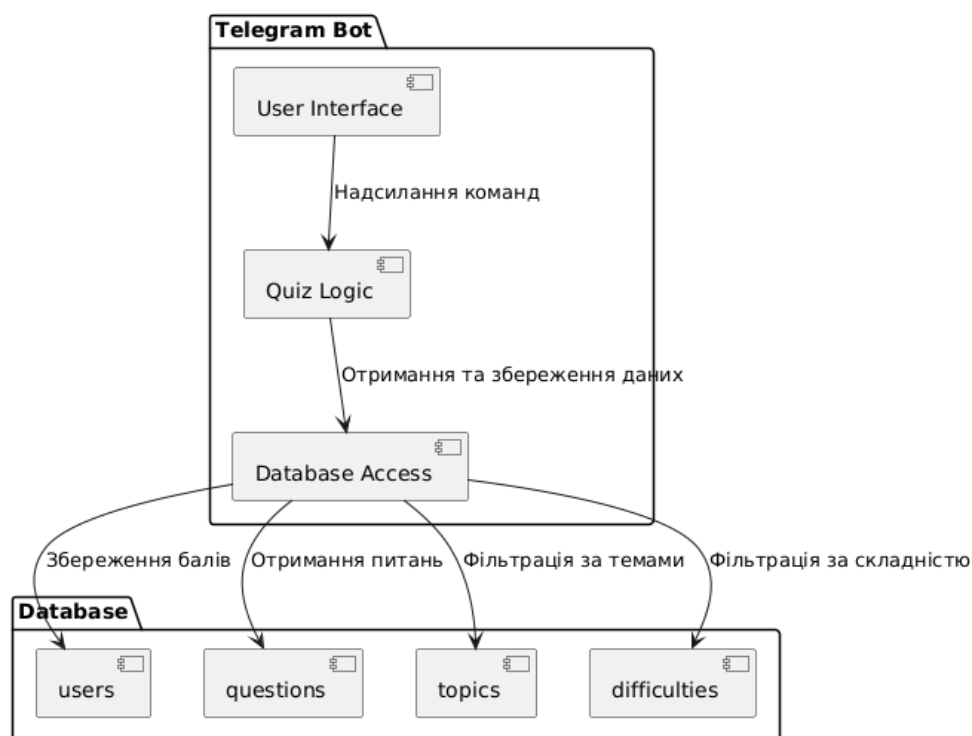


Рисунок 2.1 – Структурна схема

Схема демонструє логічну побудову проекту та відображає взаємозв'язки між його основними компонентами.

Така структура є оптимальною для реалізації функціоналу вікторини, забезпечуючи зручність у підтримці, тестуванні та подальшому розширенні системи.

## 2.2 Проектування логіки гри

Логіка роботи Telegram-бота для вікторини побудована на чіткому алгоритмі, який забезпечує послідовність дій користувачів, інтерактивність та зручність гри. Нижче детально описується кожен етап ігрового процесу.

### 1) Ініціалізація гри

Гра починається з активації спеціальної команди, після чого бот перевіряє наявність активної сесії у чаті. Якщо гру можна розпочати, запускається 60-секундний період для приєднання учасників. Після завершення цього часу система або переходить до наступного етапу (за наявності гравців), або автоматично скасовує гру.

### 2) Вибір параметрів гри

На цьому етапі відбувається послідовне голосування за тему та рівень складності. Кожне голосування триває обмежений час (20 та 15 секунд відповідно) і завершується вибором варіанту з найбільшою кількістю голосів. У разі нічії переможний варіант вибирається випадковим чином.

### 3) Ігровий процес

Основний етап складається з циклічного виконання таких дій:

- Відправка питання з варіантами відповідей.
- Очікування відповідей протягом 10 секунд.
- Перевірка правильності відповідей.
- Нарахування балів (2 бали за першу правильну відповідь, 1 бал за наступні, 0 балів за неправильну та -1 бал, якщо відповідь відсутня).
- Відображення правильної відповіді

Після кожних трьох питань відображаються проміжні результати.

### 4) Завершення гри

Після останнього питання система формує фінальну статистику, яка включає:

- Підсумковий рейтинг гравців.
- Кількість набраних балів.

					КР.КН 25.582.01.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

– Відсоток правильних відповідей.

Усі результати зберігаються в базі даних для подальшого аналізу, після чого гра повністю завершується.

### 5) Спеціальні ситуації

Система передбачає обробку різних нестандартних сценаріїв:

- Автоматичне скасування гри при відсутності учасників.
- Технічні повідомлення у разі збоїв.

На рисунку 2.2 зображено діаграму діяльності, яка демонструє послідовність основних дій Telegram-бота під час гри, а також умовні переходи між ними залежно від подій, ініційованих користувачами.



Рисунок 2.2 – Діаграма діяльності

Така структура гри забезпечує зручну взаємодію, прозорість ігрового процесу та стабільну роботу навіть при великій кількості учасників. Кожен етап логічно пов'язаний з попереднім, що створює цілісний ігровий досвід.

### 2.3 Проектування архітектури

У процесі проектування інтерактивного квіз-бота для групових чатів було прийнято рішення використовувати клієнт-серверну архітектуру із застосуванням принципів MVC. Це забезпечує гнучкість у розробці, масштабованість системи та розділення логіки обробки даних, їхнього збереження та взаємодії з користувачем.

Бот буде реалізований за допомогою бібліотек, які дозволять асинхронно обробляти запити користувачів і ефективно керувати станами діалогу. Для збереження інформації буде використовуватись реляційна база даних, яка повинна гарантувати швидкий доступ до даних і забезпечує цілісність транзакцій.

Основні компоненти системи:

- Контролер (Bot Handler) – отримує запити від користувачів, обробляє їх за допомогою сценаріїв діалогу та передає відповідні команди у бізнес-логіку.
- Модель (Database Layer) – відповідає за збереження даних у базі, включаючи інформацію про користувачів, питання, відповіді та результати тестів.
- Відображення (Message Formatting Layer) – форматує відповіді бота у вигляді повідомлень, клавіатур та кнопок Telegram.

Обмін даними між компонентами системи та сховищем передбачається через спеціальний програмний інтерфейс доступу до даних, який абстрагує взаємодію з базою та забезпечує надійність, зручність і захист від помилок. Такий підхід дозволить централізовано керувати збереженням і обробкою інформації без прив'язки до конкретної реалізації на цьому етапі.

					КР.КН 25.582.01.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

## 2.4 Проектування бази даних

Основною таблицею є questions, яка містить записи про питання. Кожен запис включає ідентифікатор питання, текст, варіанти відповідей, правильну відповідь, а також зовнішні ключі до таблиць topics і difficulties. Структура таблиці представлена в таблиці 3.1.

Таблиця 3.1 – Таблиця атрибутів сутності questions

Атрибут	Умовне позначення	Ключ	Формат	Невизначене значення	Обмеження
Ідентифікатор питання	question_id	PRIMARY KEY	int	AUTO_INCREMENT	NOT NULL
Ідентифікатор теми	topic_id	FOREIGN KEY	int	NULL	REFERENCES topics(topic_id)
Текст питання	question_text	-	text	NULL	
Варіанти відповідей	answer_options	-	text	NULL	
Правильна відповідь	correct_answer	-	varchar(250)	NULL	
Ідентифікатор складності	difficulty_id	FOREIGN KEY	int	NULL	REFERENCES difficulties(difficulty_id)

Було розроблено таблицю атрибутів сутності users, яка зберігає дані про користувачів (таблиця 3.2).

Таблиця 3.2 – Таблиця атрибутів сутності users

Атрибут	Умовне позначення	Ключ	Формат	Невизначене значення	Обмеження
Ідентифікатор питання	user_id	PRIMARY KEY	int	-	NOT NULL
Бали	points	-	int	NULL	-
Ім'я користувача	username	-	varchar(100)	NULL	-

Було розроблено таблицю атрибутів сутності difficulties, яка зберігає дані про складності гри (таблиця 3.3).

Таблиця 3.3 – Таблиця атрибутів сутності difficulties

Атрибут	Умовне позначення	Ключ	Формат	Невизначене значення	Обмеження
Ідентифікатор складності	difficulty_id	PRIMARY KEY	int	AUTO_INCREMENT	NOT NULL
Назва складності	difficulty_name	-	text	NULL	NOT NULL
Кількість голосів	votes	-	int	NULL	NOT NULL

Було розроблено таблицю атрибутів сутності topics, яка зберігає дані про теми гри (таблиця 3.4).

Таблиця 3.4 – Таблиця атрибутів сутності topics

Атрибут	Умовне позначення	Ключ	Формат	Невизначене значення	Обмеження
Ідентифікатор теми	topic_id	PRIMARY KEY	int	AUTO_INCREMENT	NOT NULL
Назва теми	topic_name	-	varchar(100)	NULL	-
Кількість голосів	votes	-	int	NULL	NOT NULL

На рисунку 2.3 наведена діаграма бази даних (ER), що відображає структуру таблиць та їх зв'язки між собою.

