

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Створення чат-бота для Гончарівської ЗОШ І-ІІ ст. Золочівського району з використанням Python та Telegram API

Виконав: студент IV курсу, групи СНС-42

спеціальності 122 Комп'ютерні науки

(шифр і назва спеціальності)

(підпис)

Боднар Д.В.

(прізвище та ініціали)

Керівник

(підпис)

Гладь Ю.Б.

(прізвище та ініціали)

Нормоконтроль

(підпис)

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

Рецензент

(підпис)

(прізвище та ініціали)

Тернопіль
2024

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.
(підпис) (прізвище та ініціали)

«__» _____ 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студенту Боднару Денису Володимировичу
(прізвище, ім'я, s по батькові)

1. Тема роботи Створення чат-бота для Гончарівської ЗОШ І-ІІ ст. Золочівського району з використанням Python та Telegram API

Керівник роботи Гладьо Юрій Богданович, кандидат технічних наук, доцент кафедри ММ
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «29» квітня 2024 року № 4/7-472

2. Термін подання студентом завершеної роботи 24 червня 2024р.

3. Вихідні дані до роботи Mockapi.io, Sinoptik.ua, Railway.app.

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. Розділ 1 Теоретичні відомості чат-ботів: вивчення концепцій та принципів 1.1 Чат-бот.

1.2 Використання чат-ботів. 1.2.1 Чат-боти для підтримки та обслуговування клієнтів.

1.2.2 Чат-боти в освіті. 1.2.3 Чат-боти для SMM та управління соціальними мережами.

1.3 Переваги та недоліки чат-ботів. 1.4 Способи створення чат-ботів. 1.5 Висновок до першого

розділу. Розділ 2 Інструменти та технології для розробки чат-бота. 2.1 Огляд платформ для

створення чат-ботів. 2.1.1 Telegram. 2.1.2 Instagram. 2.1.3 Viber. 2.2 Середовище

програмування. 2.2.1 PyCharm. 2.2.2 Переваги і недоліки PyCharm. 2.3 Мова програмування.

2.3.1 Python. 2.3.2 Переваги і недоліки Python. 2.4 Telegram API. 2.5 Бібліотека Telebot.

2.6 Висновок до другого розділу. Розділ 3 Практична реалізація. 3.1 Опис функціоналу

чат-бота. 3.2 BotFather. 3.3 Головне меню. 3.4 Шкільний графік. 3.5 Події. 3.6 Погода.

3.7 Адмін-панель. 3.8 Висновок до третього розділу. Розділ 4 Безпека життєдіяльності, основи

охорони праці. 4.1 Менеджмент безпеки учнів і вчителів. 4.2 Заходи, що покращують умови

праці у школі. 4.3 Психофізіологічне розвантаження для вчителів та учнів. 4.4 Висновок до

четвертого розділу. Висновки. Перелік джерел. Додатки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Титульна сторінка. 2. Мета кваліфікаційної роботи. 3. Чат-боти. 4. Інструменти

використання. 5. Функціонал чат-бота Гончарівський заклад. 6. Створення бота у BotFather.

7. Кнопка «Головне меню». 8. Кнопка «Розклад дзвінків». 9. Кнопка «Розклад уроків».

10. Кнопка «Події». 11. Кнопка «Погода». 12. Панель для адміністратора. 13. Висновки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	Сенчишин В.С., доцент кафедри МТ		

7. Дата видачі завдання 29 січня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	30.01.2024	
2.	Підбір джерел про чат-боти	31.01.2024-03.02.2024	
3.	Опрацювання джерел по темі кваліфікаційної роботи про чат-бот для Гончарівського закладу	04.02.2024-06.02.2024	
4.	Виконання дослідження щодо понять про чат-бот Розроблення чат-бота для Гончарівської ЗОШ І-ІІ ст. Золочівського району з використанням Python та Telegram API	07.02.2024-11.02.2024	
5.	Оформлення розділу «Теоретичні відомості чат-ботів: вивчення концепцій та принципів»	03.06.2024-05.06.2024	
6.	Оформлення розділу «Інструменти та технології для розробки чат-боту»	06.06.2024-08.06.2024	
6.	Оформлення розділу «Практична реалізація»	09.06.2024-11.06.2024	
7.	Виконання завдання до підрозділу «Безпека життєдіяльності»		
8.	Виконання завдання до підрозділу «Основи охорони праці»		
9.	Оформлення кваліфікаційної роботи	16.06.2024-17.06.2024	
10.	Нормоконтроль	18.06.2024-19.06.2024	
11.	Перевірка на плагіат	20.06.2024	
12.	Попередній захист кваліфікаційної роботи	21.06.2024	
13.	Захист кваліфікаційної роботи	28.06.2024	

Студент

(підпис)

Боднар Д.В.

(прізвище та ініціали)

Керівник роботи

(підпис)

Гладь Ю.Б.

(прізвище та ініціали)

АНОТАЦІЯ

Створення чат-бота для Гончарівської ЗОШ І-ІІ ст. Золочівського району з використанням Python та Telegram API // Кваліфікаційна робота освітнього рівня «Бакалавр» // Боднар Денис Володимирович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СНс-42 // Тернопіль, 2024 // С. 65, рис. – 15, табл. – 0, през. – 13, додат. – 9, бібліогр. – 38.

Ключові слова: месенджер, розробка, чат-бот, школа, користувач, головне меню, повідомлення.

Кваліфікаційна робота присвячена дослідженню чат-бота для Гончарівської ЗОШ І-ІІ ст. Золочівського району з використанням Python та Telegram API з метою забезпечити для школярів і батьків оперативний доступ до важливої інформації.

У першому розділі кваліфікаційної роботи описано інформацію про чат-боти, їх використання в різних галузях, переваги та недоліки, а також способи створення.

В другому розділі проведено аналіз інструментів для роботи чат-боту, зокрема оглянуті платформи для створення, середовище програмування PyCharm, мова програмування Python, Telegram API та бібліотека Telebot.

У третьому розділі кваліфікаційної роботи викладено практичну реалізацію проекту. Описано функціонал чат-бота та процедуру створення його зовнішнього вигляду в Telegram за допомогою BotFather. Детально розглянуто як виглядає та функціонує головне меню, розклад дзвінків, події, погода та адміністративне меню.

Об'єктом дослідження є створений чат-бот для Гончарівської ЗОШ І-ІІ ст. Золочівського району, а предметом дослідження є його шкільний функціонал для користувачів.

ANNOTATION

Creation of a Chatbot for I-II Grades Goncharivska Secondary School of Zolochiv District Using Python and Telegram API // Qualification work of the educational level «Bachelor» // Bodnar Denys Volodymyrovych // Ternopil Ivan Pulyu National Technical University, Computer and Information Systems and Software Engineering Faculty, Computer Sciences Department, group SNs-42 // Ternopil, 2024 // P. 65, fig. – 15, tabl. – 0, pres. – 13, annexes. – 9, references – 38.

Keywords: messenger, development, chatbot, school, user, main menu, message.

The qualification work is devoted to the research of a chatbot for Goncharivska secondary school of the I-II grades. Zolochiv district using Python and Telegram API in order to provide schoolchildren and parents with prompt access to important information.

The first section of the qualification work describes information about chat bots, their use in various industries, advantages and disadvantages, as well as methods of creation.

In the second chapter, an analysis of the tools for the work of the chatbot was carried out, in particular, the reviewed platforms for creation, the PyCharm programming environment, the Python programming language, the Telegram API and the Telebot library.

The third section of the qualification work describes the practical implementation of the project. The functionality of the chatbot and the procedure for creating its appearance in Telegram using BotFather are described. How the main menu, call schedule, events, weather and administrative menu looks and functions are considered in detail.

The object of the research is a chatbot created for Goncharivska secondary school of the I-II grades. Zolochiv district, and the subject of research is its school functionality for users.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Direct – це внутрішній месенджер платформи, який дозволяє користувачам обмінюватися повідомленнями один з одним.

HTML (англ. Hypertext Markup Language) – це мова розмітки, яка використовується для структурування та відображення веб-сторінки та її вмісту.

ID (англ. Identity Document) – унікальна ознака об'єкта, що дозволяє відрізнити його від інших об'єктів, тобто ідентифікувати.

Instagram – це соціальна платформа, яка дозволяє користувачам обмінюватися фотографіями, відео, висловлювати свої думки та відстежувати життя інших людей.

IP-адреса (англ. Internet Protocol address) – це унікальний числовий ідентифікатор, що використовується для адресації комп'ютерів або пристроїв у мережах, побудованих з використанням протоколу TCP/IP.

Messenger – це програмна система для швидкого обміну повідомленнями, створена компанією Facebook.

PyCharm – це програмне середовище для розробки, спеціалізоване на мові програмування Python.

Python – це універсальна високорівнева мова програмування, яка підтримує різні парадигми програмування, зокрема об'єктно-орієнтовану та функціональну.

Telegram – це відома месенджерська платформа, яка дозволяє користувачам обмінюватися повідомленнями, відео, фотографіями та іншими мультимедійними матеріалами.

URL (Uniform Resource Locator – уніфікований локатор ресурсу) – це стандартизована адреса конкретного ресурсу, такого як документ або зображення, у інтернеті або іншому місці.

Viber – це програма для здійснення дзвінків та обміну повідомленнями, яка з'єднує людей незалежно від їх місця знаходження.

ЗМІСТ

ВСТУП.....	9
РОЗДІЛ 1. ТЕОРЕТИЧНІ ВІДОМОСТІ ЧАТ-БОТІВ: ВИВЧЕННЯ КОНЦЕПЦІЙ ТА ПРИНЦИПІВ.....	11
1.1 Чат-бот	11
1.2 Використання чат-ботів	12
1.2.1 Чат-боти для підтримки та обслуговування клієнтів	12
1.2.2 Чат-боти в освіті	14
1.2.3 Чат-боти для SMM та управління соціальними мережами	15
1.3 Переваги та недоліки чат-ботів	16
1.4 Способи створення чат-ботів.....	18
1.5 Висновок до першого розділу.....	19
РОЗДІЛ 2. ІНСТРУМЕНТИ ТА ТЕХНОЛОГІЇ ДЛЯ РОЗРОБКИ ЧАТ-БОТУ	20
2.1 Огляд платформ для створення чат-ботів	20
2.1.1 Telegram	20
2.1.2 Instagram.....	22
2.1.3 Viber	23
2.2 Середовище програмування	25
2.2.1 PyCharm.....	25
2.2.2 Переваги і недоліки PyCharm	26
2.3 Мова програмування	27
2.3.1 Python	27
2.3.2 Переваги і недоліки Python	27
2.4 Telegram API.....	28
2.5 Бібліотека Telebot	29
2.6 Висновок до другого розділу	30
РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ	32
3.1 Опис функціоналу чат-бота	32
3.2 BotFather.....	33
3.3 Головне меню	35

3.4 Шкільний графік.....	37
3.4.1 Розклад дзвінків.....	37
3.4.2 Розклад уроків	38
3.5 Події.....	43
3.6 Погода.....	47
3.7 Адмін-панель.....	50
3.8 Висновок до третього розділу.....	53
РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	55
4.1 Менеджмент безпеки учнів і вчителів	55
4.2 Заходи, що покращують умови праці у школі.....	56
4.3 Психофізіологічне розвантаження для вчителів та учнів	58
4.4 Висновок до четвертого розділу.....	59
ВИСНОВКИ	60
ПЕРЕЛІК ДЖЕРЕЛ	62
ДОДАТКИ	

ВСТУП

Внаслідок глобалізації та розвитку інформаційних технологій, сучасні школи мають змогу покращувати якість освітніх послуг за допомогою інноваційних рішень, таких як чат-боти. Ці автоматизовані помічники здатні значно полегшити комунікацію між школою, учнями та їх батьками, а також забезпечити швидкий доступ до важливої інформації. Тому розробка та впровадження чат-бота «Гончарівський заклад» для навчального закладу є актуальним напрямком досліджень.

Метою даної кваліфікаційної роботи освітнього рівня «Бакалавр» є розробка та впровадження чат-бота для «Гончарівського закладу» з метою підвищення якості освітніх послуг і покращення комунікації між усіма учасниками навчального процесу. Для досягнення поставленої мети потрібно виконати ряд завдань, зокрема:

- Проаналізувати існуючі дослідження та технології у сфері чат-ботів.
- Визначити вимоги та функціональні можливості чат-бота «Гончарівський заклад».
- Розробити архітектуру та функціонал чат-бота.
- Впровадити та протестувати чат-бот у реальних умовах.

Практичне значення одержаних результатів цього проекту полягає у створенні чат-бота для Гончарівської ЗОШ I-II ст. Золочівського району, який значно спростить доступ до важливої інформації для учнів, вчителів та адміністрації школи. Завдяки реалізації головного меню, подій, розкладу дзвінків, розкладу уроків, погоди та адмін-панелі, чат-бот забезпечує багато функцій.

Можливість швидко переглянути розклад дзвінків та уроків дозволить учням та вчителям ефективно планувати свій час. Учні та вчителі завжди будуть у курсі всіх шкільних подій та новин, що відбуваються або відбулися в школі. Інформація про погоду допоможе підготуватися до навчального дня з урахуванням погодних умов. Адміністрація школи або призначені адміністратори зможуть редагувати або надсилати повідомлення через

адмін-панель, що забезпечить оперативне управління та оновлення інформації. Можливість переглядати новини та події через Facebook та шкільний сайт забезпечить зручний доступ до архіву інформації та додаткових ресурсів.

Використання чат-бота «Гончарівський заклад» забезпечить не тільки покращення якості освітніх послуг для поточних учнів і вчителів, але й створить основу для подальших поколінь школярів. Завдяки впровадженню цього інноваційного рішення, нові учні зможуть швидко адаптуватися до шкільного середовища, маючи постійний доступ до актуальної інформації про розклад уроків, події та важливі оголошення. Це значно полегшить процес інтеграції нових учасників навчального процесу і сприятиме створенню комфортного навчального середовища.

Завдяки цьому чат-боту буде забезпечено легкий та швидкий доступ до всієї необхідної інформації, що сприятиме покращенню комунікації та організації навчального процесу в школі. Чат-бот буде розміщено на сервері, що забезпечить його доступність 24/7 для всіх користувачів. Це дозволить будь-якому учню, вчителю або батьку скористатися його функціями в будь-який час і з будь-якого пристрою з доступом до інтернету, забезпечуючи максимальну зручність і оперативність у отриманні необхідної інформації, незалежно від місця перебування користувача.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ВІДОМОСТІ ЧАТ-БОТІВ: ВИВЧЕННЯ КОНЦЕПЦІЙ ТА ПРИНЦИПІВ

1.1 Чат-бот

Чат-бот – це система, де обмін повідомленнями відбувається у форматі «людина-комп'ютер».

Чат-боти – це комп'ютерні програми, побудовані на основі нейронних мереж та методів машинного навчання, які можуть взаємодіяти з користувачами у форматі тексту чи аудіо. Вони призначені для виконання різних завдань, таких як надання інформації або виконання обчислень, а також для розважальних цілей [1].

Чат-боти використовують методи машинного навчання для розробки сценаріїв комунікації. Завдяки постійній взаємодії з людьми вони навчаються імітувати реальні розмови та відповідати на усні та письмові запитання. Такі боти, оскільки вони використовують штучний інтелект, можуть розуміти мову та виконувати команди. Варто зауважити, що окрім ботів, що базуються на штучному інтелекті, існують також боти, які працюють за програмованими сценаріями з кількома варіантами вибору. Наприклад, вибір А може призвести до варіанту Б [2].

У середині двадцятого століття почали розвивати програми-боти. У 1960 роках професор з Массачусетського технологічного інституту, Джозеф Вайзенбаум, створив програму ELIZA, яка послужила прототипом для сучасних спам-детекторів. Перший електронний помічник, названий Julia на честь покійної доньки винахідника, був розроблений американським програмістом Майклом Молднігом у середині 90-х років. Julia могла аналізувати тексти, що написав Майкл. У період поширення Інтернету в нульові роки боти стали популярними для онлайн-вікторин, модерації форумів і цензури чатів.