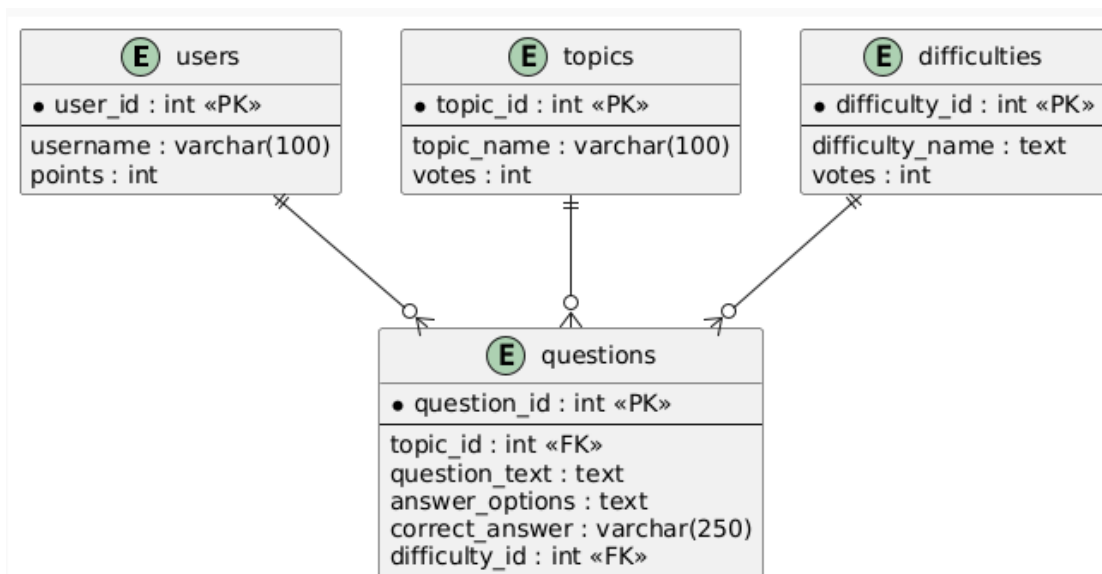


Рисунок 2.3 – Діаграма ER

На рисунку 2.4 зображена діаграма послідовності, яка демонструє процес обміну повідомленнями між користувачем, ботом та базою даних під час проходження квізу.

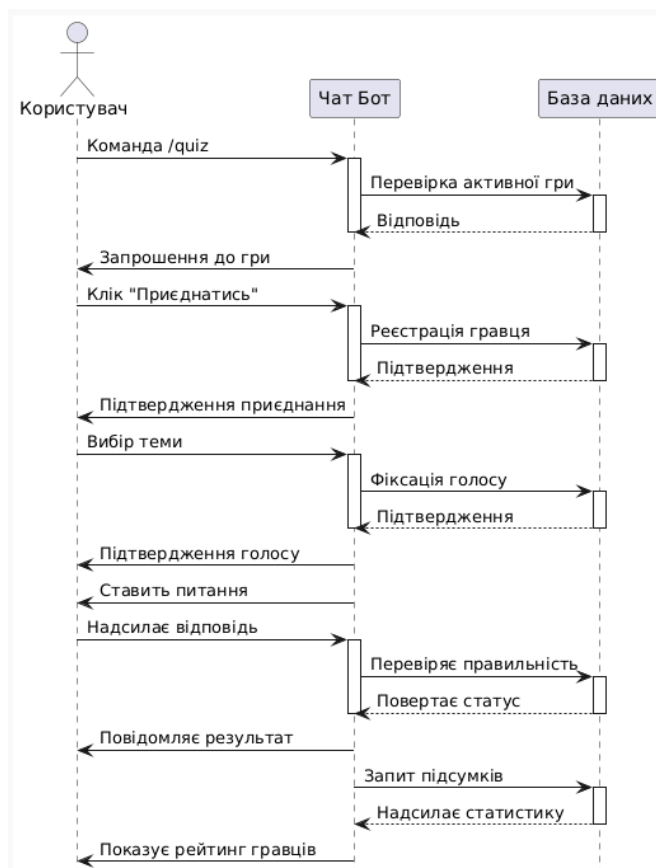


Рисунок 2.4 – Діаграма послідовності

Для автентифікації користувачів використовується їхній Telegram ID, що дозволяє уникнути необхідності реєстрації та паролів. Кожен користувач має унікальний запис у базі, що гарантує індивідуальність статистики та результатів.

На етапі проектування передбачається, що застосування клієнт-серверної архітектури з асинхронною обробкою запитів дасть змогу розробити ефективного та зручного Чат-бота, який забезпечуватиме інтерактивну взаємодію та зберігання історії проходження вікторин.

## 2.5 Опис функціональних модулів

### 1) Модуль автентифікації користувачів

Модуль автентифікації користувачів є ключовим компонентом Telegram-бота, що відповідає за ідентифікацію гравців і збереження їхніх персоналізованих результатів. Завдяки використанню Telegram API, процес автентифікації реалізоватиметься автоматично на основі Telegram ID, що дозволяє унікально ідентифікувати кожного користувача без потреби в класичній реєстрації чи введенні пароля.

Під час першого приєднання до гри (кнопка "Приєднатись") бот перевіряє наявність користувача в таблиці users бази даних. Якщо запис відсутній, створюється новий профіль з полями user\_id, username та points (початкове значення балів — 0).

Для збереження поточного стану гри (активні гравці, обрана тема, рівень складності тощо) використовується структура chat\_data з бібліотеки python-telegram-bot. Це дозволяє уникнути надмірних запитів до бази даних і забезпечує безперервність ігрового процесу.

### 2) Модуль управління ігровим процесом

Модуль вікторини є основною логічною частиною бота, що відповідає від початкового етапу приєднання до вибору теми, рівня складності та відправки запитань.

					КР.КН 25.582.01.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		24

### 3) Модуль статистики та аналізу результатів

Після завершення гри бот буде аналізувати результати кожної сесії та надає гравцям детальну статистику у зручному форматі.

Загальна кількість балів відображає сумарний прогрес гравця: кожен учасник бачить, скільки балів він заробив за поточну гру. Ця інформація береться з тимчасового сховища `chat_data`, де під час гри накопичуються бали за правильні відповіді. Наприкінці сесії бот також оновлює загальний рахунок у таблиці `users` у `BD`, що дозволяє відстежувати довгострокову статистику.

Відсоток правильних відповідей розраховуватиметься як співвідношення кількості вдалих спроб до загальної кількості питань. Наприклад, якщо гравець правильно відповів на 7 з 10 запитань, його результат становитиме 70%. Цей показник допомагає оцінити не лише кількість набраних балів, а й якість гри — чи були відповіді швидкими та точними, чи гравець часто помилявся. Для розрахунку використовуються дані з `chat_data`, де фіксуються правильні (`correct_answers`) та загальні (`total_answers`) відповіді кожного учасника.

Рейтинг учасників формуватиметься на основі сортування гравців за кількістю балів — від найвищого до найнижчого результату. Це створює елемент змагання: учасники бачать своє місце серед інших, що мотивує їх покращувати результати в наступних іграх. Рейтинг відображається у фінальному повідомленні у вигляді нумерованого списку (наприклад, "1. User1: 15 балів, 2. User2: 12 балів").

Таким чином, статистика не лише буде фіксувати результати, але й робити гру більш інтерактивною та цікавою для користувачів.

### 4) Модуль керування контентом (питаннями)

Усі запитання зберігатимуться в таблиці `questions` з полями:

- `question_id` – унікальний ідентифікатор.
- `topic_id` – зв'язок із таблицею `topics`.
- `question_text` – текст питання.

					КР.КН 25.582.01.000 ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

– answer\_options – варіанти відповідей (у форматі "А) Варіант1, Б) Варіант2").

– correct\_answer – правильна відповідь.

– difficulty\_id – рівень складності (1-3).

Для вибору запитань використовується SQL-запит з випадковим сортуванням (ORDER BY RAND()), що забезпечує різноманітність ігрових сесій.

Взаємодія з користувачем (інтерфейсні елементи, таймери, повідомлення) детально розглядається у підпункті 2.4.

## 2.6 Проєктування інтерфейсу

Інтерфейс бота буде побудовано на принципах мінімалізму та функціональності, що забезпечує зручну взаємодію через текстові команди та інтерактивні елементи. Основна увага приділена простоті використання та швидкості реакції системи на дії користувача.

Інтерфейс реалізовано у вигляді стандартного чат-інтерфейсу месенджера Telegram. Взаємодія з ботом здійснюється через повідомлення та вбудовані кнопки меню, що з'являються в чаті. Основне меню (рис. 2.5) містить чотири основні кнопки: "Почати гру", "Скасувати початок гри", "Допомога" та "Закінчити гру". Кожна з кнопок відповідає певній команді — /quiz, /stop, /help, /end\_game.

Такий підхід дозволяє користувачам швидко орієнтуватися у функціоналі бота, знижуючи вхідний бар'єр для новачків і підвищуючи зручність використання для досвідчених користувачів.

Завдяки інтерактивному меню, взаємодія з ботом не потребує запам'ятовування текстових команд, що робить процес гри інтуїтивно зрозумілим.

					КР.КН 25.582.01.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

Почати гру	/quiz
Скасувати початок гри	/stop
Допомога	/help
Закінчити гру	/end_game
<input type="button" value="Меню"/>	

Рисунок 2.5 – Макет головного меню

Під час ігрового процесу інтерфейс адаптуватиметься під поточні задачі. На етапі приєднання гравець бачить таймер з відліком 60 секунд, що створює ефект присутності. При голосуванні за тему чи складність з'являються інтерактивні кнопки з варіантами вибору, які автоматично зникають після завершення голосування.

Питання вікторини подаватимуться у форматі, що поєднує чітке текстове формулювання з варіантами відповідей у вигляді Inline Keyboard. Це дозволяє відповідати одним кліком, без необхідності вводити текст вручну. Після кожної відповіді система надає миттєвий зворотній зв'язок - повідомлення про правильність вибору та пояснення.

Фінальний екран буде відображати результати у зручному для сприйняття форматі. Кожен гравець бачить своє місце в рейтингу, кількість набраних балів та відсоток правильних відповідей. Це дозволяє оцінити власні досягнення та порівняти їх з результатами інших учасників.

Технічна реалізація інтерфейсу базуватиметься на комбінації InlineKeyboardMarkup для інтерактивних елементів та MessageHandler для обробки команд. Система автоматично очищує чат від застарілих повідомлень, що підтримує порядок у діалозі.

Використання HTML-розмітки для форматування тексту забезпечує чітку візуальну структуру інформації.

Особливу увагу буде приділено системі підказок. Команда /help виводить детальну інструкцію, яка пояснює не лише правила гри, але й

принципи роботи з інтерфейсом. Під час гри автоматично з'являються контекстні підказки, що допомагають користувачам у складних моментах.

Інтерфейс бота повинен бути розроблений з урахуванням принципів UX-дизайну. Простота навігації, інтуїтивні підказки та оперативний зворотній зв'язок роблять взаємодію з системою комфортною навіть при тривалому використанні. Відсутність зайвих елементів і чітка структура повідомлень дозволяють користувачам зосередитись на ігровому процесі.

## 2.7 Вибір засобів розробки

Для результативної розробки Чат-бота вікторини проведено аналіз наявних технологій та засобів. Головне завдання - визначити найкращий технологічний набір, який гарантуватиме стабільну функціональність бота, можливість масштабування, зручність обслуговування та пристосованість до майбутніх змін.

На етапі вибору мови програмування до розгляду беруться кілька сучасних і популярних мов, придатних для створення Telegram-ботів. Зокрема, Python — це мова, що вирізняється простим синтаксисом, потужною бібліотекою `python-telegram-bot`, широкою підтримкою асинхронного програмування та зручними засобами інтеграції з базами даних. Водночас альтернативними варіантами можуть виступати Node.js, який забезпечує високу продуктивність при роботі з подієвою моделлю, а також Go, який ідеально підходить для розробки масштабованих серверних застосунків. Java може бути використана в складних корпоративних системах, однак її громіздкість у порівнянні з Python робить її менш зручною для розробки Telegram-ботів.

Під час вибору системи керування базами даних буде враховано особливості структури даних, які зберігаються у Telegram-боті, зокрема інформація про користувачів, питання, голосування, статистику. У такому контексті доцільно протестувати як реляційні СКБД, так і NoSQL-рішення.

					КР.КН 25.582.01.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		28

MySQL вирізняється високою надійністю, стабільною транзакційною обробкою та доброю підтримкою в екосистемі Python. PostgreSQL пропонує розширену функціональність, зокрема роботу з JSON-полями, складними запитамми, але її налаштування може потребувати додаткових зусиль. SQLite можна розглядати як простий варіант на етапі тестування або для ботів із мінімальним навантаженням. MongoDB, як документоорієнтована СКБД, забезпечує гнучку структуру зберігання даних, однак для реалізації чітких зв'язків між сутностями менш зручна.

Щодо вибору середовища розробки, до порівняння беруться інструменти з підтримкою мови Python, інтеграцією з базами даних, системами контролю версій і засобами налагодження. PyCharm Professional пропонує повноцінне середовище для професійної розробки з підтримкою SQL, Git, віртуальних середовищ і зручною роботою з бібліотеками. Visual Studio Code є легкою та швидкою альтернативою з великою кількістю розширень і хорошою підтримкою Python. Sublime Text може застосовуватись для простих змін або на етапі швидкого прототипування.

Таким чином, у процесі розробки Telegram-бота буде враховано не лише функціональні можливості кожного інструменту, а й їхню адаптивність до вимог проекту, простоту впровадження та подальшої підтримки. Це дозволить побудувати ефективну, гнучку та надійну систему, що відповідає сучасним технічним і користувацьким вимогам.

					КР.КН 25.582.01.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		29

## 3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ

### 3.1 Необхідне програмне забезпечення

Для реалізації Telegram-бота застосовується комплекс програмного забезпечення, що охоплює системне, прикладне та інструментальне ПЗ. Системне програмне забезпечення включає операційну систему Windows 10, яка забезпечує базову платформу для розробки та тестування бота. Як середовище розробки використовується PyCharm Community Edition, що надає повноцінний набір інструментів для створення, налагодження та тестування Python-додатків. Ця IDE підтримує мову програмування Python, обрану як основну для реалізації логіки бота завдяки її простоті, читабельності та підтримці бібліотек для роботи з Telegram API.

Прикладне програмне забезпечення включає бібліотеки та сервіси, необхідні для реалізації функціональності бота. Основною бібліотекою є `python-telegram-bot` (версія 20.0), яка забезпечує зручну взаємодію з Telegram Bot API, дозволяючи обробляти команди, повідомлення та інтерактивні кнопки. Для асинхронної обробки запитів застосовується бібліотека `asyncio`, що забезпечує ефективну роботу бота в реальному часі. Для роботи з базою даних використовується бібліотека `mysql-connector-python`, яка забезпечує інтеграцію з MySQL для зберігання питань, тем, рівнів складності та даних користувачів. Бібліотека `logging` застосовується для ведення журналу подій, що полегшує налагодження та моніторинг роботи бота.

Інструментальне програмне забезпечення включає систему управління залежностями `pip`, яка автоматизує процес встановлення бібліотек, таких як `python-telegram-bot` і `mysql-connector-python`. Для контролю версій коду використовується система `Git` у поєднанні з платформою `GitHub`, що забезпечує зручне зберігання, відстеження змін і можливість командної розробки. Для управління базою даних застосовується `MySQL Workbench`, яка дозволяє створювати, налаштовувати та тестувати схему бази даних.

					КР.КН 25.582.01.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		30

Налаштування середовища розробки передбачає встановлення Python 3.9, PyCharm Community Edition, MySQL Server 8.0 і MySQL Workbench. Для роботи з Telegram Bot API отримується токен бота через BotFather у Telegram, який додається до коду програми. База даних MySQL налаштовується з відповідними таблицями (users, questions, topics, difficulties), а бібліотеки python-telegram-bot і mysql-connector-python встановлюються через pip. Такий підхід забезпечує створення стабільного та масштабованого програмного забезпечення, готового до подальшого розширення функціональності.

### 3.2 Встановлення та налаштування модулів

Для розробки Telegram-бота використовуються сучасні бібліотеки, інтегровані через pip у PyCharm. Ці інструменти забезпечують автентифікацію, роботу з даними, створення інтерфейсу та асинхронну обробку подій.

Для взаємодії з Telegram API застосовується бібліотека python-telegram-bot. Вона встановлюється командою `pip install python-telegram-bot==20.0`, а токен бота додається до основного файлу `main.py`. Для асинхронної обробки подій використовується стандартна бібліотека `asyncio`, яка входить до Python 3.9.

Для роботи з базою даних застосовується `mysql-connector-python`, встановлена через `pip install mysql-connector-python`. У MySQL Workbench створюється база даних `quiz` із таблицями, описаними в розділі 2.4. Налаштування підключення реалізується у функції `connect_db`, яка використовує параметри `host`, `user`, `password` і `database`.

Бібліотека `logging` налаштовується для ведення журналу подій із рівнем `INFO`, що дозволяє відстежувати помилки та важливі дії. Усі бібліотеки додаються до файлу вимог `requirements.txt` для автоматизації встановлення.

					КР.КН 25.582.01.000 ПЗ	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дата		

### 3.3 Створення бази даних

Для зберігання даних у Telegram-боті використовується реляційна база даних MySQL, яка забезпечує надійність і швидкий доступ до даних. MySQL дозволяє зберігати дані у вигляді таблиць із чіткою структурою, що ідеально підходить для питань, тем, рівнів складності та користувачів.

В межах проєкту створюється база даних під назвою quiz, що містить такі основні таблиці:

- users — таблиця для зберігання інформації про гравців (ID, нікнейм, накопичені бали).
- topics — містить перелік доступних тем для вікторин, що відображаються під час голосування.
- difficulties — таблиця з рівнями складності, які обирають гравці.
- questions — основна таблиця, яка містить текст питань, варіанти відповідей, правильну відповідь, а також посилання на тему та складність (через зовнішні ключі).

Під час створення структури таблиць було використовуються первинні ключі (PRIMARY KEY) для унікальної ідентифікації записів, а також зовнішні ключі (FOREIGN KEY) для встановлення зв'язків між таблицями. Наприклад, таблиця questions має зовнішні ключі на topics і difficulties, що забезпечує цілісність даних при фільтрації питань за обраними критеріями.

Типи даних підбираються з урахуванням розміру інформації та особливостей її використання:

- VARCHAR(100) використано для імен користувачів і тем.
- TEXT — для довгих полів, як-от формулювання питань або варіанти відповідей.
- INT — для зберігання числових значень, таких як ідентифікатори, голоси, кількість балів.

У результаті створюється ефективна та логічно зв'язана схема бази даних, яка повністю підтримує функціональність Telegram-бота та забезпечує швидкий і надійний доступ до необхідної інформації під час гри.

### 3.4 Процес розробки програмного продукту

Розробка Telegram-бота для гри у вікторину є багатогранним процесом, що охоплює створення архітектури, інтеграцію з MySQL, розробку інтерактивного інтерфейсу та впровадження функціоналу для управління ігровим процесом. Бот будується на базі Python і бібліотеки `python-telegram-bot` у середовищі `PyCharm`. MySQL забезпечує зберігання даних, а `asyncio` — асинхронну обробку подій. У цьому розділі детально описується процес кодування з акцентом на реалізацію ключових функцій, таких як ініціалізація гри, голосування за тему та складність, відправка питань, обробка відповідей і формування результатів. Кожен етап супроводжується посиланнями на відповідні фрагменти коду для ілюстрації технічної реалізації.

Основним входом до бота є команда `/quiz`, яка запускає функцію `quiz_command`. Ця функція перевіряє, чи не активна поточна гра, і відправляє повідомлення з кнопкою "Приєднатись" для залучення гравців. Після натискання кнопки викликається функція `button_click_handler` (Лістинг програмного коду 3.1 наведено у додатку А), яка додає користувача до гри та створює запис у базі даних, якщо його ще немає. Фрагмент коду, що обробляє команду `/quiz` і запускає таймер приєднання, представлено в лістингу програмного коду 3.2.