Пізніше боти перетворилися на персональних голосових помічників. Siri з'явилася в Apple, а Cortana в Microsoft. У 2014 році Amazon інтегрував свого

помічника Alexa у розумні колонки Amazon Echo, а в 2016 році Google запустив Google Assistant.

У 2017 році WhatsApp мав понад 1 мільярд щоденних користувачів. Цей швидкозростаючий новий канал зв'язку привернув увагу маркетологів, підприємців та інвесторів. Виявляється, аудиторія активно переходить від соціальних мереж і електронної пошти до систем обміну миттєвими повідомленнями. Цей ідеальний консультант може працювати цілодобово, 365 днів на рік, для підтримки найвищого рівня обслуговування, не залежно від погодних умов або настрою. Залишилося лише навчити клієнтів користуватися чат-ботом [3].

1.2 Використання чат-ботів

Чат-боти стали невід'ємною частиною різноманітних сфер і відіграють ключову роль в обслуговуванні клієнтів. Вони допомагають вирішувати проблеми та відповідати на запитання, а також ефективно взаємодіють з потенційними клієнтами в маркетингу і продажах. Чат-боти можуть також служити як навчальні посібники та проводити консультації в реальному часі. Вони допомагають автоматизувати рутинні завдання, що значно зекономлює час як для користувачів, так і для компаній.

Чат-боти стали популярними в месенджерах, таких як Viber, Telegram, WhatsApp і Facebook Messenger, і отримали велику популярність на фоні соціальних мереж [4].

1.2.1 Чат-боти для підтримки та обслуговування клієнтів

При розробці логіки та сценаріїв для чат-ботів, компанії мають можливість детально опрацювати всі потенційні запитання клієнтів. Класифікуючи часті запитання за різними категоріями, користувачі можуть швидко знаходити рішення своїх проблем без потреби в залученні адміністратора. Швидке вирішення проблем підвищує клієнтську лояльність.

Чат-боти значно економлять час як користувачів, так і операторів. Вони надають консультації та задовольняють основні потреби клієнтів, дозволяючи операторам зосередитися на більш складних і нестандартних завданнях. Крім того, чат-боти доступні цілодобово і можуть обробляти необмежену кількість запитів одночасно [5].

Гнучкість у швидкому редагуванні змісту чи функцій чат-бота гарантує, що клієнти завжди матимуть доступ до актуальної інформації. Наприклад, під час надзвичайних ситуацій інтернет-провайдер може оперативним сповістити всіх клієнтів про ремонтні роботи через чат-бота. Без чат-ботів компанії часто стикаються з раптовим збільшенням навантаження на свої контакт-центри [6].

Одним із прикладів чат-бота для підтримки клієнтів є Зоряна, віртуальний консультант від компанії Київстар. Якщо клієнт хоче змінити свій тарифний план, він може звернутися до неї, і вона надасть інструкції. Якщо клієнт зацікавлений в поточних акціях, Зоряна підкаже, де можна отримати знижки. У разі проблем з Інтернетом вона допоможе знайти рішення. Загальний вигляд чат-бота Зоряна показано на рисунку 1.1.

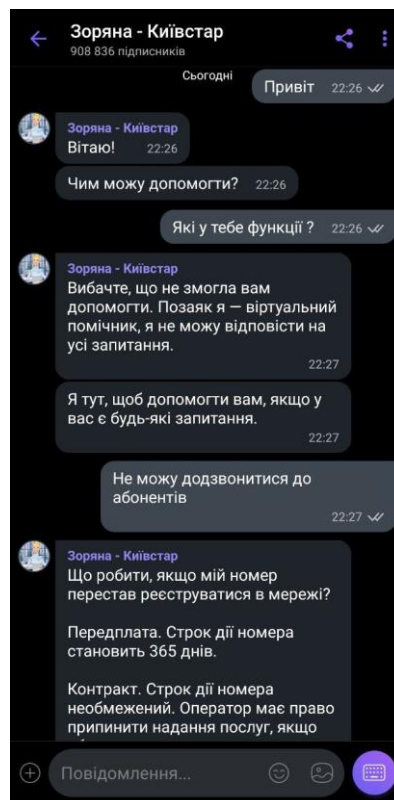


Рисунок 1.1 – Чат-бот Зоряна

Чат-боти здатні миттєво відповідати на стандартні запитання та ефективно вирішувати типові проблеми. У випадку, коли чат-бот служби підтримки не може розв'язати проблему, він автоматично перенаправляє запит до менеджера, забезпечуючи швидке вирішення складних ситуацій [4].

1.2.2 Чат-боти в освіті

Месенджер став невід'ємною частиною життя багатьох людей, які використовують його для спілкування, обміну мультимедійними файлами та вирішення важливих питань. У зв'язку з цим розмовний формат подачі навчальних матеріалів набув особливого значення завдяки швидкій передачі інформації, миттєвому зворотному зв'язку та цілодобовій підтримці.

Чат-боти активно використовують у маркетингу, банківській справі, комп'ютерних іграх та освіті. Наприклад, англomовний чат-бот може допомогти покращити навички спілкування іноземною мовою. Чат-бот Repeat Word Bot показано на рисунку 1.2.

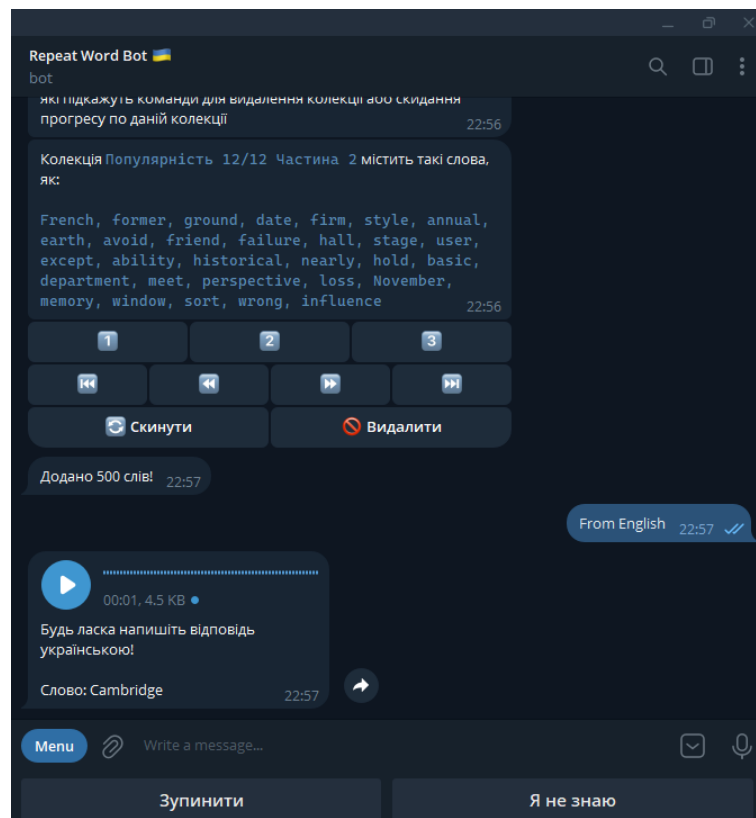


Рисунок 1.2 – Чат-бот Repeat Word Bot

Індивідуальна підтримка надається відповідно до рівня та темпу навчання кожного учня, забезпечуючи доступність освіти для будь-кого, хто має мобільний пристрій і доступ до Інтернету. Чат-боти також допомагають вчителям впроваджувати формувальне оцінювання за рахунок швидкого зворотного зв'язку та індивідуальних траєкторій навчання для кожного учня [7].

1.2.3 Чат-боти для SMM та управління соціальними мережами

Телеграм часто обирають SMM-фахівці для роботи з соціальними медіа. Цей месенджер не лише дозволяє спілкуватися, але й стає корисним інструментом для аналізу та підтримки аккаунтів в Instagram. Чат-бот Instagram Saver показано на рисунку 1.3.

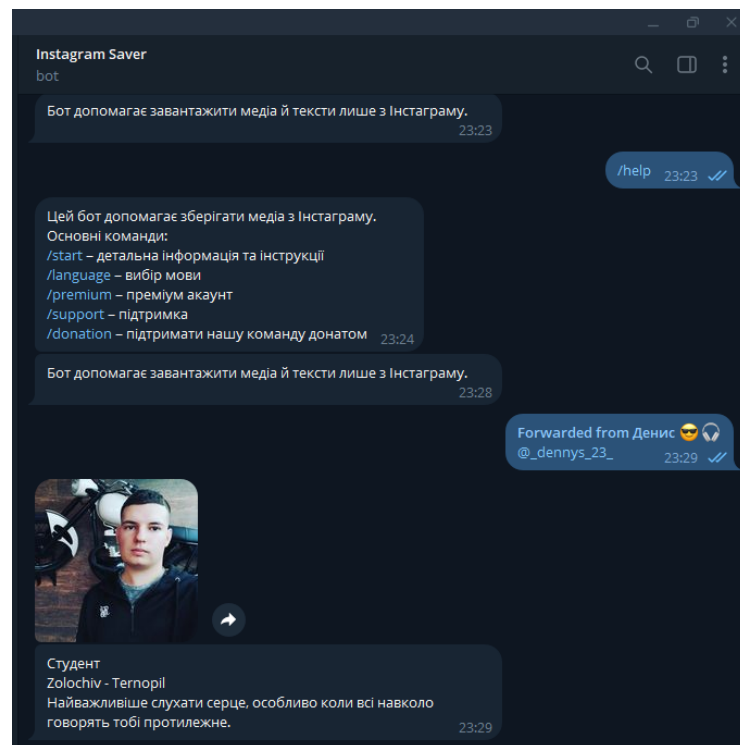


Рисунок 1.3 – Чат-бот Instagram Saver

Також телеграм-боти можуть створювати UTM-мітки для відстеження рекламних кампаній, що допомагає в аналізі ефективності рекламних стратегій. Ці боти є важливими асистентами для digital-фахівців, що полегшують їхню роботу та підвищують продуктивність [8].

1.3 Переваги та недоліки чат-ботів

Чат-боти відкривають широкий спектр можливостей для покращення взаємодії з користувачами та оптимізації бізнес-процесів. Їхня постійна доступність без обмежень у часі дозволяє користувачам отримувати необхідну інформацію та підтримку у будь-який момент. Це забезпечує підвищення лояльності клієнтів та покращення їхнього досвіду. Оперативність та ефективність чат-ботів дозволяють розглядати їх як надійного помічника для вирішення рутинних завдань та оптимізації часу як для користувачів, так і для співробітників [9].

Однією з ключових переваг чат-ботів є зменшення навантаження на співробітників. Вони можуть взяти на себе рутинні процеси обслуговування клієнтів та аналізу даних, дозволяючи співробітникам зосередитися на більш важливих завданнях. Це сприяє підвищенню ефективності роботи компанії та покращенню якості обслуговування клієнтів. Наприклад, чат-боти можуть автоматично обробляти запити щодо наявності товару, відстеження відправлень, уточнення адрес або прийняття показників, що значно економить час і ресурси.

Крім того, інтеграція чат-ботів з іншими системами підтримки та аналітики дозволяє компаніям зберігати історію взаємодії з клієнтами та отримувати цінні аналітичні дані для подальшого вдосконалення послуг та стратегій. Такий підхід сприяє постійному розвитку та удосконаленню чат-ботів, що робить їх надійним інструментом у сфері підтримки клієнтів та маркетингу. Збір даних про клієнтів та персоналізовані пропозиції дозволяють підвищити ефективність маркетингових кампаній і покращити клієнтський досвід.

Чат-боти також можуть допомагати в автоматизації процесів навчання та підтримки співробітників. Вони здатні швидко надавати відповіді на питання та розповсюджувати навчальні матеріали, що підвищує швидкість навчання нових працівників. Чат-боти можуть використовуватися для проведення опитувань та збору зворотного зв'язку від клієнтів, що допомагає компаніям краще розуміти потреби та очікування своєї аудиторії [10].

Однак, попри всі ці переваги, чат-боти мають і певні недоліки, які варто враховувати. Для багатьох користувачів Messenger залишається лише засобом спілкування. Використання його для замовлень є новим досвідом для багатьох людей, особливо для старшого покоління, яке може не бути знайомим з його функціональністю. Новизна і невідомість таких технологій можуть створювати бар'єри для користування, оскільки люди, які не звикли до подібних сервісів, можуть відчувати дискомфорт і невпевненість у їх застосуванні. Це може спричинити відмову від використання чат-ботів, навіть якщо вони здатні значно полегшити виконання певних завдань.

Часто користувачі ставляться до чат-ботів з недовірою. Вони вважають їх зайвим етапом, після якого все одно доведеться звертатися до адміністратора і повторювати інформацію. Іноді користувачі можуть недооцінювати можливості чат-ботів, вважаючи, що вони не зможуть вирішити їхні питання належним чином. Це може призводити до ситуацій, коли користувачі оминають бота і одразу звертаються до живого консультанта, навіть не намагаючись скористатися автоматизованою системою. Така недовіра до чат-ботів часто виникає через негативний досвід або через недостатнє ознайомлення з їхніми можливостями.

Деякі чат-боти не забезпечують переклад тексту, а їх інтерфейс може бути надто простим або незрозумілим. Це створює труднощі для користувачів, які не знають, які дії очікує від них бот, і можуть відчувати розгубленість. Наприклад, якщо бот використовує складні команди або не має чітких інструкцій, користувач може не розуміти, як правильно взаємодіяти з системою, що значно знижує ефективність використання чат-бота. Невдалий інтерфейс може викликати фрустрацію, що призводить до негативного досвіду і відмови від використання чат-бота в майбутньому [11].

Незалежно від того, наскільки розумним є бот, його можливості обмежені. Зазвичай він може надавати базову консультацію, інформацію про наявність товару, допомагати з відстеженням відправлень, уточнювати адреси або приймати показники. Однак він не може вирішити складніші питання або надати глибоку допомогу. Наприклад, якщо користувач має специфічну проблему, яка

вимагає індивідуального підходу або творчого мислення, бот навряд чи зможе запропонувати адекватне рішення. В таких випадках, користувачі можуть відчувати розчарування через обмеженість можливостей чат-бота.

Чат-боти, які використовуються для підтримки, можуть надавати лише загальні рекомендації або допомагати зі стандартними процедурами. Вони не можуть детально вникнути в суть проблеми, що часто викликає роздратування у клієнтів, які потребують негайної допомоги. Це може вплинути на загальний імідж компанії і знизити рівень задоволеності клієнтів.

Таким чином, при всій своїй зручності, чат-боти мають низку обмежень, які варто враховувати при їх використанні. Важливо розуміти ці обмеження і бути готовими до того, що не завжди чат-бот зможе повністю задовольнити потреби користувачів. Попри це, чат-боти залишаються корисним інструментом для автоматизації процесів і покращення обслуговування, якщо використовувати їх правильно і враховувати всі можливі недоліки [12].

1.4 Способи створення чат-ботів

У сфері чат-ботів можна відокремити два основних способи розробки. Перший спосіб, який включає в себе розробку чат-бота з нуля, передбачає формування команди програмістів та фахівців з різних областей, таких як дизайн інтерфейсу, штучний інтелект, тестування та аналіз даних. Цей підхід, як правило, потребує значних витрат часу та фінансових ресурсів, оскільки вимагає розробки складних алгоритмів та інтеграції з системами управління базами даних. Проте він дозволяє створити чат-бота, який повністю відповідає унікальним потребам та вимогам бізнесу, що може бути ідеальним рішенням для великих корпорацій або підприємств зі складними бізнес-процесами.

З іншого боку, використання конструкторів чат-ботів є більш простим та доступним способом розробки автоматизованих агентів. Ці інструменти зазвичай мають інтуїтивний інтерфейс та широкий набір функцій, що дозволяє створювати базові чат-боти без необхідності в програмістських навичках. Їх використовують малі підприємства або стартапи, що шукають швидке та просте

рішення для полегшення комунікації з клієнтами або автоматизації певних бізнес-процесів [13].

Обидва ці підходи мають свої переваги та недоліки, і вибір між ними повинен бути здійснений з урахуванням конкретних потреб та можливостей вашого бізнесу. Важливо ретельно проаналізувати всі підходи та визначити, який метод краще відповідає вашим потребам та цілям [14].

1.5 Висновок до першого розділу

У першому розділі кваліфікаційної роботи досліджено різноманітні підходи використання чат-ботів у сучасному світі. Аналіз цих підходів дозволяє зробити висновок про значущість цієї технології в різних сферах. Використання чат-ботів у сфері обслуговування клієнтів, освіти та управлінні соціальними мережами підтверджує глибокий вплив цієї інновації на різні погляди повсякденного життя.

Разом із перевагами чат-ботів, такими як підвищення ефективності та зручності, виникають виклики, такі як необхідність постійного удосконалення та проблеми зі сприйняттям природної мови. Ці підходи потребують подальшого дослідження та вдосконалення для максимального використання потенціалу чат-ботів.

Розділ також розкриває різні методи створення чат-ботів, що створює безмежні можливості для інновацій та творчості у цій сфері. Відкритий простір для експериментів та вдосконалення дозволяє постійно розвивати цю технологію та розширювати її застосування в різних галузях.

Загалом, на основі розділу можна зробити висновок про важливість та перспективність подальшого дослідження та використання чат-ботів у різних галузях. Попри обмеження, чат-боти залишаються корисним інструментом для автоматизації процесів та покращення обслуговування, якщо їх використовувати з урахуванням всіх можливих недоліків.

РОЗДІЛ 2. ІНСТРУМЕНТИ ТА ТЕХНОЛОГІЇ ДЛЯ РОЗРОБКИ ЧАТ-БОТУ

2.1 Огляд платформ для створення чат-ботів

Сучасні технології дозволяють створювати чат-ботів у різних популярних месенджерах, таких як Discord, WhatsApp, Viber, Instagram, Telegram та Facebook Messenger. У цій роботі обрано месенджер Telegram для розробки чат-бота. Проте далі будуть детально розглянуті декілька найпопулярніших платформ.

2.1.1 Telegram

Чат-бот Telegram – це спеціальний обліковий запис, який автоматично відповідає на повідомлення користувачів. Цей інструмент дозволяє бізнесам ефективніше взаємодіяти з клієнтами, забезпечуючи постійну доступність і миттєву відповідь у будь-який час без додаткових витрат на зарплату [15].

BotFather – популярний чат-бот Telegram, є ідеальним інструментом для тих, хто хоче створити свого власного бота. Зовнішній вигляд чат-бота показано на рисунку 2.1.

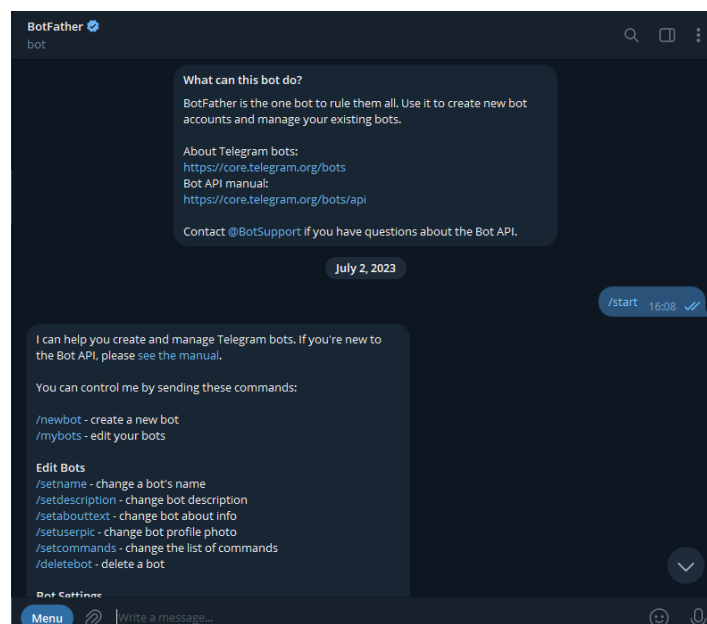


Рисунок 2.1 – BotFather

Він пропонує безліч можливостей:

1. Створення бота. BotFather дозволяє створити нового бота з унікальним іменем та ID, а також вибрати тип бота – від бота для обміну повідомленнями до бота для опитувань.

2. Налаштування бота. Після створення бота можна налаштувати його функції та параметри, включаючи автовідповідачі, ключі API та права доступу.

3. Статистика використання. Користувачі можуть отримувати статистику про використання свого бота, зокрема кількість надісланих повідомлень та активність користувачів.

4. Інтеграція з іншими сервісами. BotFather підтримує інтеграцію з іншими службами, що дозволяє розширити функціональність бота.

5. Створення кнопок і меню. Також BotFather дозволяє створювати кнопки та меню для зручної взаємодії з ботом.

6. Відправка повідомлень. BotFather дозволяє відправляти інформаційні бюлетені для передплатників бота.

7. Підтримка різних мов. BotFather підтримує кілька мов, що дозволяє створювати ботів для різних аудиторій.

8. Відкрите API. BotFather має відкрите API, що дозволяє розробникам створювати нові функції та інтегрувати їх з іншими службами [16].

Ринок чат-ботів значно зріс на 92% за останні кілька років, і вони стали найшвидшим каналом комунікації для брендів. Більшість користувачів мають нейтральний або позитивний досвід спілкування з чат-ботами. Серед B2B-компаній 58% використовують чат-боти, тоді як у B2C-компаніях цей показник становить 42%. Використання чат-ботів Telegram дозволяє отримувати до 90% відгуків від найбільш зацікавлених клієнтів, при цьому середній рівень відповідей сягає 35-40%. Крім того, більшість користувачів надають перевагу чат-ботам через їх здатність швидко надавати відповіді [17].

2.1.2 Instagram

Чат-боти в Instagram представляють собою інструмент для автоматизованого спілкування у цій соціальній мережі. Вони використовують штучний інтелект, щоб автоматично реагувати на повідомлення користувачів.

Ці боти можуть відправляти автоматичні відповіді тим, хто коментує записи або починає розмову в Direct, як показано на рисунку 2.2. Вони доступні цілодобово і можуть вести багато розмов одночасно.

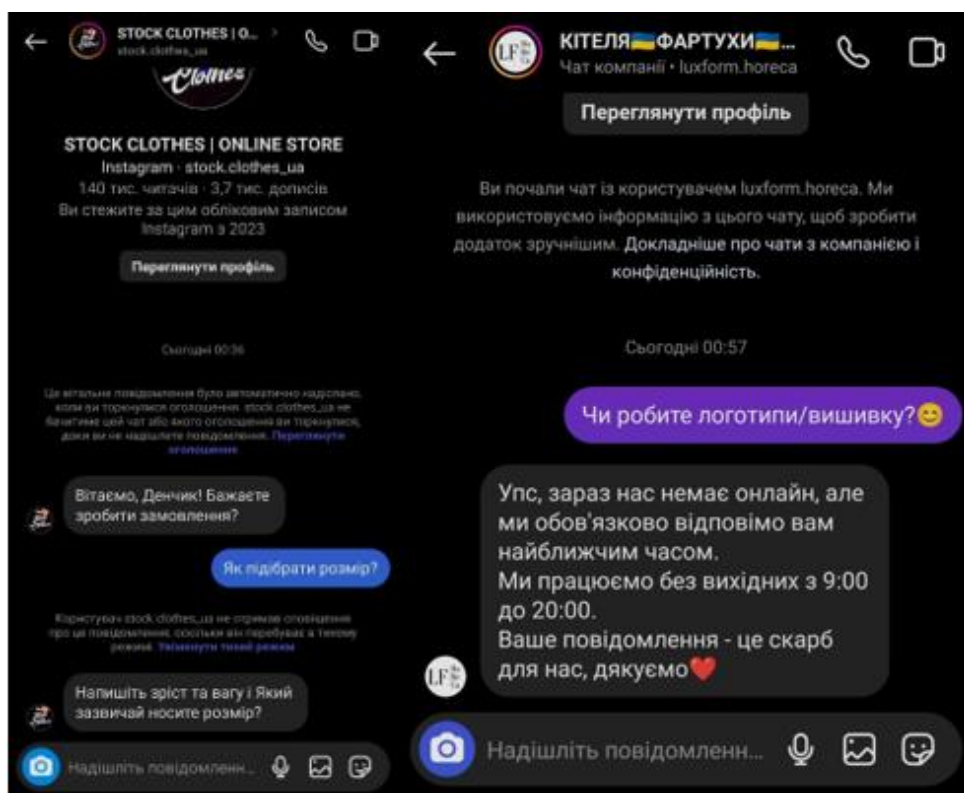


Рисунок 2.2 – Приклади відповідей чат-бота для Instagram

Створення чат-бота в Instagram можливе для бізнес-акаунтів будь-якої кількості підписників або для облікових записів авторів контенту з мінімум 500 000 підписників.

Робота чат-бота в Instagram базується на використанні шаблонів розмов, що дозволяє ботам реагувати на користувачів залежно від їхніх дій.

Ці боти можуть надсилати повідомлення користувачам після підписки на сторінку, початку розмови в Direct або відповіді на певні тригерні слова. Приклади таких ботів з тригерними словами показано на рисунку 2.3.

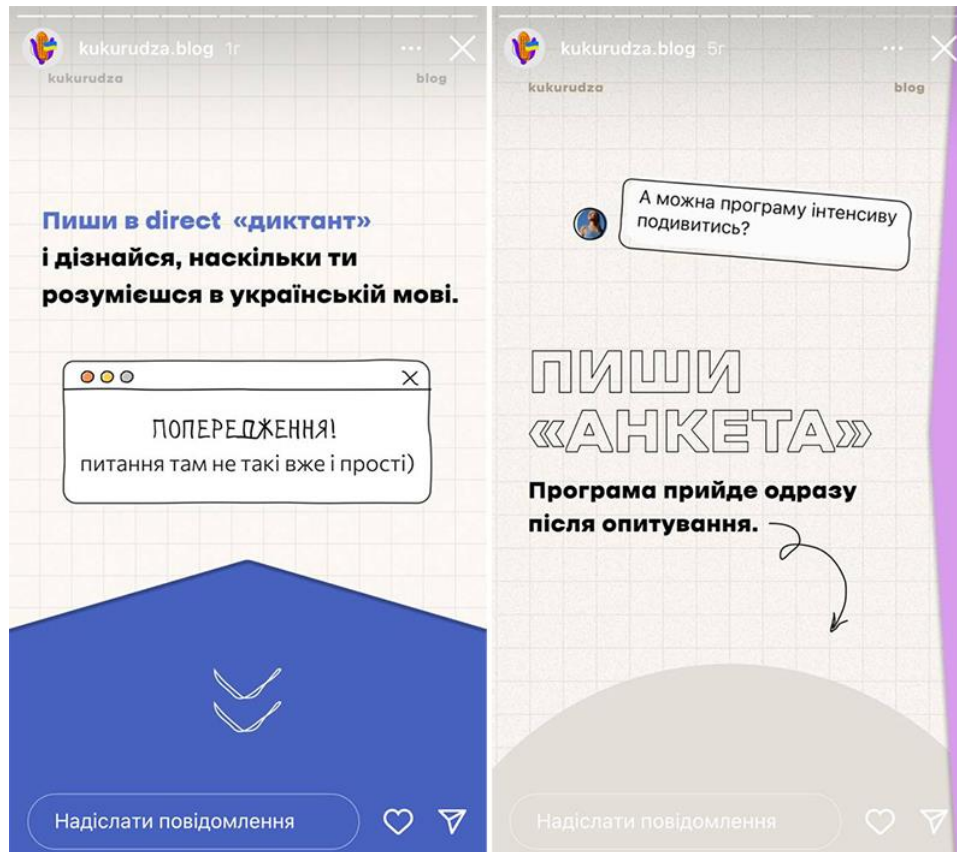


Рисунок 2.3 – Приклад слова-тригера для активації автоматичних відповідей у Direct

Чат-боти Instagram виявляються ефективними для здійснення покупок через цю платформу. Вони допомагають брендам перетворити інтерес користувачів у реальні покупки, відповідаючи на їхні запитання та надсилаючи інформацію через Direct [18].

2.1.3 Viber

Чат-бот Viber – це інструмент, що імітує роботу людини, забезпечуючи підтримку клієнтів або надаючи необхідну інформацію. Бот може бути використаний для комерційних цілей, спрощуючи взаємодію з користувачами.

Реалізується чат-бот через публічні облікові записи, які можуть керувати як люди, так і боти. Приклад Viber чат-бота Укрпошта показано на рисунку 2.4.

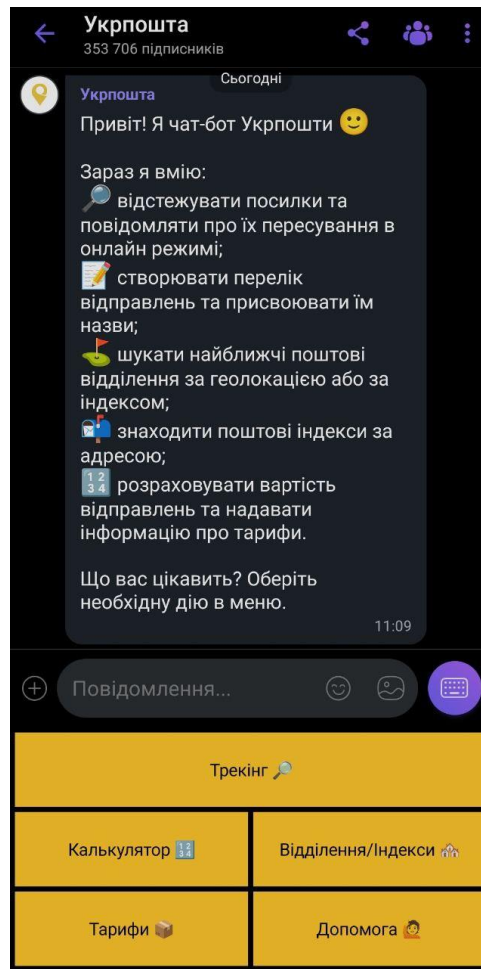


Рисунок 2.4 – Viber чат-бот Укрпошта

Чат-бот Viber має різноманітні функції, включаючи надсилання рахунків-фактур, здійснення платежів, поштові розсилки та надсилання повідомлень у форматі каруселі. Він може працювати цілодобово, що дає можливість клієнтам отримувати швидку підтримку в будь-який час [19].

Для бізнесу використання чат-бота Viber має декілька переваг. Наприклад, він може значно полегшити обробку запитів клієнтів, особливо якщо їх багато або вони схожі між собою. Крім того, завдяки автоматизації, бот може працювати безперервно, звільняючи людські ресурси для вирішення більш складних питань.

Однією з головних переваг є можливість розширення команди підтримки за рахунок чат-бота. Він може взаємодіяти з клієнтами, виконуючи стандартні завдання, тим самим зменшуючи навантаження на операторів.

Чат-бот Viber також може бути використаний для кваліфікації потенційних клієнтів та попереднього нагріву холодних лідів. Він може ставити питання користувачам, оцінюючи їхній інтерес у покупці, і збирати контактну інформацію для подальшого зв'язку. Також чат-бот Viber може збирати відгуки клієнтів у реальному часі, що дозволяє компанії швидко реагувати на негативні відгуки та покращувати якість обслуговування.

У підсумку, використання чат-бота Viber може підвищити ефективність обслуговування клієнтів, зменшити витрати і полегшити взаємодію з користувачами, що робить його корисним інструментом для бізнесу [20].

2.2 Середовище програмування

Середовищем програмування для цієї роботи було обрано PyCharm. Причини, переваги та недоліки цього вибору наведені нижче.

2.2.1 PyCharm

PyCharm, випущений компанією JetBrains, представляє собою високопродуктивне та універсальне середовище розробки для мови програмування Python. Він володіє розширеним набором інструментів, спрямованих на полегшення процесу розробки та налагодження програмного коду.

Це програмне забезпечення є ідеальним вибором для розробників Python будь-якого рівня – від початківців до досвідчених фахівців. Воно забезпечує швидку та зручну роботу з проектами, автозавершення коду, перевірку синтаксису, налагодження та багато іншого [21].