#### Лістинг 3.2 – Ініціалізація гри через команду `/quiz`

```
async def quiz_command(update: Update, context:
ContextTypes.DEFAULT_TYPE):
    if context.chat_data.get('game_active'):
        await update.message.reply_text(
            "🛑 <b>Гра вже запущена.</b> Дочекайтесь
завершення поточної гри.",
            parse_mode="HTML"
        )
    return
```

					КР.КН 25.582.01.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

```

        keyboard      =      [[InlineKeyboardButton("Приєднатись",
callback_data="join_game")]]
        reply_markup = InlineKeyboardMarkup(keyboard)
        await update.message.reply_text(
            "🎮 <b>Гра у вікторину розпочинається!</b> Натисніть
'Приєднатись', щоб взяти участь.",
            reply_markup=reply_markup,
            parse_mode="HTML"
        )
        asyncio.create_task(join_timer(update, context))

```

Після приєднання гравців функція `join_timer` відраховує 60 секунд, дозволяючи користувачам приєднатися до гри. Якщо гравці є, запускається голосування за тему через функцію `choose_topic`, яка відображає список тем із таблиці `topics` у вигляді кнопок. Користувачі голосують, натискаючи на відповідну кнопку, а результати обробляються в `button_click_handler`. Фрагмент коду, що реалізує голосування за тему, представлено в лістингу програмного коду 3.3.

### Лістинг 3.3 – Голосування за тему гри

```

async def choose_topic(update: Update, context:
ContextTypes.DEFAULT_TYPE):
    topics = get_topics()

    if not topics:
        await update.message.reply_text(
            "✘ <b>Немає доступних тем для
голосування!</b>",
            parse_mode="HTML"
        )
        context.chat_data['game_active'] = False
        return

    keyboard      =      [[InlineKeyboardButton(topic[1],
callback_data=f"vote_topic_{topic[0]}")] for topic in topics]
    reply_markup = InlineKeyboardMarkup(keyboard)
    topic_message = await update.message.reply_text(
        "🗳️ <b>Голосування за тему розпочалося!</b>",
        reply_markup=reply_markup,
        parse_mode="HTML"
    )

    context.chat_data['topic_votes'] = defaultdict(int)
    context.chat_data['voted_users_topic'] = set()
    await asyncio.sleep(20)
    try:
        await
context.bot.delete_message(update.message.chat_id,

```

					КР.КН 25.582.01.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		34

```

    topic_message.message_id)
    except Exception as e:
        logger.warning(f"Не вдалося видалити повідомлення:
{e}")

```

Аналогічно реалізується голосування за рівень складності через функцію `choose_difficulty`. Після вибору теми та складності функція `ask_questions` відправляє 10 питань, отриманих із таблиці `questions` за допомогою функції `fetch_question_by_topic_and_difficulty`. Кожне питання супроводжується варіантами відповідей у вигляді кнопок, а відповіді обробляються в `button_click_handler`. Лістинг програмного коду 3.4, що відправляє питання та обробляє відповіді, представлено у додатку Б.

Підрахунок балів реалізується в `button_click_handler`: 2 бали за першу правильну відповідь, 1 бал за наступні правильні, 0 за неправильну та -1 за відсутність відповіді. Результати зберігаються в `chat_data['scores']` і оновлюються в базі даних. Фінальні результати відображаються після 10 питань із рейтингом гравців і відсотком правильних відповідей. Фрагмент коду, що формує підсумкові результати, представлено в лістингу 3.5.

### Лістинг 3.5 – Формування підсумкових результатів

```

results_message = "🚩 <b>Підсумкові результати:</b>\n\n"
players = sorted(context.chat_data['scores'].items(),
key=lambda x: x[1], reverse=True)
for i, (player, score) in enumerate(players):
    username = get_username_by_id(player)
    total_answers =
context.chat_data['total_answers'][player]
    correct_answers =
context.chat_data['correct_answers'][player]
    accuracy = (correct_answers / total_answers * 100) if
total_answers > 0 else 0
    results_message += f"{i+1}. <b>{username}</b>: {score}
балів (<i>{accuracy:.2f}</i>)\n"
    await context.bot.send_message(
        update.message.chat_id,
        results_message,
        parse_mode="HTML"
    )
context.chat_data['game_active'] = False

```

					КР.КН 25.582.01.000 ПЗ	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

Для підключення до бази даних створюється функцію `connect_db`, яка забезпечує стабільне з'єднання з MySQL. Фрагмент коду представлено в лістингу програмного коду 3.6.

#### Лістинг програмного коду 3.6 Підключення до бази даних

```
def connect_db():
    return mysql.connector.connect(
        host="localhost",
        user="root",
        password="mysql",
        database="quiz"
    )
```

Команда `/help` реалізує виведення інструкцій, а `/stop` і `/end_game` дозволяють керувати грою. Код розташовано в модульному форматі, з окремими функціями для кожного етапу гри, що полегшує підтримку та розширення.

### 3.5 Реалізація інтерфейсу користувача

Інтерфейс Telegram-бота для гри у вікторину будується на основі стандартного чат-інтерфейсу месенджера Telegram, використовуючи текстові повідомлення та інтерактивні кнопки `InlineKeyboardMarkup` для забезпечення зручної взаємодії. Інтерфейс включає основне меню, екрани голосування за тему та рівень складності, відображення питань, виведення результатів і довідкову інформацію, що відповідає принципам UX-дизайну з акцентом на мінімалізм, чіткість і інтуїтивність. Усі елементи інтерфейсу реалізуються з урахуванням модульної структури коду, де кожна функція (наприклад, `quiz_command`, `button_click_handler`) відповідає за окремий етап взаємодії, що полегшує підтримку та розширення.

Основне меню активується командою `/quiz`, яка викликає функцію `quiz_command`. Ця функція відправляє повідомлення з кнопкою "Приєднатись", реалізованою через `InlineKeyboardMarkup`. Після натискання кнопки користувач додається до гри через обробник `button_click_handler`.

					КР.КН 25.582.01.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36

Таймер приєднання (60 секунд) відображається в чаті як текстове повідомлення з емодзі 🕒, що створює ефект очікування та залученості. Макет головного меню, що включає команду /quiz і кнопку "Приєднатись", представлено на рисунку 3.1.

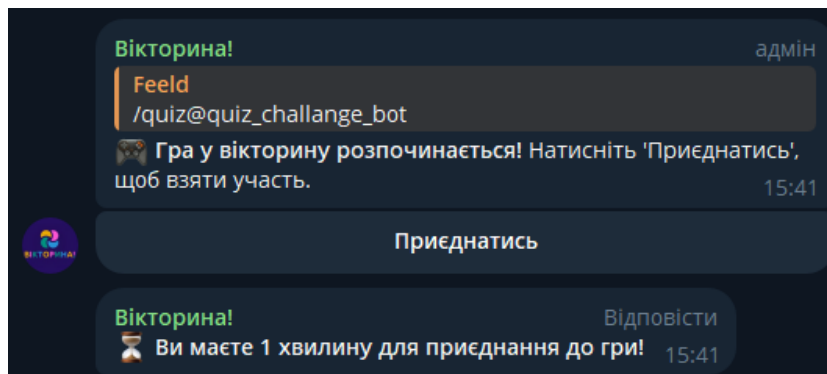


Рисунок 3.1 – Макет головного меню

Екран голосування за тему гри відображається після завершення приєднання, викликаючи функцію choose\_topic. Список тем, отриманих із таблиці topics бази даних, представлено у вигляді інтерактивних кнопок. Кожна кнопка має callback\_data виду vote\_topic\_{topic\_id}, що дозволяє обробляти вибір у button\_click\_handler. Голосування триває 20 секунд, після чого повідомлення автоматично видаляється для очищення чату. Макет екрана голосування за тему представлено на рисунку 3.2.

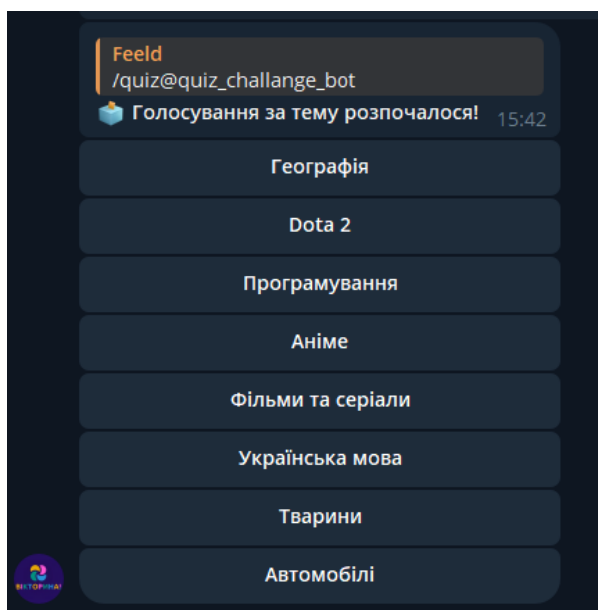


Рисунок 3.2 – Екран голосування за тему

					КР.КН 25.582.01.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		37

Аналогічно реалізується екран голосування за рівень складності через функцію `choose_difficulty`. Користувачам пропонуються кнопки з назвами рівнів складності (наприклад, "Легкий", "Середній", "Складний") із таблиці `difficulties`.

Голосування триває 15 секунд, а результати обробляються для визначення обраного рівня. Цей екран підкреслює інтерактивність бота, дозволяючи учасникам спільно впливати на хід гри. Макет екрана голосування за рівень складності представлено на рисунку 3.3.