Навіть поза областю Python, PyCharm є потужним інструментом для веб-розробки. Його інтеграція з фреймворками, такими як Django і Flask, робить його

ідеальним вибором для створення веб-застосунків. Крім того, він має вбудований клієнт баз даних, який спрощує роботу з SQL і ORM, а також надає інструменти для тестування та налагодження веб-застосунків [22].

2.2.2 Переваги і недоліки PyCharm

У сфері розробки на Python, PyCharm часто визначається як оптимальний вибір. Втім, навіть з усіма його перевагами, важливо пам'ятати про його обмеження. Ось деякі з переваг та недоліків використання PyCharm, які варто розглянути.

Переваги:

- Простота перегляду вихідного коду Python за допомогою одного клацання миші.
- Широкий вибір плагінів, що розширюють можливості IDE.
- Легкість в використанні, що дозволяє швидше розробляти код.
- Наявність більш потужної комерційної версії для продвинутих користувачів.
- Простий процес встановлення, що зменшує час настройки.

Недоліки:

- Наявність комерційної версії може відповідати потребам лише деяких користувачів.
- Можливі труднощі з інтеграцією деяких інструментів, таких як `venv`.
- Не найкращий вибір для початківців, оскільки вимагає певного рівня експертизи.
- Високі вимоги до ресурсів комп'ютера, таких як пам'ять і диск, що може призводити до значного споживання ресурсів.

PyCharm вважається одним з найефективніших та потужних інструментів для розробки на Python, він має свої обмеження, які важливо враховувати при виборі IDE для проектів [23].

2.3 Мова програмування

Мовою програмування для цієї роботи було обрано Python. Причини, переваги та недоліки цього вибору наведені нижче.

2.3.1 Python

Мова програмування Python, друга за популярністю у світі, широко використовується для різних цілей, включаючи розробку веб-додатків, програмного забезпечення та машинного навчання. Завдяки своїй простоті вивчення, ефективності і кросплатформенності, вона знайшла застосування у таких відомих компаніях, як Google, Instagram, Facebook, IBM, NASA, Dropbox, Netflix та інші.

Python використовується для різних завдань, включаючи розробку програмного забезпечення у різних областях, серверну частину мобільних застосунків, створення комп'ютерних ігор (наприклад, World of Tanks), вбудовані системи для пристроїв, написання скриптів та плагінів для автоматизації процесів, тестування, а також машинне навчання.

Для виконання різних завдань використовуються різні бібліотеки Python, такі як Matplotlib для графіки, Pandas для операцій з даними, NumPy для роботи з масивами, а також фреймворки, що полегшують розробку, такі як Django, Піраміда та Web2py [24].

2.3.2 Переваги і недоліки Python

Ось деякі з переваг та недоліків використання PyCharm, які варто розглянути.

Переваги:

– Python пропонує сучасний синтаксис та підтримку різних парадигм програмування, таких як об'єктно-орієнтоване, функціональне та реактивне програмування.

– У світі Python існує активна та дружня спільнота користувачів, яка завжди готова допомогти і відповісти на запитання, навіть на таких платформах, як StackOverflow.

– Python має величезну бібліотеку стандартних пакетів, яка включає у себе потужні інструменти для різних завдань, від обробки зображень до машинного навчання.

– Програми, написані на Python, можуть запускатися на будь-якій платформі, що підтримує інтерпретатор Python, забезпечуючи переносимість та гнучкість розробки.

– Python широко використовується для створення прототипів завдяки своєму стислому синтаксису та розширеній бібліотеці стандартних пакетів, що спрощує швидке розроблення та перевірку концепцій.

– Велика кількість матеріалів та документації, доступних онлайн, сприяє швидкому вивченню та розвитку навичок у програмуванні на Python.

Недоліки:

– Складність у публікації пакетів з відкритим кодом.

– Відсутність явної типізації, що може вимагати більшої уваги до тестування.

– Складнощі з управлінням залежностями та створенням образів програм для контейнеризації [25].

2.4 Telegram API

Telegram API – це засіб для розробників, який надає набір інструментів та інтерфейсів програмування. Цей інтерфейс дозволяє розробникам створювати різноманітні програми, ботів та сервіси, які взаємодіють з програмою Telegram та використовують її функції. Він визначає правила і протоколи для спілкування різних програм та сервісів між собою [26]. Наприклад, за допомогою API Telegram розробники можуть здійснювати надсилання повідомлень, управляти контактами, створювати групи або канали, а також працювати з медіафайлами [27].

Telegram API відкриває перед розробниками широкі можливості для створення різноманітних програм та сервісів. Наприклад:

- Розробники можуть створювати ботів, які автоматично взаємодіють з користувачами, надсилаючи повідомлення та виконуючи різні завдання.
- Також вони можуть створювати власні програми, які інтегруються з функціоналом Telegram, що може бути, наприклад, соціальною мережею чи сервісом обміну медіа.
- API Telegram дозволяє інтегрувати його з іншими зовнішніми сервісами, наприклад, для автоматичної відправки повідомлень за допомогою RSS-стрічок чи CRM-систем.

Для початку використання API Telegram потрібно отримати API-ключ, який ідентифікує вашу програму чи бота. За його допомогою відбувається безпечна взаємодія з Telegram API. Після цього можна надсилати HTTP-запити до відповідних кінцевих точок для виконання різних операцій, таких як надсилання повідомлень чи створення груп. Для складніших операцій доступні спеціальні бібліотеки та SDK для різних мов програмування, які спрощують роботу з Telegram API [28].

2.5 Бібліотека Telebot

Бібліотека Telebot, також відома як `pyTelegramBotAPI`, є одним із найпопулярніших інструментів для створення Telegram-ботів на Python. Вона надає зручний інтерфейс для взаємодії з Telegram Bot API, що дозволяє розробникам швидко створювати та налаштовувати ботів для різноманітних завдань[29]. Основні концепції та теоретичні основи роботи з telebot:

1. Telegram Bot API – це набір HTTP-методів, які дозволяють взаємодіяти з серверами Telegram для виконання різних дій, таких як відправлення та отримання повідомлень, управління медіа, створення клавіатур тощо. Бібліотека telebot забезпечує зручну обгортку для цих методів, спрощуючи процес розробки ботів.

2. Основні компоненти telebot:

- Об'єкт TeleBot є центральним елементом бібліотеки, відповідальним за взаємодію з Telegram Bot API та управління всіма підходами роботи бота.

- Обробники повідомлень дозволяють визначати, як бот реагуватиме на різні типи повідомлень. Вони налаштовуються за допомогою декораторів.

3. Типи обробників:

- Command Handlers – для обробки команд (наприклад, /start, /help).

- Message Handlers – для обробки текстових повідомлень.

- Callback Query Handlers – для обробки запитів від inline-кнопок.

4. Взаємодія з користувачами:

- Бот може відправляти текстові повідомлення, використовуючи метод `send_message`.

- Бібліотека підтримує відправлення різних типів медіа, таких як фото, відео, документи.

Бібліотека Telebot спрощує розробку Telegram-ботів на Python, забезпечуючи зручний інтерфейс для взаємодії з Telegram Bot API. Використовуючи об'єкт TeleBot та різні типи обробників повідомлень, можна швидко створювати функціональних ботів для різних завдань, від простого відправлення повідомлень до складних взаємодій з користувачами та медіа [30].

2.6 Висновок до другого розділу

У другому розділі кваліфікаційної роботи було розглянуто ключові підходи, необхідні для розробки чат-ботів, зокрема платформи для створення ботів, середовище програмування, мову програмування, Telegram API та бібліотеку Telebot. Ці компоненти є основними складовими для створення функціонального та ефективного чат-бота.

Платформи для створення чат-ботів, такі як Telegram, Instagram та Viber, мають свої унікальні особливості та переваги. Telegram виділяється своєю популярністю та багатим функціоналом, що включає просту інтеграцію та розширені можливості для налаштування ботів. Instagram, хоча і має деякі обмеження API, залишається привабливим для автоматизації маркетингових

завдань та підтримки клієнтів завдяки великій аудиторії користувачів. Viber, у свою чергу, є популярним в певних регіонах і пропонує інтуїтивно зрозумілі інструменти для створення ботів, що можуть бути ефективними у підтримці клієнтів та маркетингових кампаніях.

Вибір середовища програмування також є критичним для ефективної розробки чат-ботів. У роботі було розглянуто PyCharm, потужне інтегроване середовище розробки, яке підтримує мову програмування Python. PyCharm забезпечує високу продуктивність завдяки таким інструментам, як автодоповнення коду, інтеграція з системами контролю версій та тестування, що робить процес розробки більш зручним та швидким.

Python була обрана як основна мова програмування завдяки своїй простоті та читабельності коду, а також широкому спектру бібліотек та фреймворків, які значно спрощують розробку чат-ботів. Python має активну спільноту розробників та добре документовані ресурси, що допомагає швидко знаходити рішення для різних завдань.

Розглянувши Telegram API, було відзначено його потужність та гнучкість у наданні доступу до функцій Telegram, що дозволяє надсилати та отримувати повідомлення, керувати групами та каналами. Бібліотека Telebot, в свою чергу, забезпечує зручний інтерфейс для взаємодії з Telegram API, що значно полегшує процес створення та налаштування ботів, обробку повідомлень та команд.

Загалом розділ надав всебічний огляд основних інструментів та технологій для розробки ефективних чат-ботів. Платформи для створення ботів, середовище програмування PyCharm, мова програмування Python, Telegram API та бібліотека Telebot забезпечують всі необхідні можливості для створення чат-ботів з багатим функціоналом, що дозволяє розробникам зосередитися на реалізації інноваційних рішень та забезпеченні високої якості обслуговування користувачів.

РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ

3.1 Опис функціоналу чат-бота

Розробка чат-бота включає декілька ключових етапів, які визначають його функціонал та можливості.

Починаючи з першого етапу, чат-бот зосереджується на привітанні користувача. Це важливий момент, оскільки перше враження може визначити подальшу взаємодію. Тому чат-бот вітає користувача та коротко вказує, з якою метою він створений.

Далі надається можливість перейти до головного меню. Це місце, де користувач може обрати те, що його цікавить або яку функцію він бажає використовувати.

Головне меню включає такі пункти:

1. Розклад уроків. Цей пункт є надзвичайно важливим для учнів та вчителів, адже дозволяє всім бути в курсі актуального розкладу. Він може бути адаптований до різних класів та груп, забезпечуючи інформацію про час та місце проведення кожного уроку.

2. Розклад дзвінків. Точний розклад дзвінків допомагає утримувати порядок протягом шкільного дня. Він визначає, коли саме учні навчаються, коли у них є час на відпочинок, та коли настає час обідньої перерви.

3. Погода. Інформація про погоду є корисною для планування шкільних заходів на відкритому повітрі, таких як спортивні ігри чи екскурсії. Вона також допомагає учням вибрати відповідний одяг на день.

4. Події. Пункт подій відображає у собі посилання на фейсбук і сайт школи, а також новини, які зібрані із сайту. У цьому пункті є інформація про всі важливі шкільні події, які включають святкування, збори, театральні вистави та багато іншого. Це допомагає учням та батькам планувати свій час та бути в курсі шкільного життя.

Також існує ще один підпункт – адмін-панель, який доступний тільки адміністраторові, який має можливість керувати розкладом уроків, що дає їм

гнучкість у випадку необхідності змін. Вони також можуть використовувати цю панель для надсилання важливих повідомлень всім користувачам, що є ефективним способом комунікації.

На рисунку 3.1 показані складові головного меню чат-бота.



Рисунок 3.1 – Складові головного меню чат-бота

Коли вже визначені складові головного меню та основний задум розробки, можна приступати до реалізації задуманих ідей. На цьому етапі важливо дотримуватися початкового плану, регулярно тестувати функціональність, а також вносити необхідні корективи для покращення чат-боту.

3.2 BotFather

Щоб створити чат-бота, його спершу необхідно зареєструвати за допомогою BotFather. BotFather – це Telegram-бот, призначений для реєстрації нових чат-ботів і управління ними, створених користувачами.

Використовуючи команду /newbot, створюється новий бот. @BotFather запитує ім'я користувача, а потім генерує маркер автентифікації для нового бота, як показано на рисунку 3.2.

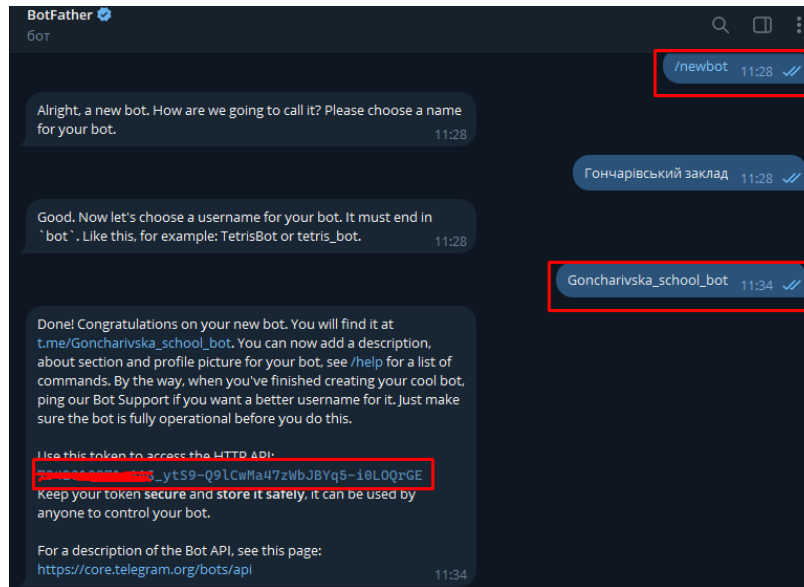


Рисунок 3.2 – Створення чат-бота у Telegram

Також токен, який нам дав BotFather дозволяє створювати і керувати Telegram-ботами. Токен є унікальним ідентифікатором, який дозволяє боту з'єднатися з Telegram API та виконувати функції, які були налаштовані через BotFather. Таким чином, бот може отримувати повідомлення від користувачів, відправляти їм відповіді та взаємодіяти з ними в Telegram.

```
bot = telebot.TeleBot('7342010371:AAG_ytS9-Q9lCwMa47zWbJBYq5-i0LOQrGE')
```

В BotFather на рисунку 3.3 було додано опис та аватарку, що розповідають про те, що робить чат-бот.

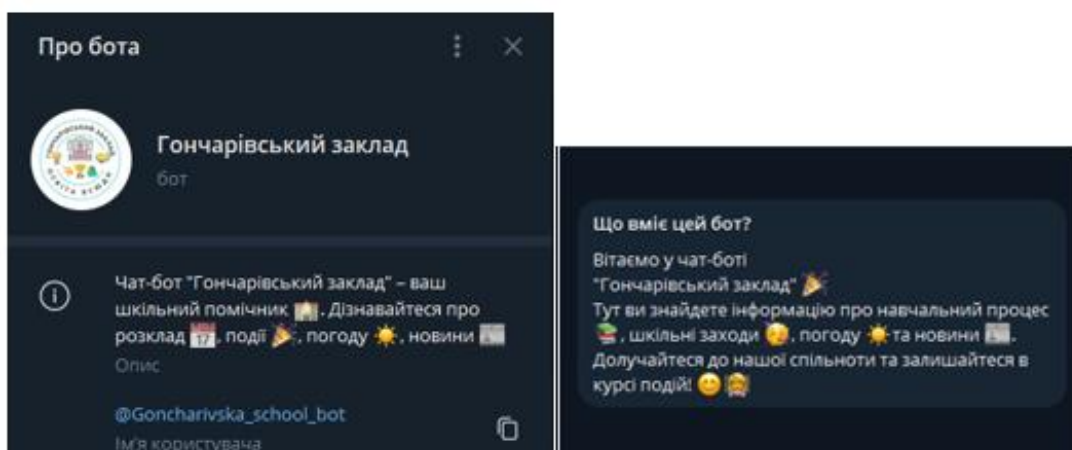


Рисунок 3.3 – Аватарка і опис чат-боту

Опис є важливою складовою презентації чат-бота, оскільки він стисло пояснює користувачам його функціонал і переваги. Завдяки добре написаному опису, нові користувачі можуть одразу зрозуміти, як саме чат-бот є корисним для них. Аватарка додає візуальну привабливість та полегшує ідентифікацію бота серед інших контактів.