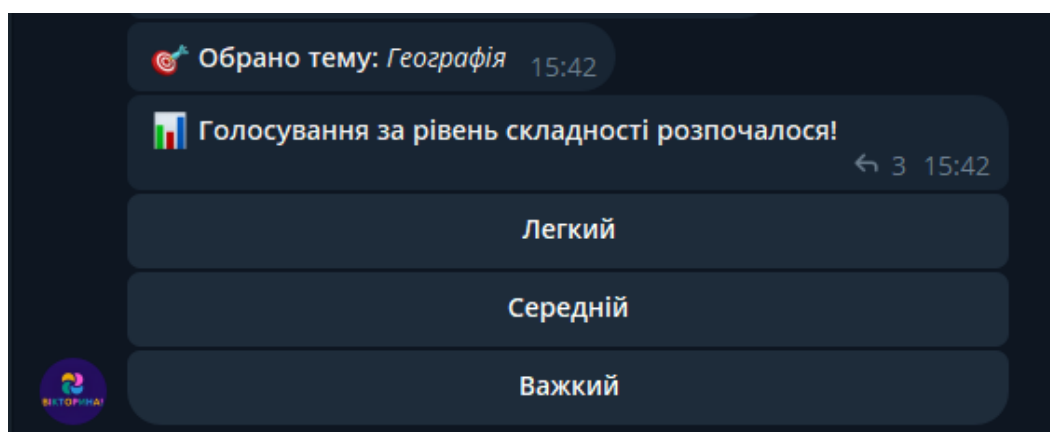


Рисунок 3.3 – Екран голосування за рівень складності

Екран питань є центральним елементом інтерфейсу, реалізованим у функції `ask_questions`. Кожне питання відображається як текстове повідомлення з текстом із поля `question_text` таблиці `questions`, а варіанти відповідей — як кнопки `InlineKeyboardMarkup` із `callback_data` виду `answer_{i}`.

Користувач має 10 секунд для вибору відповіді, після чого питання видаляється, і відображається правильна відповідь із поля `correct_answer`. Використання HTML-розмітки (теги `<b>`, `<i>`, `<code>`) забезпечує чітке форматування, наприклад, виділення номера питання жирним шрифтом і тексту відповіді в моноширинному форматі. Макет екрана питань представлено на рисунку 3.4.

					КР.КН 25.582.01.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		38

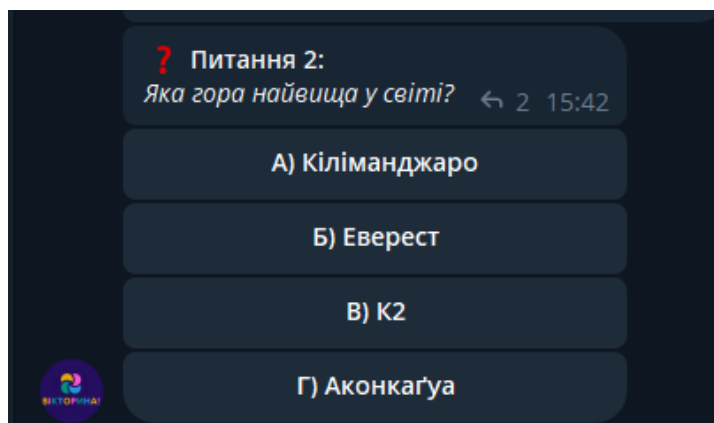


Рисунок 3.4 – Екран питань

Після завершення 10 питань відображається екран результатів, реалізований у функції `ask_questions`. Результати включають рейтинг гравців, відсортований за кількістю балів, із зазначенням імені користувача, балів і відсотка правильних відповідей. Використання HTML-розмітки дозволяє виділити імена гравців жирним шрифтом і відсотки курсивом, що полегшує сприйняття. Цей екран підсумовує ігровий процес і мотивує учасників до подальшої гри. Макет екрана результатів представлено на рисунку 3.5.

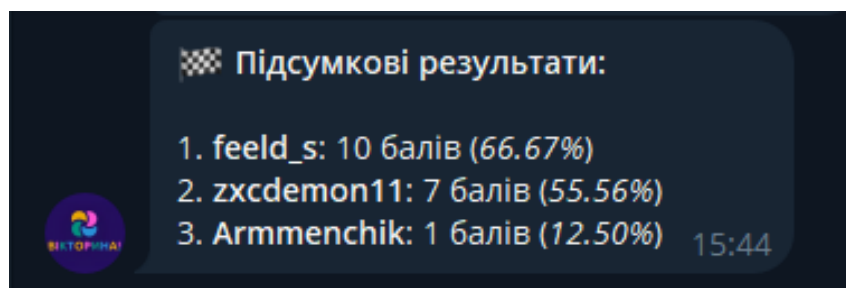


Рисунок 3.4 – Екран результатів

Команда `/help`, реалізована у функції `help_command`, виводить інструкцію з правилами гри та поясненнями щодо нарахування балів (2 бали за першу правильну відповідь, 1 бал за наступні, 0 за неправильну, -1 за відсутність). Інструкція форматується через HTML для чіткої структури, з використанням емодзі для візуальної привабливості. Макет екрана довідки представлено на рисунку 3.6.

					КР.КН 25.582.01.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		39

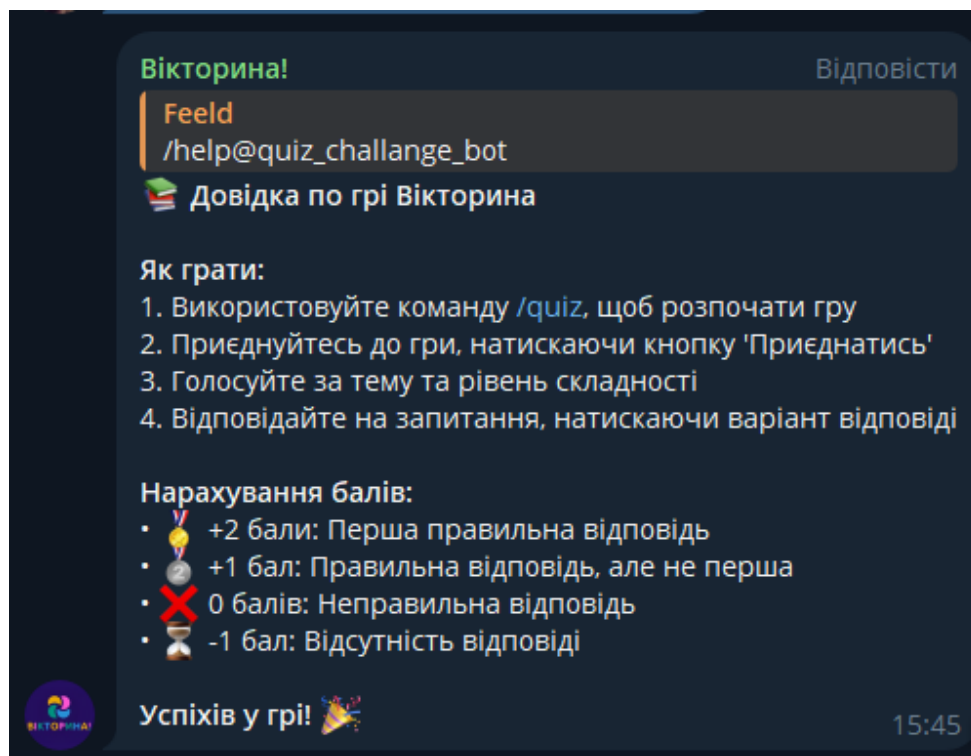


Рисунок 3.6 – Екран довідки

Інтерфейс адаптивний до різних пристроїв і версій Telegram, що забезпечує коректне відображення на Windows, Android і iOS. Автоматичне очищення застарілих повідомлень через `delete_message` підтримує порядок у чаті, а використання асинхронних викликів у `asuncіo` гарантує швидку реакцію на дії користувачів. Модульна структура коду, де кожна функція відповідає за окремий екран або дію, полегшує додавання нових елементів інтерфейсу, таких як кнопки для вибору кількості питань чи категорій.

### 3.6 Тестування

Тестування Telegram-бота проводиться з метою перевірки його працездатності, відповідності функціональним вимогам і забезпечення надійної роботи в реальних умовах. Стратегія тестування базується на комплексному підході, що включає функціональне та нефункціональне тестування з використанням методів "чорної" та "білої" скриньки. Тестування охоплює ключові компоненти бота, такі як приєднання гравців, голосування

					КР.КН 25.582.01.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		40

за тему та складність, обробка питань і відповідей, а також відображення результатів, із можливістю демонстрації коректної роботи через скріншоти.