3.3 Головне меню

При вході користувача до чат-бота, він вітається особистим привітанням, яке надає відчуття затишку та важливості для бота. Це повідомлення містить короткий огляд функціоналу бота та надає інструкції щодо подальшої взаємодії.

У додатку А реалізовано функції `handle_start`, яка обробляє команду `/start`.

Функція спочатку перевіряє наявність файлу `user_ids.txt` для зчитування ідентифікаторів користувачів. Якщо файл відсутній, вона автоматично створює порожній список користувачів.

Потім функція перевіряє, чи ідентифікатор користувача вже присутній у списку. Якщо ні, вона додає його до файлу `user_ids.txt`.

Після цього у боті на рисунку 3.4 команда надсилає привітальне повідомлення користувачу, використовуючи його ім'я або інші персоналізовані дані, які вона отримала.

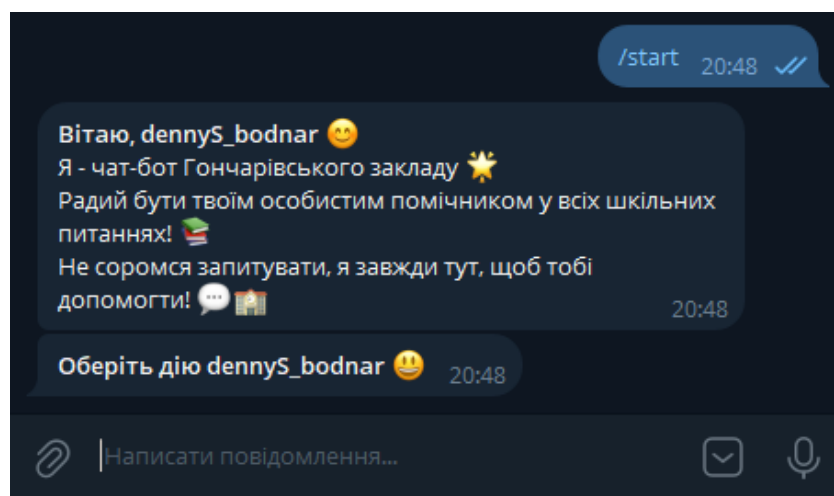


Рисунок 3.4 – Привітання у чат-боті

Функція `send_main_menu` створює головне меню з кнопками для різних дій, доступних користувачеві. Якщо користувач є адміністратором, додаються додаткові кнопки. Реалізація функції наведена у додатку Б.

На рисунку 3.5 показано головне меню чат-бота, яке містить кілька кнопок.

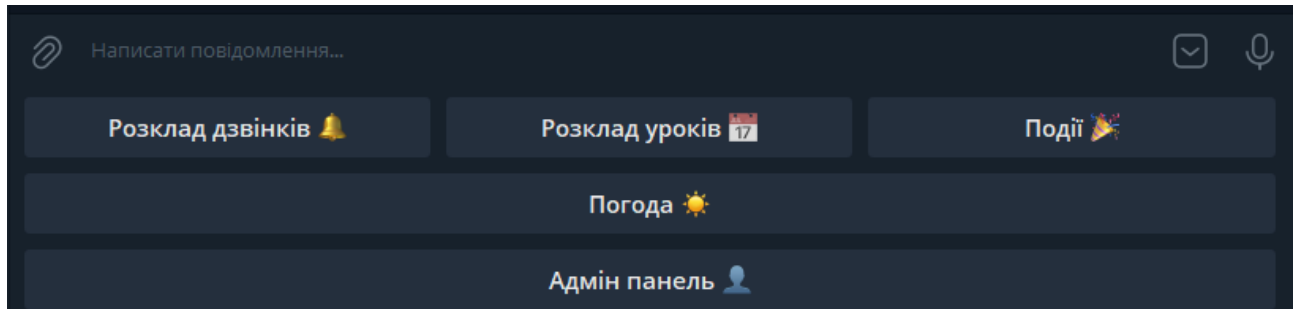


Рисунок 3.5 – Головне меню у чат-боті

Кожна кнопка відповідає за певну функцію, спрямовану на зручність та інформування користувачів:

1. Кнопка «Розклад дзвінків» відправляє користувачу розклад дзвінків для різних класів. Це здійснюється за допомогою функції `send_bell_schedule`.

2. Кнопка «Розклад уроків» надає розклад уроків для обраного класу та дня. Обробка цієї інформації відбувається через функцію `send_lesson_schedule`.

3. Кнопка «Події» інформує користувача про майбутні події, використовуючи функцію `send_events` для надсилання відповідних повідомлень.

4. Кнопка «Погода» надає інформацію про поточну погоду, яку отримують за допомогою функції `send_weather`.

5. Кнопка «Адмін-панель (лише для адміністраторів)» дозволяє адміністраторам редагувати розклад уроків, надсилати глобальні повідомлення та виконувати інші адміністративні завдання. Доступ до цієї панелі перевіряється функцією `is_admin`.

У додатку В кожна кнопка в головному меню має відповідну функцію для обробки дій користувача.

Для розкладу дзвінків використовується функція `send_bell_schedule`, для розкладу уроків – функція `send_lesson_schedule`, події інформують через

функцію `send_events`, поточну погоду надає функція `send_weather`, а адмін-панель перевіряє, чи є користувач адміністратором, і надає доступ до адміністративного меню через функцію `send_admin_menu`.

3.4 Шкільний графік

3.4.1 Розклад дзвінків

У розкладі дзвінків для початку використовується обробник, який викликається, коли користувач натискає на одну з кнопок з даною назвою. Він отримує повідомлення `message`, перевіряє, чи текст повідомлення збігається з одним з варіантів (1 КЛАС, 2-4 КЛАС, 5-9 КЛАС) та викликає функцію `send_bell_schedule_for_class`, щоб відправити розклад дзвінків відповідного класу, як наведено у лістингу 3.1.

Лістинг 3.1 – Обробник повідомлень

```
@bot.message_handler(func=lambda message: message.text in ["1 КЛАС",
"2-4 КЛАС", "5-9 КЛАС"])
def handle_class_schedule_request(message):
    class_number = message.text
    send_bell_schedule_for_class(bot, message, class_number)
```

Наступним написано функцію `send_bell_schedule`, яка призначена для відправлення користувачеві у Telegram клавіатури з кнопками для вибору класу розкладу дзвінків. Клавіатура з кнопками розкладу дзвінків показана на рисунку 3.6.

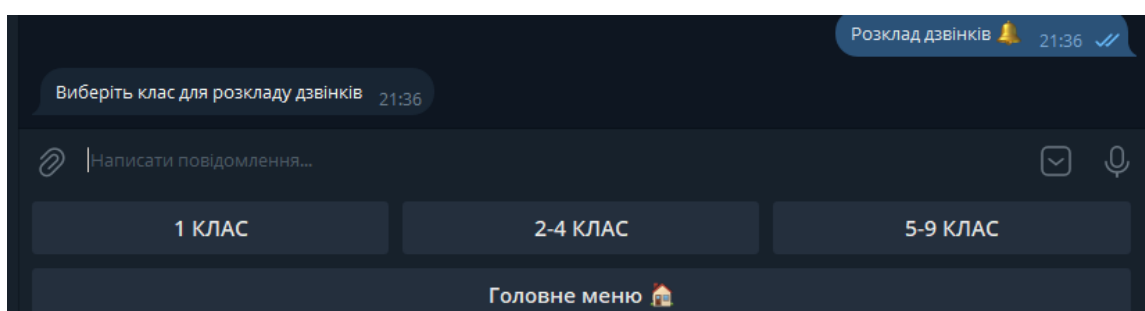


Рисунок 3.6 – Клавіатура з кнопками розкладу дзвінків

Після отримання параметрів `bot` (об'єкт телеграм-бота) та `message` (повідомлення від користувача) переходять до створення клавіатури використовуючи `types.ReplyKeyboardMarkup(resize_keyboard=True)`, функція створюється і може змінювати свій розмір для кращого відображення на різних пристроях. Для кожного класу (1-й, 2-4-й, та 5-9-й) та кнопки «Головне меню» створюються відповідні об'єкти класу `KeyboardButton`. Створені кнопки додаються до клавіатури за допомогою методу `add`, щоб вони відображалися на ній. З використанням методу `bot.send_message`, функція надсилає користувачеві повідомлення з текстом «Виберіть клас для розкладу дзвінків», до якого прикріплена створена клавіатура. У додатку Д показано, що користувач може вибрати клас для перегляду розкладу дзвінків, який йому потрібний.

Натискаючи відповідну кнопку користувач може продовжити взаємодію з ботом за допомогою цієї клавіатури, як наведено у лістингу 3.2.

Лістинг 3.2 – Обробник повідомлень розкладу дзвінків

```
def send_bell_schedule(bot, message):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
    btn1 = types.KeyboardButton("1 КЛАС")
    btn2 = types.KeyboardButton("2-4 КЛАС")
    btn3 = types.KeyboardButton("5-9 КЛАС")
    back = types.KeyboardButton("Головне меню 🏠")
    markup.add(btn1, btn2, btn3, back)
    bot.send_message(message.chat.id, text="Виберіть клас для
розкладу дзвінків", reply_markup=markup)
```

Функція `send_bell_schedule_for_class` відправляє розклад дзвінків для вибраного класу. Дану функцію наведено у додатку Е. Вона приймає три параметри: `bot` (об'єкт телеграм-бота), `message` (повідомлення від користувача) та `class_number` (вибраний клас).

3.4.2 Розклад уроків

Telegram-бот надає користувачам можливість переглядати розклад уроків для різних класів та днів тижня. Для початку опишемо обробку вибору класу

користувачем у Telegram-боті. Використовувався декоратор `@bot.message_handler`, який визначає функцію для обробки повідомлень, текст яких відповідає одному з наведених класів. Умовний вираз `func=lambda message: message.text in [1 клас, 2 клас, 3 клас, 4 клас, 5 клас, 6 клас, 7 клас, 8 клас, 9 клас]` фільтрує повідомлення, які мають текст, що відповідає назвам класів.

У лістингу 3.3 описано функцію-обробник `handle_class_selection`, яка зберігає вибраний клас у глобальну змінну `selected_class`, після чого викликає функцію `send_day_selection_menu` для відправлення меню вибору дня тижня.

Лістинг 3.3 – Обробник повідомлень розкладу уроків

```
@bot.message_handler(
    func=lambda message: message.text in ["1 клас", "2 клас", "3
клас", "4 клас", "5 клас", "6 клас", "7 клас",
                                         "8 клас", "9 клас"])
def handle_class_selection(message):
    global selected_class
    selected_class = message.text
    send_day_selection_menu(bot, message)
```

Результат обробника повідомлень розкладу уроків показано на рисунку 3.7, для всіх класів вибору.

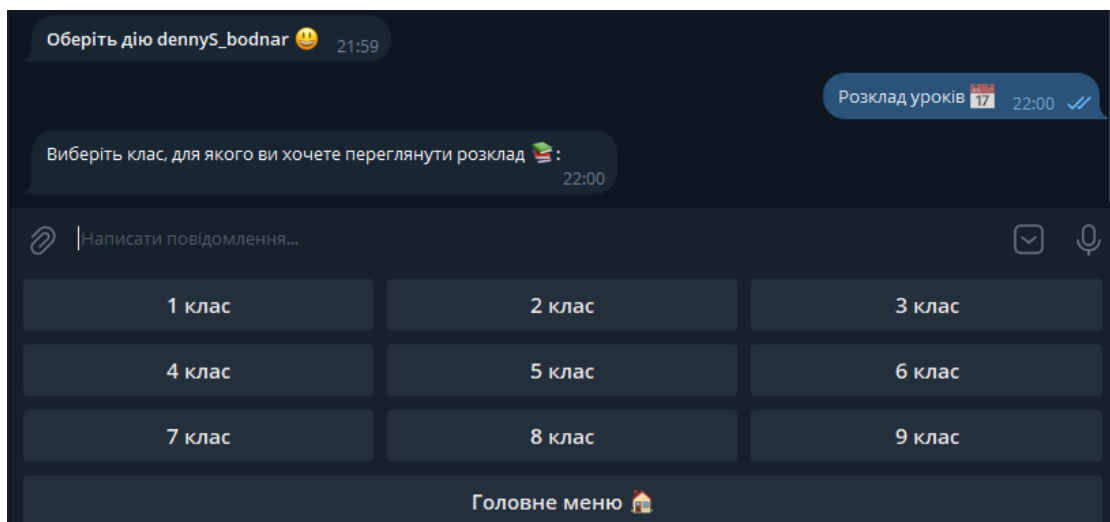


Рисунок 3.7 – Клавіатура з кнопками класів для вибору розкладу уроків

Аналогічно, для обробки вибору дня тижня використовується декоратор `@bot.message_handler`, який визначає функцію для обробки повідомлень, текст

яких відповідає одному з наведених днів тижня. Умовний вираз `func=lambda message: message.text in [Понеділок, Вівторок, Середа, Четвер, П'ятниця]` фільтрує повідомлення, які мають текст, що відповідає назвам днів тижня. У лістингу 3.4 описано функцію-обробник `handle_day_selection`, яка зберігає вибраний день у глобальну змінну `selected_day`, після чого викликає функцію `send_class_schedule` для відправлення розкладу уроків для вибраного класу та дня.

Лістинг 3.4 – Обробка вибору дня тижня

```
@bot.message_handler(func=lambda message: message.text in
["Понеділок", "Вівторок", "Середа", "Четвер", "П'ятниця"])
def handle_day_selection(message):
    global selected_class, selected_day
    selected_day = message.text
    send_class_schedule(bot, message,
str(selected_class.split()[0]), selected_day.lower())
```

Після обробки вибору дня тижня отримано результат на рисунку 3.8.

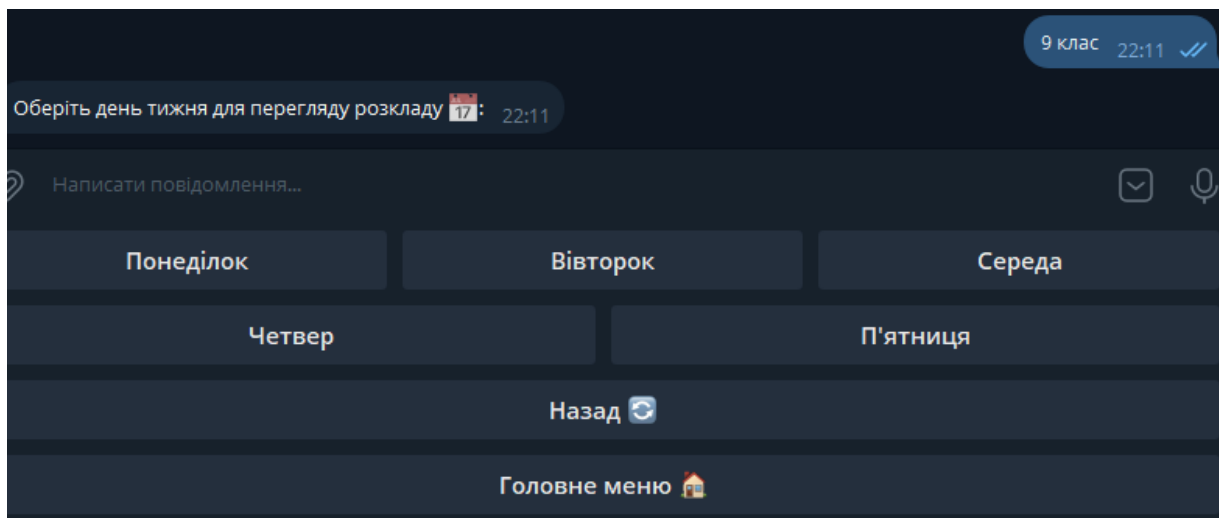


Рисунок 3.8 – Вибір дня тижня у чат-боті для перегляду розкладу уроків

У додатку Ж показано сервіс `moskari.io`, який використовувалося для збереження розкладу уроків. З цього сервісу береться API, в якому зберігається інформація про розклад, який застосовується за допомогою посилання URL.