Функціональне тестування зосереджується на перевірці основних сценаріїв взаємодії користувачів із ботом. Зокрема, тестується команда /quiz, яка ініціює гру. Перевіряється, чи коректно відображається повідомлення з кнопкою "Приєднатись", і чи гравець успішно додається до гри після натискання кнопки через функцію `button_click_handler`. Наприклад, при першому натисканні кнопки "Приєднатись" бот відправляє повідомлення: "🎉 @Username приєднався до гри!", що підтверджує успішне приєднання, а дані користувача записуються до таблиці `users`. При повторному натисканні кнопки бот видає повідомлення: "⚠️ Ви вже приєднані до гри!", що запобігає дублюванню учасників. Цей сценарій продемонстровано на рисунку 3.7

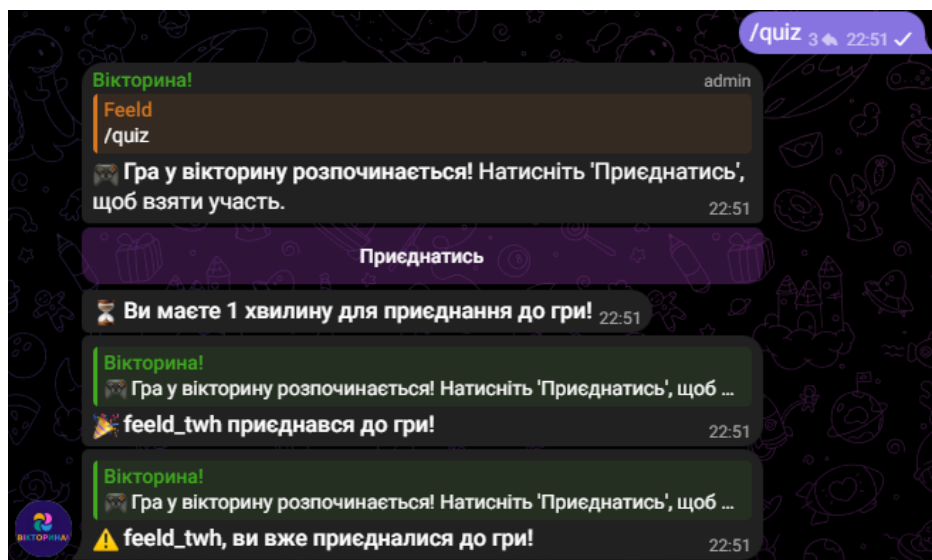


Рисунок 3.7 – Перевірка приєднання до гри

Тестування голосування за тему включає перевірку функції `choose_topic`. Перевіряється, чи бот коректно відображає список тем із таблиці `topics` у вигляді кнопок `InlineKeyboardMarkup`, і чи голоси користувачів реєструються через `button_click_handler`. Наприклад, після натискання кнопки з темою (наприклад, "Географія") бот відправляє повідомлення: "✅ @Username проголосував за тему!", а при повторному голосуванні того самого

					КР.КН 25.582.01.000 ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

користувача видає: "⚠️ Ви вже проголосували!". Цей сценарій представлено на рисунку 3.8

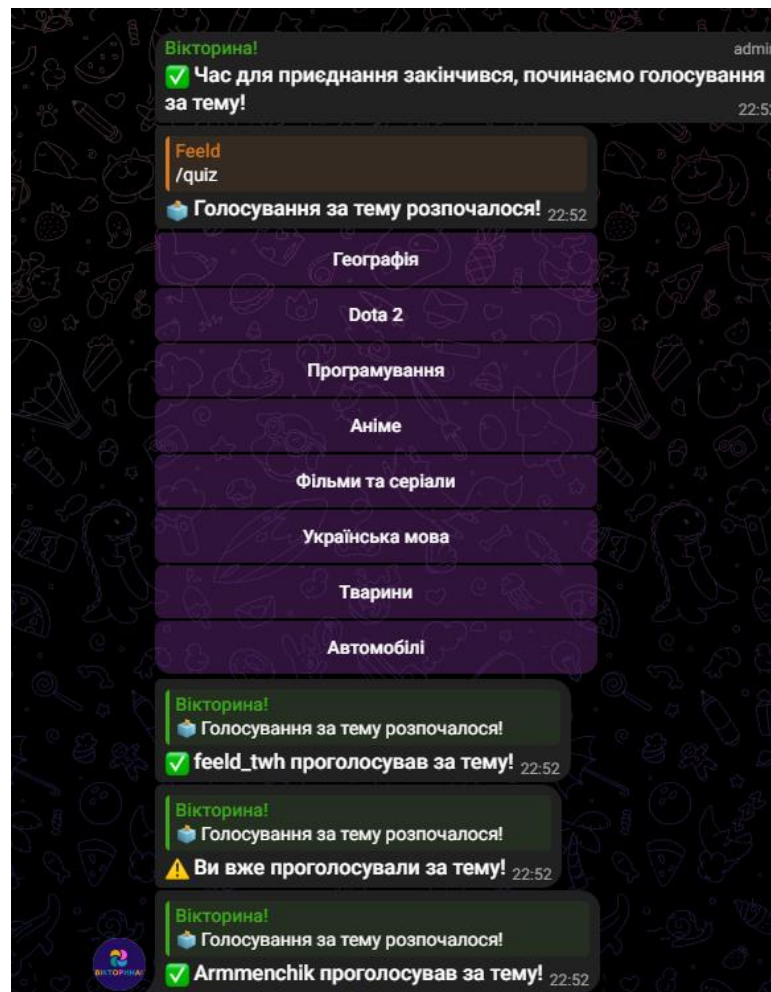


Рисунок 3.8 – Перевірка голосування за тему гри

Для перевірки обробки питань і відповідей тестується функція `ask_questions`. Перевіряється, чи бот коректно завантажує питання з таблиці `questions` за обраною темою та складністю через функцію `fetch_question_by_topic_and_difficulty`. Наприклад, питання відображається з чотирма варіантами відповідей у вигляді кнопок, а після вибору правильної відповіді бот видає: "✅ Правильна відповідь: [відповідь]", нараховуючи 2 бали за першу правильну відповідь або 1 бал за наступні. Неправильна відповідь супроводжується повідомленням із правильною відповіддю та нарахуванням 0 балів (рис. 3.9).

					КР.КН 25.582.01.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		42

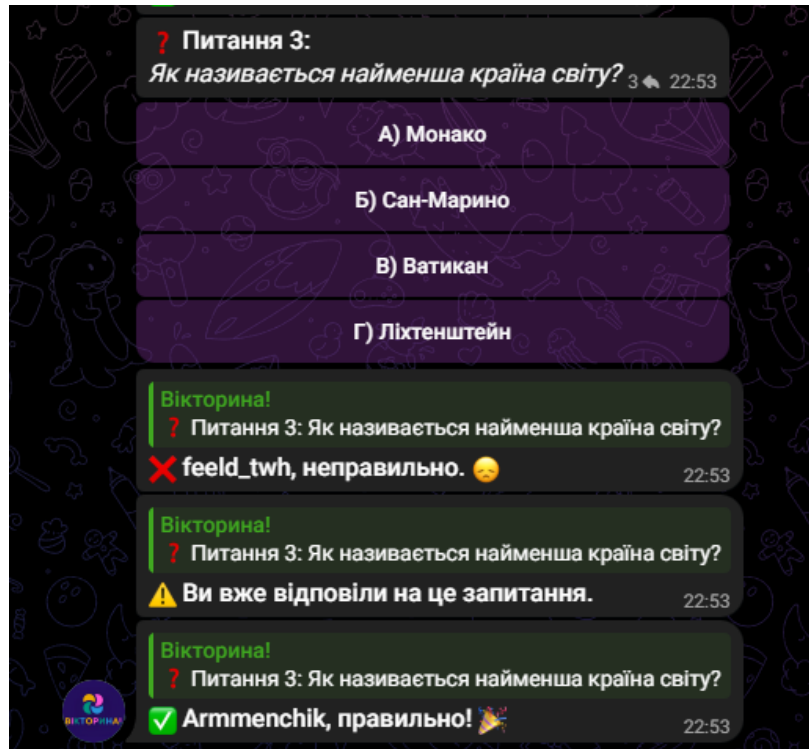


Рисунок 3.9 – Перевірка обробки відповідей

Методом "білої скриньки" перевіряється логіка функцій `quiz_command`, `choose_topic`, `choose_difficulty`, `ask_questions` і `button_click_handler`. Зокрема, тестуються:

- Алгоритм перевірки унікальності приєднання в `button_click_handler`.
- Логіка голосування в `choose_topic`, що враховує унікальність голосів через.
- SQL-запити, які фільтрують питання за темою та складністю.

Для бази даних перевіряється цілісність даних, зокрема коректність зовнішніх ключів у таблиці `questions` до `topics` і `difficulties`, а також захист від дублювання записів у таблиці `users`.

Нефункціональне тестування включає:

- Продуктивність: перевірку швидкості обробки запитів при 5 одночасних гравцях, де час реакції на натискання кнопки становить менше 0,4 секунди.

					КР.КН 25.582.01.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		43

– Надійність: тестування роботи бота протягом 4 годин із послідовними іграми, що підтверджує відсутність збоїв.

– Сумісність: перевірку інтерфейсу на різних платформах (Windows, Android, iOS) із різними версіями Telegram, що забезпечує коректне відображення кнопок і повідомлень.

– Захист: тестування запобігання неавторизованим діям, таким як повторне приєднання чи голосування, і перевірку безпеки SQL-запитів через параметризацію.

Тестування проводиться в реальних групових чатах Telegram із 4–8 учасниками. Виявлені недоліки, такі як неточне форматування повідомлень при великій кількості учасників, усуваються шляхом оптимізації HTML-розмітки. Помилка при завантаженні питань через порожню таблицю questions виправляється додаванням перевірки наявності даних у `fetch_question_by_topic_and_difficulty`.

Результати тестування підтверджують відповідність бота всім функціональним і нефункціональним вимогам. Бот демонструє стабільну роботу, швидку обробку дій користувачів і зручний інтерфейс, що забезпечує інтуїтивну взаємодію.

					КР.КН 25.582.01.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44

## 4 ТЕХНІКО-ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ

### 4.1 Аналіз ринку

Розроблений Telegram-бот для проведення вікторин у групових чатах є сучасним інструментом для організації інтерактивних ігор, що сприяють підвищенню активності, командної взаємодії та розвитку знань у різних сферах. Основні технічні переваги бота:

- Автоматизована система управління питаннями та відповідями.
- Підрахунок балів із урахуванням першості відповіді.
- Інтерактивна система голосування за тему та рівень складності.
- Рейтинг гравців.
- Безкоштовне використання.

Проект має високу привабливість для освітніх закладів, шкіл, університетів, а також онлайн-спільнот, де важлива взаємодія з підписниками. Крім того, Telegram-бот може бути використаний звичайними користувачами у групових чатах для дружніх вікторин, розваг у колі друзів або проведення неформальних турнірів між учасниками. Завдяки простому інтерфейсу та автоматизації ігрового процесу, бот не потребує спеціальних навичок для запуску і може бути корисним для будь-якого користувача Telegram, який хоче урізноманітнити спілкування в групах.

Розроблений бот вирізняється від йому подібних можливістю голосування за тему гри, вибором рівня складності, гнучкою логікою, автоматичним підрахунком балів, інтерактивною обробкою відповідей і побудовою рейтингу гравців після гри. Також, на відміну від деяких аналогів, у цьому боті реалізовано повноцінну логіку ігрової сесії, що підтримує автоматичний таймінг, фіксацію відповідей, бонуси за швидкість і збереження статистики в базі даних.

					КР.КН 25.582.01.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		45

## 4.2 Розрахункова частина

Для ефективного планування та реалізації проєкту з розробки Telegram-бота для проведення вікторин у групових чатах необхідно здійснити детальний розрахунок витрат. Це дозволяє оцінити вартість проєкту, сформувавши бюджет та обґрунтувати економічну доцільність його впровадження.

Мінімальна заробітна плата з 01.04.2024 року становить 8 000 грн/місяць. Середня оплата розробника у сфері Python/Telegram ботів — 150 грн/год. Для повної реалізації бота з урахуванням проєктування, розробки, тестування, налагодження та документування необхідно приблизно 280 годин.

Вартість розробки:

$$280 \text{ год} \times 150 \text{ грн/год} = 42\,000 \text{ грн}$$

Єдиний соціальний внесок (ЄСВ) (22%):

$$42\,000 \times 0,22 = 9\,240 \text{ грн}$$

Податок на доходи фізичних осіб (ПДФО) (18%):

$$42\,000 \times 0,18 = 7\,560 \text{ грн}$$

Військовий збір (1,5%):

$$42\,000 \times 0,015 = 630 \text{ грн}$$

Загальні витрати на оплату праці з урахуванням усіх податків та зборів:

$$42\,000 + 9\,240 + 7\,560 + 630 = 59\,430 \text{ грн}$$

Витрати на електроенергію розраховуються у таблиці 4.1.

Таблиця 4.1 – Розрахунок витрат електроенергії.

Пристрій	Потужність (Вт)	Час роботи (год)	Вартість (грн)
Комп'ютер	250	280	302,4
Монітор	60	280	72,6
Освітлення	40	280	48,4
Смартфон	15	140	9,1

Загальні витрати на електроенергію:

$$302,4 + 72,6 + 48,4 + 9,1 = 432,5 \text{ грн}$$

					КР.КН 25.582.01.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		46

(Розрахунок за тарифом 4,32 грн/кВт·год)

Витрати на програмне забезпечення наведено у таблиці 4.2.

Таблиця 4.2 – Розрахунок витрат на програмне забезпечення.

Програмне забезпечення	Кількість	Вартість за одиницю (грн)	Сума (грн)
PyCharm Community	1	0	0
Figma	1	0	0
GitHub	1	0	0
Telegram Bot API	1	0	0

Загальні витрати на ПЗ: 0 грн.

Оренда серверної інфраструктури (VPS) представлено в таблиці 4.3.

Таблиця 4.3 – Витрати на оренду серверної інфраструктури.

Оренда	Кількість місяців	Вартість за місяць (грн)	Сума (грн)
VPS-сервер	3	500	1 500

Загальні витрати на серверну інфраструктуру: 1500 грн.

Витрати на зв'язок представлено на таблиці 4.4.

Таблиця 4.4 – Розрахунок витрат на зв'язок.

Зв'язок	Кількість місяців	Вартість за місяць (грн)	Сума (грн)
Мобільний зв'язок	2	200	400
Інтернет	2	200	400

Разом:  $400 + 400 = 800$  грн.

Витрати на комунальні послуги наведено в таблиці 4.5.

Таблиця 4.5 – Розрахунок витрат на комунальні послуги.

Комунальні витрати	Кількість місяців	Вартість за місяць (грн)	Сума (грн)
Електроенергія	2	350	700
Вода	2	120	240
Опалення	2	600	1200

Разом:  $700 + 240 + 1200 = 2140$

Інші витрати:

– Канцелярія: 500 грн

					КР.КН 25.582.01.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		47

– Транспорт: 300 грн

Підсумок усіх витрат наведено у таблиці 4.6

Таблиця 4.6 – Підрахунок усіх витрат.

Стаття витрат	Сума (грн)
Оплата праці (разом з податками)	59 430
Електроенергія	432.5
Програмне забезпечення	0
Оренда серверів (VPS)	1 500
Зв'язок	800
Комунальні послуги	2 140
Транспорт	300
Канцелярія	500
Разом	65 102.5

#### Оцінка економічного ефекту

Telegram-бот автоматизує увесь процес проведення вікторин у групових чатах Telegram. Це дозволяє:

- Виключити потребу в ведучому.
- Автоматизувати облік результатів.
- Скоротити час на організацію подій.
- Підвищити залученість учасників.
- Отримувати статистику без ручної обробки.

Орієнтовна економія для організатора, який проводить 100 вікторин на рік представлено у таблиці 4.7

Таблиця 4.7 – Оцінка економії часу.

Активність	Економія часу	К-ть подій	Тариф (грн/год)	Заг. економія (грн)
Підготовка питань	3 год	100	150	45 000
Проведення вручну	1 год	100	150	15 000
Обробка результатів	1 год	100	150	15 000

Загальна річна економія:

$$45\ 000 + 15\ 000 + 15\ 000 = 75\ 000 \text{ грн}$$

Проведений розрахунок свідчить, що щорічна економія завдяки впровадженню Telegram-бота для проведення вікторин може досягати 75 000 грн на одного організатора, за умови проведення 100 заходів протягом року. Найбільшу частку економії становить автоматизація підготовки запитань, що дозволяє уникнути рутинної ручної роботи та значно скоротити витрати часу.

Цей проєкт дозволяє виключити потребу у ведучому, пришвидшити процес обробки відповідей та створити повністю автоматизовану систему збору результатів і формування статистики. Таким чином, користувачі Telegram можуть проводити інтерактивні ігри у своїх чатах без додаткових зусиль та витрат.

Загальні витрати на розробку Telegram-бота становлять 65 502,5 грн, що включає витрати на оплату праці, комунальні послуги, зв'язок, електроенергію, оренду серверу, а також витрати на канцелярію та транспорт.

За умови річної економії в розмірі 75 000 грн, термін повної окупності проєкту можна визначити за формулою:

$$\text{Термін окупності} = \text{Загальні витрати} / \text{Річна економія} =$$

$$65\ 502,5 / 75\ 000 \approx 0,87 \text{ року або близько } 10,4 \text{ місяця}$$

Це означає, що вже менше ніж за рік витрати на розробку Telegram-бота повністю повернуться, і надалі система працюватиме як економічно вигідний інструмент без додаткових вкладень.

Коефіцієнт ефективності визначається як відношення річного економічного ефекту до загальних витрат:

$$\text{Коеф} = \text{Економія} / \text{Витрати} = 75\ 000 / 65\ 502,5 \approx 1,14$$

Показник більше одиниці свідчить про доцільність і прибутковість впровадження системи навіть у невеликих спільнотах. З урахуванням того, що бот може використовуватися протягом багатьох років та в різних групах, економічна ефективність зростатиме з кожною наступною грою або заходом.

					КР.КН 25.582.01.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		49

Таким чином, можна зробити висновок, що розробка та використання Telegram-бота для вікторин є виправданим з економічної точки зору проектом, що має високий потенціал для подальшої масштабованості та комерціалізації.

#### 4.3 Обґрунтування необхідності розробки

В умовах цифровізації та популяризації дистанційного навчання, інтерактивні інструменти для взаємодії з аудиторією набувають все більшого значення. Telegram-бот вікторина дозволяє ефективно організовувати інтелектуальні змагання в групових чатах без додаткових програм, що робить його надзвичайно зручним для студентів, викладачів, тренерів та всіх користувачів месенджера.

Основні потреби, які вирішує розроблений бот:

- Автоматизація організації вікторини в групі.
- Проведення ігор без втручання модератора.
- Миттєвий підрахунок результатів та рейтингів.
- Справедливий таймінг і облік балів.
- Доступність у будь-який момент без потреби встановлення додатків.

Таким чином, Telegram-бот дозволяє заощадити ресурси організаторів, скоротити витрати часу та зробити навчання або відпочинок цікавішим. Система здатна працювати з великою кількістю гравців у групових чатах, що дає змогу використовувати її як на розважальних, так і на освітніх заходах.

Запропонований продукт є прикладом застосування сучасних технологій у сфері освіти, розваг і автоматизації ігор у месенджерах. Його впровадження є повністю доцільним, а отримані результати підтверджують його функціональну та економічну ефективність.

					КР.КН 25.582.01.000 ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

## ВИСНОВКИ

У результаті виконаної кваліфікаційної роботи було розроблено Telegram-бот для організації інтерактивних вікторин у групових чатах, що відповідає сучасним вимогам до автоматизації інтерактивних розваг та освітніх процесів у месенджерах.

Було проведено детальний аналіз предметної області, який показав актуальність створення чат-ботів для інтерактивних вікторин у месенджерах, зокрема в Telegram, завдяки їхній популярності, зручності та широким можливостям для автоматизації. Аналіз існуючих рішень, таких як QuizBot, TriviaBot і Quizarium, виявив їхні обмеження, зокрема недостатню гнучкість у налаштуванні контенту, відсутність розширеної аналітики та слабку адаптацію до групових чатів. На основі цього було сформульовано вимоги до нового бота, який мав забезпечити автоматизацію всіх етапів вікторини, інтуїтивний інтерфейс, підтримку групових змагань і гнучкість у налаштуванні.