За допомогою цього URL можна отримати доступ до розкладу уроків, який був записаний раніше за допомогою відповідного коду.

```
API_URL = "https://665c4b543e4ac90a04d9299c.mockapi.io/schedule/"
```

Для завантаження розкладів уроків через API веб-сервера використовується функція `load_schedules`. За допомогою `requests.get(API_URL)` виконується GET-запит до вказаного URL для отримання даних розкладу. Умова `if response.status_code == 200` перевіряє, чи був запит успішним (HTTP статус 200), і у такому разі повертаються отримані дані у форматі JSON за допомогою `response.json()`. У випадку, якщо статус відповіді не 200, виводиться повідомлення про помилку, і функція повертає порожній список, як наведено у лістингу 3.5.

Лістинг 3.5 – Функція для завантаження та збереження даних розкладу

```
def load_schedules():
    response = requests.get(API_URL)
    if response.status_code == 200:
        return response.json()

    else:
        print(f"Помилка завантаження розкладів:
        {response.status_code}")
        return []
```

Для збереження даних розкладу уроків у веб-сервері через API використовується функція `save_schedules`. У циклі `for schedule in schedules` кожен розклад у списку `schedules` оновлюється на сервері за допомогою PUT-запиту. Для кожного розкладу виконується PUT-запит, щоб оновити дані на сервері за допомогою його унікального ідентифікатора (`id`), який використовується для побудови URL. Дані розкладу передаються у форматі JSON через параметр `json`. Після виконання запиту перевіряється статус відповіді. Якщо статус відповіді не 200 (тобто запит не був успішним), виводиться повідомлення про помилку збереження, яке містить статус відповіді. Дане виконання функції наведено у лістингу 3.6.

Лістинг 3.6 – Збереження даних розкладу уроків у API

```
def save_schedules(schedules):
    for schedule in schedules:
        response = requests.put(f"{API_URL}/{schedule['id']}",
                                json=schedule)
        if response.status_code != 200:
            print(f"Помилка збереження розкладу для класу
{schedule['id']}: {response.status_code}")
```

Також прописується функція `send_lesson_schedule`, яка відправляє користувачеві розклад уроків для певного класу через бота. Після завантаження розкладу уроків за допомогою функції `load_schedules`, створюється клавіатура з можливістю зміни розміру за допомогою `types.ReplyKeyboardMarkup`. Кнопки для кожного класу створюються на основі завантаженого розкладу, а також створюється кнопка «Головне меню» для повернення до головного меню. Усі ці кнопки додаються до клавіатури за допомогою методу `markup.add()`. Відправляється повідомлення з запитом вибору класу та клавіатурою до чату користувача через бота, як наведено у лістингу 3.7.

Лістинг 3.7 – Функція для відправлення розкладу уроків

```
def send_lesson_schedule(bot, message):
    class_schedules = load_schedules()
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
    class_buttons = [types.KeyboardButton(f"{i['id']} клас") for i
in class_schedules]
    back = types.KeyboardButton("Головне меню 🏠")
    markup.add(*class_buttons, back)
    bot.send_message(message.chat.id, text="Виберіть клас, для
якого ви хочете переглянути розклад 📅: ", reply_markup=markup)
```

У функції `send_class_schedule` застосується відправлення користувачеві розкладу уроків для конкретного класу та дня тижня через бота. Першим кроком є завантаження розкладу уроків за допомогою функції `load_schedules`. Далі функція шукає інформацію про розклад для вибраного класу за його ідентифікатором. Якщо розклад для вказаного класу та дня тижня існує, то витягується розклад на вказаний день для обраного класу. Дана функція реалізована у лістингу 3.8.

Лістинг 3.8 – Функція для відправлення розкладу уроків для певного класу

та дня тижня

```
def send_class_schedule(bot, message, class_number, day):
    class_schedules = load_schedules()

    class_info = next((item for item in class_schedules if
item["id"] == str(class_number)), None)
    if class_info and day.lower() in class_info["days"]:
        schedule = class_info["days"][day.lower()]
        formatted_schedule = f"Розклад у {class_number} класі на
{day}:\n" + "\n".join(f"{i+1}) {subj}" for i, subj in
enumerate(schedule))
        bot.send_message(message.chat.id, formatted_schedule)
```

Цей розклад форматується у зручний для відображення текстовий вигляд, де вказується номер класу, день тижня та послідовність уроків. Відформатований розклад уроків відправляється користувачеві через бота у вигляді повідомлення, як показано у додатку 3.

3.5 Події

Кнопка «Події» в чат-боті дозволяє користувачам взаємодіяти з ботом шляхом обрання дій або отримання шкільної інформації через кнопки.

Функція `send_events (bot, message)` відповідає за відправлення користувачеві меню подій. Вона створює клавіатуру з кнопками, які представляють різні дії, і надсилає її у повідомленні. Клавіатура для вибору шкільної інформації показано на рисунку 3.9.

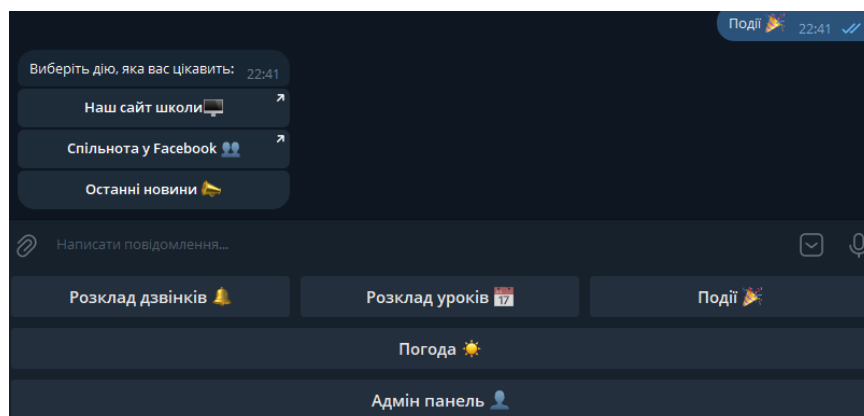


Рисунок 3.9 – Клавіатура для вибору шкільної інформації

Основні етапи роботи функції:

1. Створення клавіатури для кнопок за допомогою `types.InlineKeyboardMarkup()`.
2. Створення окремих кнопок для різних подій.
3. Додавання кожної кнопки до клавіатури за допомогою методу `markup.add()`.
4. Надсилання повідомлення з запитом вибору події та відповідною клавіатурою за допомогою `bot.send_message()`.

У лістингу 3.9 наведена функція, яка дає користувачеві можливість обрати дію, яка його цікавить, таку як перегляд веб-сайту школи, перехід до спільноти школи на Facebook або отримання останніх новин.



Лістинг 3.9 – Функція для відправлення подій

```
def send_events(bot, message):
    markup = types.InlineKeyboardMarkup()
    # Приклад кнопок для подій
    button1 = types.InlineKeyboardButton("Наш сайт школи 📄",
url="https://goncharivkaschool.e-schools.info/m/")
    button2 = types.InlineKeyboardButton("Спільнота у Facebook 👥",
url="https://www.facebook.com/profile.php?id=100076188379496")
    button3 = types.InlineKeyboardButton("Останні новини 📢",
callback_data="send_news")
    markup.add(button1)
    markup.add(button2)
    markup.add(button3)
    bot.send_message(message.chat.id, text="Виберіть дію, яка вас
цікавить: ", reply_markup=markup)
```

Дальше застосовується функція `send_news(bot, message)` для надсилання користувачеві останніх новин через чат-бот. Вона створює клавіатуру з кнопками, які містять заголовки новин і посилання на них. У додатку К показано результат застосування функції для надсилання останніх новин. Для цього спочатку отримується HTML-код сторінки з новинами за допомогою функції `fetch_news(school_url)`, після чого перевіряється наявність новин. Якщо вони є, проводиться парсинг HTML-коду для отримання списку новин, після чого для кожної новини створюється кнопка з її заголовком і посиланням, як наведено у

лістингу 3.10. Кнопки додаються до клавіатури, яка потім відправляється разом з повідомленням про новини. У випадку відсутності новин також відправляється відповідне повідомлення.

Лістинг 3.10 – Функція для відправлення останніх новин

```
def send_news(bot, message):
    markup = types.InlineKeyboardMarkup()
    html = fetch_news(school_url)
    if html:
        news_list = parse_news(html)
        for news in news_list:
            button = types.InlineKeyboardButton(news['title'],
url=news['link'])
            markup.add(button)
        bot.send_message(message.chat.id, text="Останні новини
:", reply_markup=markup)
    else:
        bot.send_message(message.chat.id, text="Останніх новин
немає :(")
```

У коді задано функцію `fetch_news`, яка відповідає на завантаження HTML-коду сторінки з новинами за вказаним URL. Вона використовує бібліотеку `requests`, щоб виконати GET-запит та отримати вміст сторінки. Після отримання вмісту перевіряється статус відповіді: якщо статус 200, що відповідає успішному запиту, то повертається текст HTML-сторінки. В іншому випадку, якщо статус відповіді не 200, виводиться повідомлення про помилку та повертається значення `None`. У лістингу 3.11 описано функцію для завантаження HTML-сторінки з новинами.

Лістинг 3.11 – Функція для завантаження HTML-сторінки з новинами

```
def fetch_news(url):
    response = requests.get(url)
    if response.status_code == 200:
        return response.text
    else:
        print(f"Failed to retrieve the webpage. Status code:
{response.status_code}")
        return None
```

Функція `parse_news` відповідає за парсинг HTML-коду сторінки з новинами і створення списку словників, кожний з яких містить заголовок новини та посилання на неї. Вона використовує бібліотеку `BeautifulSoup` для розбору HTML-коду. Після створення об'єкту `BeautifulSoup`, функція знаходить всі елементи `<div>` з класом `'sch_news_item'`, що представляють новини. Далі вона ітерується по кожному елементу новини, знаходить заголовок та посилання, та додає їх до списку новин. Якщо заголовок для новини не знайдено, виводиться повідомлення про помилку.

На виході функція `parse_news` повертає список новин, кожна з яких представлена словником із заголовком та посиланням на неї. На лістингу 3.12 наведено функцію для завантаження HTML-сторінки з новинами.

Лістинг 3.12 – Функція для завантаження HTML-сторінки з новинами

```
def parse_news(html):
    soup = BeautifulSoup(html, 'html.parser')
    news_items = soup.find_all('div', class_='sch_news_item')

    news_list = []
    for item in news_items:
        title_tag = item.find('div', class_='title')
        if title_tag:
            title = title_tag.text.strip()
            link_tag = item.find('a', href=True)
            link = link_tag['href'] if link_tag else 'No link
available'
            news_list.append({
                'title': title,
                'link': f"https://goncharivkaschool.e-
schools.info{link}"
            })
        else:
            print("Title not found for a news item")

    return news_list
```

Ефективне використання цих функцій значно полегшує взаємодію користувачів з телеграм-ботом, забезпечуючи їм зручний доступ до потрібної інформації та функціоналу.

3.6 Погода

Кнопка «Погода» у чат-боті дозволяє користувачам швидко отримати актуальну інформацію про погодні умови без необхідності залишати чат. Це особливо корисно для планування дня або прийняття рішень, пов'язаних з виходом на вулицю.

Для завантаження і надсилання даних про погоду через телеграм-бота використовують посилання `weather_url` для отримання інформації з веб-сайту. URL сторінки, звідки отримуються дані про погоду для Гончарівки беруться з сайту `sinoptik.ua`.

Функція `get_page_content(url)` виконує GET-запит за вказаним URL і повертає HTML-код сторінки. Вона використовує бібліотеку `requests` для виконання запиту та повертає текст відповіді, тобто HTML-код сторінки.

```
def get_page_content(url):
    response = requests.get(url)
    return response.text
```

Функція `parse_weather_today(page_content, start_tag, end_tag)` парсить текст між заданими HTML-тегами. Вона спочатку знаходить індекс початку тегу за допомогою методу `find()`. Якщо тег не знайдено, повертається значення `-1`. Потім вона знаходить індекс кінця тегу, також за допомогою `find()`. Використовуючи бібліотеку `BeautifulSoup`, функція парсить текст між цими тегами та повертає його, як наведено на лістингу 3.13.

Лістинг 3.13 – Функція для парсингу погоди на сьогодні

```
def parse_weather_today(page_content, start_tag, end_tag):
    start_index = page_content.find(start_tag)
    if start_index == -1:
        return ""
    end_index = page_content.find(end_tag, start_index)
    return BeautifulSoup(page_content[start_index:end_index],
        "html.parser").get_text()
```

Функція `get_today_date(timezone_str)` отримує поточну дату у заданому часовому поясі. Спочатку вона встановлює часовий пояс за допомогою бібліотеки `pytz`, а потім повертає поточну дату у форматі "YYYY-MM-DD", використовуючи бібліотеку `datetime`.

```
def get_today_date(timezone_str):
    timezone = pytz.timezone(timezone_str)
    return datetime.now(timezone).strftime("%Y-%m-%d")
```

Функція `get_weather_data(page_content)` отримує дані про погоду з HTML-коду сторінки. Вона використовує функцію `parse_weather_today` для парсингу мінімальної, максимальної температур, опису та поточної температури, як описано у лістингу 3.14.

Лістинг 3.14 – Функція для отримання даних про погоду

```
def get_weather_data(page_content):
    min_temp = parse_weather_today(page_content.lower(), '<div
class="min">', '</div>').replace("&deg;", "°")
    max_temp = parse_weather_today(page_content.lower(), '<div
class="max">', '</div>').replace("&deg;", "°")
    description = parse_weather_today(page_content.lower(), '<div
class="description">', '</div>').replace("&#039;", "'")
    current_temp = parse_weather_today(page_content.lower(), '<p
class="today-temp">', '</p>').replace("&deg;", "°")
    return min_temp, max_temp, description, current_temp
```

Функція `format_weather_message(min_temp, max_temp, current_temp, description)` створює форматоване повідомлення про погоду. У лістингу 3.15 описано повертання повідомлення у форматі, з використанням змінних: максимальна та мінімальна температура, поточна температура та опис погоди.

Лістинг 3.15 – Функція для форматування повідомлення про погоду

```
def format_weather_message(min_temp, max_temp, current_temp,
description):
    return f"*Погода на сьогодні
👉*\n{max_temp}\n{min_temp}\nЗараз на вулиці:
{current_temp}\n\n{description}"
```


Функція `send_weather(bot, message)` відправляє дані про погоду користувачеві. Вона встановлює часовий пояс на `Europe/Kiev` та отримує поточну дату за допомогою функції `get_today_date()`. Потім формує URL для запиту даних про погоду, додаючи поточну дату до базового URL. Отримує HTML-код сторінки з даними про погоду за допомогою `get_page_content(url)`. Після цього, використовуючи `get_weather_data(page_content)`, отримуються дані про погоду та форматове повідомлення про погоду за допомогою `format_weather_message(min_temp, max_temp, current_temp, description)`. Бот відправляє це повідомлення користувачеві, використовуючи метод `bot.send_message()`, з форматуваним текстом у режимі Markdown. У лістингу 3.16 наведено функцію для відправлення даних про погоду з сайту `Sinoptik.ua`.

Лістинг 3.16 – Функція для відправлення даних про погоду

```
def send_weather(bot, message):
    timezone_str = "Europe/Kiev"
    today = get_today_date(timezone_str)
    url = weather_url + today

    page_content = get_page_content(url)
    min_temp, max_temp, description, current_temp =
get_weather_data(page_content)

    weather_message = format_weather_message(min_temp, max_temp,
current_temp, description)
    bot.send_message(message.chat.id, text=weather_message,
parse_mode="Markdown")
```

Загальний вигляд кнопки «Погода» показано на рисунку 3.10.