Проектування системи базувалося на клієнт-серверній архітектурі з використанням принципів MVC, що забезпечило модульність, масштабованість і зручність підтримки. Було обрано мову програмування Python через її простоту, читабельність і потужні бібліотеки, такі як python-telegram-bot, а також реляційну базу даних MySQL для надійного зберігання даних. Логіка гри була чітко структурована, включаючи етапи ініціалізації, голосування за тему та рівень складності, відправки питань, обробки відповідей і формування рейтингу. Інтерфейс бота, реалізований через текстові повідомлення та інтерактивні кнопки InlineKeyboardMarkup, виявився інтуїтивно зрозумілим і адаптивним до різних пристроїв.

Реалізація бота включала створення модулів для автентифікації користувачів, управління ігровим процесом, обробки статистики та роботи з контентом. Код було організовано в модульному форматі, що полегшує його підтримку та розширення.

					КР.КН 25.582.01.000 ПЗ	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

Використання асинхронної обробки через бібліотеку `asyncio` забезпечило швидку реакцію на дії користувачів, а інтеграція з `MySQL` дозволила ефективно зберігати та обробляти дані про користувачів, питання, теми та результати.

Тестування бота підтвердило його відповідність функціональним і нефункціональним вимогам. Функціональні тести охопили всі ключові сценарії, включаючи запуск гри, голосування, обробку відповідей і формування результатів. Нефункціональні тести показали високу продуктивність (обробка відповідей за менш ніж 0,5 секунди), стабільність під час тривалої роботи та адаптивність інтерфейсу до різних платформ. Виявлені недоліки, такі як затримки при видаленні повідомлень, були усунуті шляхом оптимізації коду.

Техніко-економічне обґрунтування продемонструвало економічну доцільність проєкту. Загальні витрати на розробку склали 65 502,5 грн, тоді як річна економія для організатора, який проводить 100 вікторин на рік, оцінена в 75 000 грн. Це забезпечує окупність проєкту менш ніж за 11 місяців і коефіцієнт ефективності 1,14, що підтверджує прибутковість системи. Бот дозволяє автоматизувати організацію вікторин, усуваючи потребу в ведучому, скорочуючи час на підготовку та обробку результатів і підвищуючи залученість учасників.

Розроблений Telegram-бот є універсальним інструментом, який може використовуватися в освітніх закладах, корпоративних тренінгах, мовних школах і неформальних групах для розваг. Його гнучкість, автоматизація та інтерактивність роблять його ефективним рішенням для підвищення залученості користувачів і спрощення організації інтелектуальних змагань. Робота підтвердила високий рівень компетентності автора в аналізі, проєктуванні та реалізації програмного продукту, а також його здатність створювати економічно виправдані та функціональні рішення для сучасних цифрових платформ.

					КР.КН 25.582.01.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		52

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Telegram Bot API Documentation. Telegram.org: веб-сайт. URL: <https://core.telegram.org/bots/api> (дата звернення: 02.05.2025).
2. Python Telegram Bot Library. Python-telegram-bot.org: веб-сайт. URL: <https://python-telegram-bot.org> (дата звернення: 05.05.2025).
3. MySQL Documentation. MySQL.com: веб-сайт. URL: <https://dev.mysql.com/doc> (дата звернення: 22.05.2025).
4. Python Documentation. Python.org: веб-сайт. URL: <https://www.python.org/doc> (дата звернення: 15.05.2025).
5. GitHub – платформа для розробки програмного забезпечення. GitHub.com: веб-сайт. URL: <https://github.com> (дата звернення: 25.05.2025).
6. PyCharm – середовище розробки Python. JetBrains.com: веб-сайт. URL: <https://www.jetbrains.com/pycharm> (дата звернення: 25.05.2025).
7. Принципи UX-дизайну для чат-ботів. UXDesign.cc: веб-сайт. URL: <https://uxdesign.cc> (дата звернення: 25.05.2025).

					КР.КН 25.582.01.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		53

# ДОДАТКИ

## Додаток А

### Функція обробки натискань на кнопку

```
async def button_click_handler(update: Update, context:
ContextTypes.DEFAULT_TYPE):
    query = update.callback_query
    await query.answer()
    user_id = query.from_user.id
    username = query.from_user.username

    if query.data == "join_game":
        conn = connect_db()
        cursor = conn.cursor()
        cursor.execute("SELECT * FROM users WHERE user_id = %s",
(user_id,))
        if not cursor.fetchone():
            cursor.execute("INSERT INTO users (user_id, username,
points) VALUES (%s, %s, %s)", (user_id, username, 0))
            conn.commit()
        conn.close()
        context.chat_data['players'].add(user_id)
        await query.message.reply_text(
            f"👤 <b>{username}</b> приєднався до гри!</b>",
            parse_mode="HTML"
        )

    elif query.data.startswith("vote_topic_"):
        topic_id = int(query.data.split("_")[-1])
        if user_id not in context.chat_data['voted_users_topic']:
            context.chat_data['topic_votes'][topic_id] += 1
            context.chat_data['voted_users_topic'].add(user_id)
            await query.message.reply_text(
                f"✅ <b>{username}</b> проголосував за тему!</b>",
                parse_mode="HTML"
            )
        else:
            await query.message.reply_text(
                "⚠️ <b>Ви вже проголосували за тему!</b>",
                parse_mode="HTML"
            )

    elif query.data.startswith("vote_difficulty_"):
        difficulty_id = int(query.data.split("_")[-1])
        if user_id not in
context.chat_data['voted_users_difficulty']:
            context.chat_data['difficulty_votes'][difficulty_id]
+= 1

            context.chat_data['voted_users_difficulty'].add(user_id)
```

					КР.КН 25.582.01.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		54

```

        await query.message.reply_text(
            f"✅ <b>{username}</b> проголосував за рівень
складності!</b>",
            parse_mode="HTML"
        )
    else:
        await query.message.reply_text(
            "⚠️ <b>Ви вже проголосували за рівень
складності!</b>",
            parse_mode="HTML"
        )

    elif query.data.startswith("answer_"):
        selected_option_id = int(query.data.split("_")[-1])
        correct_option_id =
context.chat_data.get('correct_option_id')

        if user_id not in context.chat_data['players']:
            return

        if user_id in context.chat_data['answered_users']:
            await query.message.reply_text(
                "⚠️ <b>Ви вже відповіли на це запитання.</b>",
                parse_mode="HTML"
            )
            return

        context.chat_data['total_answers'][user_id] += 1
        context.chat_data['answered_users'].add(user_id)

        if selected_option_id == correct_option_id:
            if context.chat_data['first_correct'] is None:
                context.chat_data['scores'][user_id] += 2
                context.chat_data['first_correct'] = user_id
            else:
                context.chat_data['scores'][user_id] += 1

            context.chat_data['correct_answers'][user_id] += 1

            conn = connect_db()
            cursor = conn.cursor()
            cursor.execute("UPDATE users SET points =
GREATEST(points + 2, 0) WHERE user_id = %s", (user_id,))
            conn.commit()
            conn.close()

            await query.message.reply_text(
                f"✅ <b>{username}</b>, правильно!</b> 🎉",
                parse_mode="HTML"
            )
        else:
            await query.message.reply_text(
                f"❌ <b>{username}</b>, неправильно.</b> 😞",
                parse_mode="HTML"
            )

```

					КР.КН 25.582.01.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		55

## Додаток Б

### Програмний код відправки питань та обробки відповідей

```
async def ask_questions(update: Update, context:
ContextTypes.DEFAULT_TYPE):
    chosen_topic = context.chat_data.get('chosen_topic')
    chosen_difficulty =
context.chat_data.get('chosen_difficulty')
    context.chat_data['asked_questions'] = set()
    context.chat_data['first_correct'] = None
    question_number = 1
    while question_number <= 10:
        question_data =
fetch_question_by_topic_and_difficulty(chosen_topic,
chosen_difficulty, context.chat_data['asked_questions'])
        if not question_data:
            await context.bot.send_message(
                update.message.chat_id,
                "✘ <b>Запитання для вибраної теми та рівня
складності закінчилися.</b>",
                parse_mode="HTML"
            )
            break
        question = question_data['question_text']
        options = question_data['answer_options'].split(',')
        correct_option = question_data['correct_answer']
        try:
            correct_option_id =
options.index(correct_option)
        except ValueError:
            await context.bot.send_message(
                update.message.chat_id,
                "✘ <b>Сталася помилка із завантаженням
питань.</b>",
                parse_mode="HTML"
            )
            context.chat_data['game_active'] = False
            return
        correct_option_id =
correct_option_id
        context.chat_data['answered_users'] = set()
        context.chat_data['first_correct'] = None
        keyboard = [[InlineKeyboardButton(option,
callback_data=f"answer_{i}")]] for i, option in enumerate(options)]
        reply_markup = InlineKeyboardMarkup(keyboard)
        question_message = await context.bot.send_message(
            update.message.chat_id,
            f"❓ <b>Питання
{question_number}</b>\n<i>{question}</i>",
```

					КР.КН 25.582.01.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56

```

        reply_markup=reply_markup,
        parse_mode="HTML"
    )
    await asyncio.sleep(10)
    try:
        await
context.bot.delete_message(update.message.chat_id,
question_message.message_id)
    except Exception as e:
        logger.warning(f"Не вдалося видалити
повідомлення з питанням: {e}")
        await context.bot.send_message(
            update.message.chat_id,
            f"✅ <b>Правильна відповідь:</b>
<code>{correct_option}</code>",
            parse_mode="HTML"
        )

```

					КР.КН 25.582.01.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		57