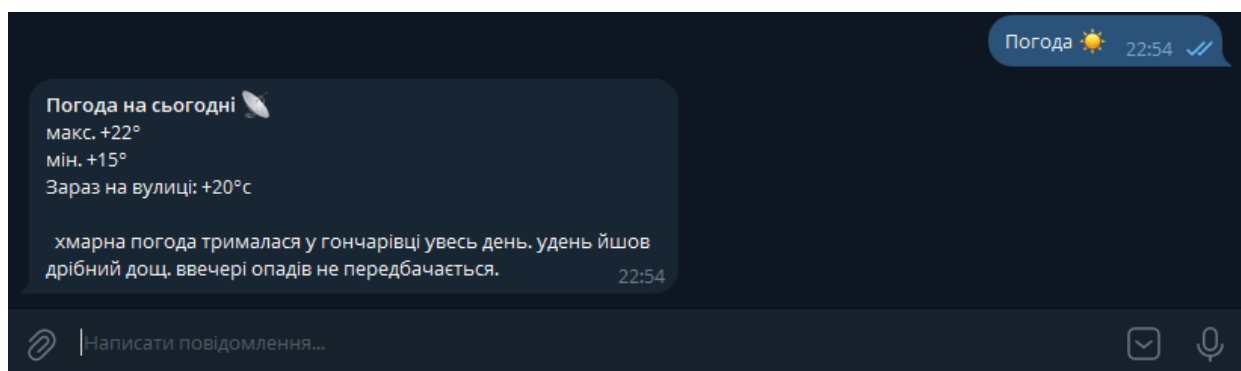


Рисунок 3.10 – Кнопка «Погода»

Погода з рисунку прогнозується чат-ботом як хмарна, з температурою +20°C, і показано, що максимальна температура протягом дня +22°C, а мінімальна +15°C. Написано у чат-боті, що хмарна погода трималася у Гончарівці увесь день. Удень йшов дрібний дощ. Ввечері опадів не передбачається.

3.7 Адмін-панель

Кнопка «Адмін-панелі» дозволяє адміністраторам здійснювати управління ботом та забезпечує їм зручний доступ до функцій адміністрування без необхідності вводити спеціальні команди або виконувати складні дії.

Для початку прописується код, який складається з кількох функцій, кожна з яких виконує певну функцію в адмініструванні розкладу уроків бота. Наприклад, функція `handle_admin_panel` відображає панель адміністратора після отримання команди «Адмін-панель». `handle_edit_lesson_schedule` дозволяє редагувати розклад уроків за запитом, тоді як `process_global_message` обробляє та надсилає глобальне повідомлення всім користувачам бота. `handle_send_global_message` та `handle_send_news` відповідають відповідно за надсилання глобальних повідомлень та новин користувачам. Використання кнопки «Адмін-панель» показано на рисунку 3.11.



Рисунок 3.11– Кнопка «Адмін-панель»

Крім того, є функції, які дозволяють адміністратору редагувати розклад для конкретного класу або дня: `handle_edit_class_schedule` та `handle_edit_day_schedule`. Також є можливість редагувати конкретний урок в

розкладі за допомогою функції `handle_edit_lesson`, та оновлювати назву конкретного уроку за допомогою `process_new_lesson_name`, як наведено у додатку Л.

Написано скрипт для Telegram-бота, який надає адміністратору можливість редагувати розклади уроків для різних класів та надсилати глобальні повідомлення користувачам. Для редагування розкладу уроків використовується клавіатура спочатку для вибору класу, а потім для вибору дня тижня, як показано на рисунку 3.12.

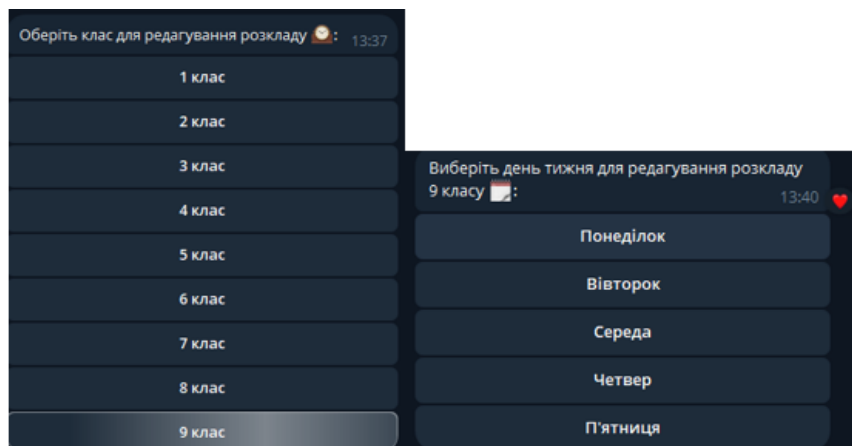


Рисунок 3.12 – Клавіатура змін розкладу уроків для вибору класу і дня тижня

Після вибору дня тижня відбувається запит, у якому потрібно вказати який урок потрібно редагувати і написати назву нового. На рисунку 3.13 показано редагування вибраного предмету зі списку.

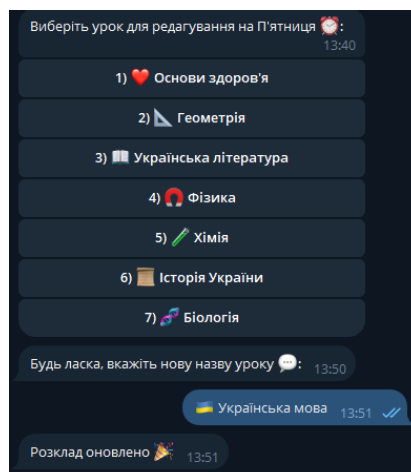


Рисунок 3.13 – Редагування вибраного уроку

Отримання розкладів та їх зберігання здійснюється через HTTP-запити. Функція `load_class_schedules()` завантажує розклади, а `save_class_schedules(schedules)` зберігає їх у системі.

Права адміністратора перевіряються функцією `is_admin(user_id)`, яка визначає, чи належить ID користувача до списку адміністраторів.

```
def is_admin(user_id):
    admin_ids = [937416743]
    return user_id in admin_ids
```

У лістингу 3.17 описано глобальні повідомлення, які надсилаються через функцію `send_global_message(bot, message_text)`. Вона зчитує список користувачів з файлу та надсилає кожному з них вказане повідомлення, як показано на рисунку 3.14.

Лістинг 3.17 – Функція для глобальних повідомлень

```
def send_global_message(bot, message_text):
    with open("user_ids.txt", 'r') as file:
        user_ids = file.readlines()
    for user_id in user_ids:
        try:
            bot.send_message(user_id.strip(), message_text,
                parse_mode='Markdown')
        except Exception as e:
            print(f"Не вдалося надіслати повідомлення користувачу {user_id.strip()}: {e}")
```

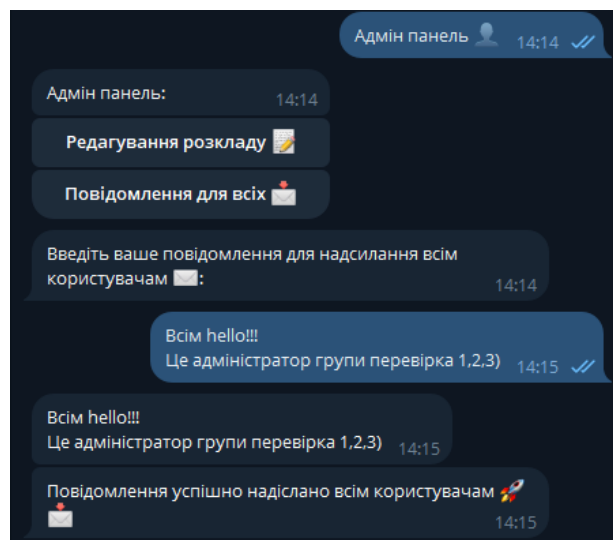


Рисунок 3.14 – Надсилання повідомлення всім користувачам

Для спілкування з адміністратором створюється адміністративне меню, яке відправляється за допомогою функції `send_admin_menu(bot, message)`. Воно містить кнопки для редагування розкладу та надсилання повідомлень, як наведено у лістингу 3.18.

Лістинг 3.18 – Функція для адміністративного меню

```
def send_admin_menu(bot, message):
    markup = types.InlineKeyboardMarkup()
    button1 = types.InlineKeyboardButton("Редагування розкладу
✎", callback_data="edit_lesson_schedule")
    button2 = types.InlineKeyboardButton("Повідомлення для всіх
✉", callback_data="send_global_message")
    markup.add(button1)
    markup.add(button2)
    bot.send_message(message.chat.id, text="Адмін-панель: ",
reply_markup=markup)
```

Адміністративне меню задає зміни для роботи чат-бота, забезпечуючи головній людині доступ до ключових опцій та сервісів. Воно є прихованою точкою для взаємодії, дозволяючи легко перейменовувати розклад і надсилати повідомлення усім користувачам.

3.8 Висновок до третього розділу

У третьому розділі кваліфікаційної роботи детально розглянуто роботу чат-бота «Гончарівський заклад», що має різноманітний функціонал. Починаючи з процесу реєстрації та налаштування бота, кожний підхід його роботи докладно описано, а головне меню включає в собі отримання розкладу уроків та дзвінків, подій, погодних умов та для адміністратора редагування уроків та надсилання повідомлень всім учасникам.

Однією з ключових особливостей цього чат-бота є його різноманітний функціонал. Користувачі можуть легко отримати розклад уроків, переглянути останні новини школи, отримати інформацію про погоду, а також дізнатися про події, що відбуваються в навколишньому середовищі. Це забезпечує

багатогранність інформації, яка надається користувачам, та робить чат-бот універсальним інструментом для отримання різноманітної інформації.

Крім того, функції для роботи з розкладом уроків, новинами та погодою надають користувачам інформацію у зручному форматі, що сприяє її зрозумінню та ефективному використанню. Наприклад, користувач може легко переглянути розклад уроків на певний день або тиждень, переглянути останні новини з використанням зручного інтерфейсу чат-бота, а також дізнатися про погодні умови на поточний момент чи на певний період.

Ефективність роботи з даними є ще однією важливою характеристикою цього чат-бота. Використання API для завантаження та збереження даних дозволяє швидко та надійно отримувати та зберігати інформацію про розклад уроків, новини та погоду. Це дозволяє забезпечити користувачів актуальною інформацією та забезпечує ефективну роботу чат-бота.

Можливість взаємодії з подіями та кнопкою «Події» розширює можливості користувачів отримувати більше інформації про життя школи та події, що відбуваються. Це дозволяє користувачам бути в курсі останніх подій та заходів школи та взаємодіяти з ними.

Чат-бот «Гончарівський заклад» є потужним інструментом для спілкування та надання інформації. Даний чат-бот є у вільному доступі у Telegram і має свій хостинг, через який працює і надсилає усі повідомлення користувачам.

РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Менеджмент безпеки учнів і вчителів

Менеджмент безпеки – це система наукових знань та досвіду забезпечення безпеки, втілених у діяльності професійних управлінців для досягнення цілей системи безпеки через використання праці, інтелекту та мотивів поведінки людей.

Менеджмент безпеки учнів і вчителів є складним, оскільки пов'язаний із непередбачуваними подіями. Він здійснюється у двох формах: безпосередній і опосередкованій. Безпосереднє управління передбачає діяльність на правовій або делегованій основі з правом прийняття і реалізації управлінських рішень. Опосередковане управління включає участь у підготовці, прийнятті та реалізації управлінських рішень.

Процес управління безпекою полягає у перетворенні інформації про об'єкт управління або проблемну ситуацію на управлінські рішення. У цьому процесі реалізуються як загальні, так і спеціальні функції управління. Загальні функції включають прогнозування, планування, організацію, регулювання, координацію та контроль.

Прогнозування дозволяє створювати гарантії ефективності менеджменту. З урахуванням результатів прогнозу і детального аналізу ситуації на відповідній території або об'єкті здійснюється планування.

Планування заходів з безпеки включає розробку превентивних та ситуаційних планів, а також перспективних і поточних програм. Це дозволяє підтримувати пропорційність і раціональність у використанні ресурсів. Превентивні плани – це програми дій з підвищення надійності обладнання та стійкості роботи об'єктів. Ситуаційні плани дій включають аналітичний та оперативний матеріал з локалізації і ліквідації аварій та надзвичайних ситуацій.

Організація полягає у створенні організаційного механізму, формуванні управлінських підрозділів, підборі та розстановці кадрів. Це забезпечує взаємозв'язок і ефективність усіх інших функцій управління.

Спеціальні функції управління безпекою школярів і вчителів включають запобігання і мінімізацію наслідків небезпечних подій, організацію захисту населення і територій в умовах небезпечних ситуацій, а також ліквідацію наслідків небезпечних ситуацій.

Менеджмент безпеки включає: захист населення від негативних наслідків надзвичайних ситуацій сповіщеннями та інформуванням, укриття у захисних спорудах, здійснення евакуаційних заходів, медичний і психологічний захист, навчання діям у надзвичайних ситуаціях [31-34]

Ліквідація наслідків небезпечних ситуацій передбачає проведення рятувальних робіт, життєзабезпечення потерпілого населення та персоналу, проведення відновлювальних робіт, відшкодування збитків та надання допомоги постраждалим [35].

4.2 Заходи, що покращують умови праці у школі

Покращення умов праці в школі вимагає уважного підходу до забезпечення безпеки та комфорту для всіх учасників навчального процесу. Це означає ретельне спостереження за умовами праці та навчання, щоб уникнути потенційних небезпек та створити оптимальне середовище для навчання та розвитку.

Одним із ключових напрямків покращення умов праці є проведення регулярних тренінгів та інструктажів з безпеки для педагогічного та адміністративного персоналу. Це допомагає усвідомити потенційні ризики та правильно реагувати у небезпечних ситуаціях. Разом з тим, важливо розробити чіткі процедури дії в екстрених ситуаціях, таких як пожежа чи медичний випадок, щоб забезпечити швидке та ефективне реагування на них.

Будівля школи повинна забезпечувати оптимальні умови для організації навчально-виховного процесу, відпочинку, харчування учнів. Будівлю школи на території земельної ділянки слід розташовувати так, щоб рівень освітлення та інсоляції навчальних приміщень був оптимальним, а взаємозв'язок між ділянкою і будівлею – раціональним. Блочно-секційна і павільйонна частина навчальних

будівель найбільш раціональні для використання. В одній секції слід розміщувати максимум три паралелі класів для дітей одного віку або дві паралелі класів двох суміжних вікових груп (1 і 2, 3 і 4). Будівлі загальноосвітніх навчальних закладів повинні мати не більше трьох поверхів. При розміщенні загальноосвітніх навчальних закладів у попередньо збудованих 4-, 5- поверхових будівлях, четвертий і п'ятий поверхи відводять під кабінети, які мають найменше навчальне навантаження. Кількість учнів загальноосвітнього навчального закладу не повинна перевищувати його проектну місткість.

У будівлях загальноосвітніх шкіл передбачаються такі функціональні групи приміщень: класні кімнати, навчальні кабінети, лабораторії з лаборантськими, кабінети трудового навчання та профорієнтації, продовженого дня, фізкультурно-спортивні, медичні, актовий зал, бібліотека, адміністративні, допоміжні приміщення – вестибюль. При розміщенні школи в пристосованій будівлі набір приміщень, їх площа визначаються за погодженням з територіальною санітарно-епідеміологічною станцією у кожному випадку окремо, виходячи з типу загальноосвітнього навчального закладу, кількості і віку учнів, кількості класів [36].

Для безпечного життя людини, її здоров'я та самопочуття важливим є стан навколишнього середовища. Нового та особливого характеру це питання набуває під час війни в Україні. При здійсненні ракетних ударів та обстрілів різних інфраструктурних та промислових об'єктів, будинків, машин, а також в результаті вибухів та пожеж в докільля викидається надзвичайно велика кількість шкідливих речовин, цей негативний вплив підсилюється можливим використанням хімічної зброї.

Психологічна безпека в школі вимагає створення підтримуючого та відкритого середовища для всіх учасників. Розробка програм психологічної підтримки та відкрите спілкування сприяють створенню дружньої та доброзичливої атмосфери, що сприяє психічному благополуччю усіх членів шкільної спільноти.

Досягнення безпеки й комфорту в школі включає в себе постійний моніторинг та оновлення безпекових стандартів і процедур відповідно до змін у

середовищі та потребах шкільної громади. З цією метою, важливо проводити оцінку ризиків та аналіз потенційних небезпек на різних етапах навчального процесу, від організації уроків до управління навчальними заходами.

Окрім того, активна участь громадськості та батьків у забезпеченні безпеки в школі грає важливу роль у підтримці цієї мети. Спільна робота шкільної адміністрації, педагогічного колективу, учнів, батьків та громадських організацій є важливим елементом забезпечення ефективного функціонування системи безпеки в школі [37].

4.3 Психофізіологічне розвантаження для вчителів та учнів

Психофізіологічне розвантаження для вчителів та учнів є невід'ємною складовою охорони праці у навчальних закладах. Навіть за умови, що навантаження та стрес є невід'ємною частиною навчального процесу, важливо забезпечити педагогічний персонал та учнівську аудиторію ефективними засобами та стратегіями для зниження негативних наслідків на здоров'я та загальний стан благополуччя.

Психофізіологічне розвантаження вчителів та учнів здійснюється за допомогою різноманітних практик та методів, спрямованих на зняття стресу, відновлення енергії та збереження психічного здоров'я.

Для вчителів, це може означати використання технік релаксації, таких як медитація, глибоке дихання або йога, які допомагають зняти напругу та покращити концентрацію. Крім того, фізична активність, якщо це можливо, є важливим засобом психофізіологічного розвантаження. Короткі перерви для фізичних вправ або навіть невеликі прогулянки можуть підвищити ефективність вчителя та поліпшити його настрій.

Створення сприятливої атмосфери в класі також може позитивно впливати на психофізіологічне благополуччя учнів. Підтримка та позитивна атмосфера можуть допомогти учням почувати себе комфортно та впевнено, що зменшує рівень стресу та покращує загальний настрій.

Забезпечення психофізіологічного розвантаження для вчителів та учнів є невід'ємною складовою процесу охорони праці у навчальних закладах. Важливо розробляти та застосовувати ефективні стратегії, які сприяють збереженню здоров'я та загального благополуччя педагогічного персоналу та учнівської громади [38].

4.4 Висновок до четвертого розділу

Четвертий розділ присвячений менеджменту безпеки учнів і вчителів, покращенню умов праці у школах, а також заходам для психофізіологічного розвантаження. Розглянуті аспекти показують, що ефективний менеджмент безпеки є комплексним процесом, що включає як безпосереднє, так і опосередковане управління. Він передбачає реалізацію загальних функцій управління, таких як прогнозування, планування, організація, регулювання, координація та контроль.

Покращення умов праці в школах потребує систематичного підходу, що включає регулярні тренінги та інструктажі з безпеки, оптимальне планування та облаштування навчальних приміщень, врахування екологічних факторів, особливо в умовах війни. Психологічна безпека також є важливою складовою, яка забезпечується через створення підтримуючого середовища та впровадження програм психологічної підтримки.

Психофізіологічне розвантаження для вчителів та учнів є невід'ємною частиною охорони праці, що сприяє зниженню рівня стресу та підтримці загального благополуччя. Важливо впроваджувати техніки релаксації, фізичну активність та створювати позитивну атмосферу в навчальному середовищі.

Таким чином, досягнення безпеки й комфорту в школах можливе завдяки комплексному підходу до управління безпекою, активній участі всіх членів шкільної спільноти та постійному вдосконаленню безпекових стандартів і процедур. Це створює сприятливі умови для навчання та роботи, сприяючи гармонійному розвитку учнів і ефективній діяльності педагогічного колективу.

ВИСНОВКИ

У кваліфікаційній роботі бакалавра досліджено різні погляди використання чат-ботів та їх вплив на сучасний світ. В роботі аналізуються технологічні, функціональні та безпекові підходи впровадження чат-ботів, що дає можливість комплексно оцінити їх значення та перспективи розвитку.

У першому розділі досліджено різноманітні підходи використання чат-ботів у сучасному світі. Аналіз цих підходів дозволяє зробити висновок про значущість цієї технології в різних сферах. Використання чат-ботів у сфері обслуговування клієнтів, освіти та управлінні соціальними мережами підтверджує глибокий вплив цієї інновації на різні аспекти повсякденного життя. Разом із перевагами чат-ботів, такими як підвищення ефективності та зручності, виникають виклики, такі як необхідність постійного удосконалення та проблеми зі сприйняттям природної мови. Ці аспекти потребують подальшого дослідження та вдосконалення для максимального використання потенціалу чат-ботів. Розділ також розкриває різні методи створення чат-ботів, що створює безмежні можливості для інновацій та творчості у цій сфері.

У другому розділі досліджено ключові компоненти для створення чат-ботів: платформи (Telegram, Instagram, Viber), середовище програмування (PyCharm), мова програмування (Python), Telegram API та бібліотека Telebot. Кожен компонент має свої унікальні особливості та переваги. Наприклад, Telegram відзначається своєю популярністю та багатим функціоналом, Instagram приваблює велику аудиторію, а Viber має інтуїтивно зрозумілі інструменти. PyCharm допомагає забезпечити продуктивність розробки, а Python відзначається простим та широким спектром бібліотек. Також розглянуто Telegram API та бібліотеку Telebot, які полегшують взаємодію з Telegram. Всі ці компоненти забезпечують необхідні можливості для розробки чат-ботів з високим функціоналом, що дозволяє розробникам фокусуватися на інноваціях та обслуговуванні користувачів.

У третьому розділі роботи детально розглянуто та виконано чат-бота «Гончарівський заклад», який має різноманітний функціонал, починаючи з

процесу реєстрації та налаштування. Кожен вид його роботи описано, а головне меню включає в себе отримання розкладу уроків та дзвінків, подій, погодних умов, а для адміністратора – редагування уроків та надсилання повідомлень. Користувачі можуть отримати різноманітну інформацію, що робить чат-бот універсальним інструментом. Використання API дозволяє отримувати та зберігати інформацію надійно та швидко. Можливість взаємодії з подіями розширює можливості користувачів отримувати більше інформації. Чат-бот «Гончарівський заклад» являє собою ефективний засіб для комунікації та надання інформації. Він доступний у Telegram і працює через власний хостинг, забезпечуючи користувачам можливість отримувати повідомлення.

У розділі «Безпека життєдіяльності, основи охорони праці» детально висвітлено важливі види створення безпечного середовища в навчальних закладах. Зокрема, визначено важливість дотримання норм і правил безпеки для забезпечення безпечності та здоров'я учасників навчального процесу. Представлені заходи, спрямовані на поліпшення умов праці в школах, включають в себе психофізіологічне розвантаження для вчителів і учнів. Ці заходи сприяють збереженню здоров'я та підвищенню загального рівня безпеки. Для успішної реалізації цих заходів забезпечують постійний моніторинг та контроль за їх виконанням, а також надання відповідного навчання та підтримку персоналу у питаннях безпеки та охорони праці. Зазначено, що введення цих заходів є ключовим для створення безпечної та сприятливої освітньої атмосфери, що сприяє успішному навчанню та розвитку учнів і забезпечує безпечні умови праці для персоналу навчального закладу.

ПЕРЕЛІК ДЖЕРЕЛ

1. Чат-бот [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/чат-бот>
2. Чат-бот [Електронний ресурс] – Режим доступу до ресурсу: <https://sendpulse.ua/support/glossary/chatbot>
3. Як чат-боти впливають на зростання продажів та чому AI це вже не ілюстратор? [Електронний ресурс] – Режим доступу до ресурсу: <https://nachasi.com/tech/2021/08/25/hto-tse-vygadav-yak-chat-boty-vplyvayut-na-rist-prodazhiv-ta-chomu-ai-tse-vzhe-ne-illyustrator/>
4. Від тесту Тьюрінга до ChatGPT [Електронний ресурс] – Режим доступу до ресурсу: <https://mc.today/uk/shho-take-chat-bot/>
5. Чат-бот для сервісу та послуг [Електронний ресурс] – Режим доступу до ресурсу: https://gerabot.com/service_type
6. Ю. Гладь. Застосування методу екстремального наведення для антенних систем супутникового зв'язку / Ю. Гладь, Н. Гашин // Матеріали V науково-технічної конференції «Інформаційні моделі, системи та технології», 1-2 лютого 2018 року. — Т. : ТНТУ, 2018. — С. 6. — (Секція 1. Математичне моделювання).
7. Чат-боти для навчання: огляд найпопулярніших та особливості використання [Електронний ресурс] – Режим доступу до ресурсу: <https://teach-hub.com/chat-boty-dlia-navchannia-ohliad-na-ypopuliarnishykh-ta-osoblyvosti-vykorystannia/>
8. Telegram-боти для інтернет-маркетолога [Електронний ресурс] – Режим доступу до ресурсу: <https://webpromoeexperts.net/ua/blog/telegram-boti-dlya-internet-marketologa/>
9. Ключові переваги чат-бота [Електронний ресурс] – Режим доступу до ресурсу: https://gerabot.com/article/klyuchovi_perevagi_chatbota_pered_mobilnim_dodatkom

10. Чат-бот. Переваги, засоби використання та як створити бота [Електронний ресурс] – Режим доступу до ресурсу: https://gerabot.com/article/detalno_pro_chatboti

11. Чому вам потрібні чат-боти? [Електронний ресурс] – Режим доступу до ресурсу: <https://dalistrategies.com/ua/shho-take-chat-bot/>

12. Чат-боти: плюси і мінуси [Електронний ресурс] – Режим доступу до ресурсу: <https://ukr-bot.com/plyusi-ta-minusi-chat-botiv/>

13. Як створити чат-бота для Telegram-каналу [Електронний ресурс] – Режим доступу до ресурсу: <https://netpeak.net/uk/blog/yak-stvoriti-chat-bot-dlya-telegram-kanalu-instruksiya-dlya-administratoriv/>

14. Все про чат-боти: переваги та схема роботи [Електронний ресурс] – Режим доступу до ресурсу: <https://interkassa.com/blog/vse-pro-chat-boti-perevagi-tipi-ta-shema-roboti>

15. Що таке чат-бот та як створити бота в телеграм [Електронний ресурс] – Режим доступу до ресурсу: <https://marchenko.marketing/scho-take-chat-bot-ta-yak-stvoriti-bota-v-telegram/>

16. BotFather. Можливості та функціонал [Електронний ресурс] – Режим доступу до ресурсу: https://gerabot.com/article/botfather_mozhlivosti_ta_funkcional

17. Чат-бот для покращення сервісу: 5 причин, чому працює [Електронний ресурс] – Режим доступу до ресурсу: <https://cases.media/article/chat-bot-dlya-rokrashennya-servisuv-5-prichin-chomu-ce-pracyuye>

18. Чат-бот Instagram [Електронний ресурс] – Режим доступу до ресурсу: <https://itforce.ua/blog/chat-bot-v-instagram/>

19. Чат-бот у Viber [Електронний ресурс] – Режим доступу до ресурсу: <https://hostpro.ua/blog/ua/viber-chatbot/>

20. Як створити чат-бота у вайбері [Електронний ресурс] – Режим доступу до ресурсу: <https://helpcrunch.com/blog/uk/chat-bot-u-viber/>

21. JetBrains PyCharm [Електронний ресурс] – Режим доступу до ресурсу: <https://itpro.ua/product/jetbrains-pycharm/?tab=description>

22. Середовище розробки для Python [Електронний ресурс] – Режим доступу до ресурсу: <https://foxminded.ua/seredovyshche-rozrobky-python/>

23. Використання середовища розробки PyCharm при вивченні програмування [Електронний ресурс] – Режим доступу до ресурсу: https://informatika.udpu.edu.ua/?page_id=7591

24. Що таке мова програмування Python? [Електронний ресурс] – Режим доступу до ресурсу: <https://freehost.com.ua/ukr/faq/wiki/chto-takoe-jazik-programmirovaniya-python/>

25. Все, що ви маєте знати про Python [Електронний ресурс] – Режим доступу до ресурсу: <https://geek.justjoin.it/все-що-ви-маєте-знати-про-python-які-в-нього-н/>

26. А. Паламар. Програмне забезпечення комп'ютерної системи для моніторингу стану пристроїв безперебійного живлення. А. Паламар, В. Дьомін, В. Волоський. Матеріали X науково-технічної конференції «Інформаційні моделі, системи та технології», 7–8 грудня 2022 року. – Т. : ТНТУ, 2022. – С. 83. – (Комп'ютерні системи та мережі).

27. Яка в Telegram мова програмування? [Електронний ресурс] – Режим доступу до ресурсу: <https://lemon.school/blog/yaka-u-telegram-mova-programuvannya>

28. Чи можна використовувати API Telegram? [Електронний ресурс] – Режим доступу до ресурсу: <https://telegramm.com.ua/pytannya-ta-vidpovidi/chimozhna-vykorystovuvaty-api-telegram/>

29. Telegram-боти на Python: огляд п'яти найкращих фреймворків [Електронний ресурс] – Режим доступу до ресурсу: <https://drukarnia.com.ua/articles/telegram-boti-na-python-oglyad-p-yati-naikrashikh-freimvorkiv-bibliotek-L7UA7>

30. С. Грушак, С. Павленко. Віртуальні приватні мережі та їх використання в захищених корпоративних мережах. Комп'ютерні та інформаційні системи і технології (Харків, Україна, 2019). – 202 с.

31. V. Kozlovskiy, Y. Balanyuk, H. Martyniuk, O. Nazarevych, L. Scherbak and G. Shymchuk, «Information Technology for Estimating City Gas Consumption During the Year,» 2022 International Conference on Smart Information Systems and

Technologies (SIST), Nur-Sultan, Kazakhstan, 2022, pp. 1-4, doi: 10.1109/SIST54437.2022.9945786.

32. Approach to gas consumption process forecasting on the basis of a mathematical model in the form of a random cyclic process / Serhii Lupenko, Iaroslav Lytvynenko, Oleg Nazarevych, Grigorii Shymchuk, Volodymyr Hotovych // ICAAEIT 2021, 15-17 December 2021. – Tern. : TNTU, Zhytomyr «Publishing house „Book-Druk“» LLC, 2021. – P. 213–219. – (Mathematical modeling in power engineering and information technologies).

33. Additive mathematical model of gas consumption process / Iaroslav Lytvynenko, Serhii Lupenko, Oleh Nazarevych, Hryhorii Shymchuk, Volodymyr Hotovych // Scientific Journal of TNTU. – Tern. : TNTU, 2021. – Vol 104. – No 4. – P. 87–97.

34. Y. Leshchyshyn, L. Scherbak, O. Nazarevych, V. Gotovych, P. Tymkiv and G. Shymchuk, «Multicomponent Model of the Heart Rate Variability Change-point,» 2019 IEEE XVth International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH), Polyana, Ukraine, 2019, pp. 110-113, doi: 10.1109/MEMSTECH.2019.8817379.

35. В. Зацарний, О. Зацарна, О. Землянська, Н. Праховнік. Безпека життєдіяльності. Навчальний посібник. – Київ: НТУУ «КПІ», 2016. 173 с.

36. МОЗ України. Державні санітарні правила і норми влаштування, утримання загальноосвітніх навчальних закладів та організації навчально-виховного процесу (ДСанПіН 5.5.2.008-01). – Київ, 2001. 49 с.

37. Д. Величко, А. Паламар. Система моніторингу якості повітря в приміщеннях // Матеріали V Міжнародної студентської науково-технічної конференції «Природничі та гуманітарні науки. Актуальні питання», 28-29 квітня 2022 р. – Т.: ТНТУ, 2022. – С. 138. – (Управління та адміністрування).

38. М. Гандзюк, Є. Желібо, М. Халімовський. Основи охорони праці. – К.: Каравела, 2007. 408 с.

ДОДАТКИ

Обробка команди start у чат-боті

```
@bot.message_handler(commands=['start'])
def handle_start(message):
    user_id = message.from_user.id
    try:
        with open("user_ids.txt", 'r') as file:
            user_ids = file.read().splitlines()
    except FileNotFoundError:
        user_ids = []

    if str(user_id) not in user_ids:
        with open("user_ids.txt", 'a') as file:
            file.write(f"{user_id}\n")

    bot.send_message(message.chat.id, text=f"*Вітаю,
{message.chat.username}* 😊\nЯ - чат-бот Гончарівського закладу ✨
\nРадий бути твоїм особистим помічником у всіх шкільних питаннях!
📖\nНе соромся запитувати, я завжди тут, щоб тобі допомогти!
💬🏠", parse_mode='Markdown')
    send_main_menu(message)
```

Код кнопок головного меню

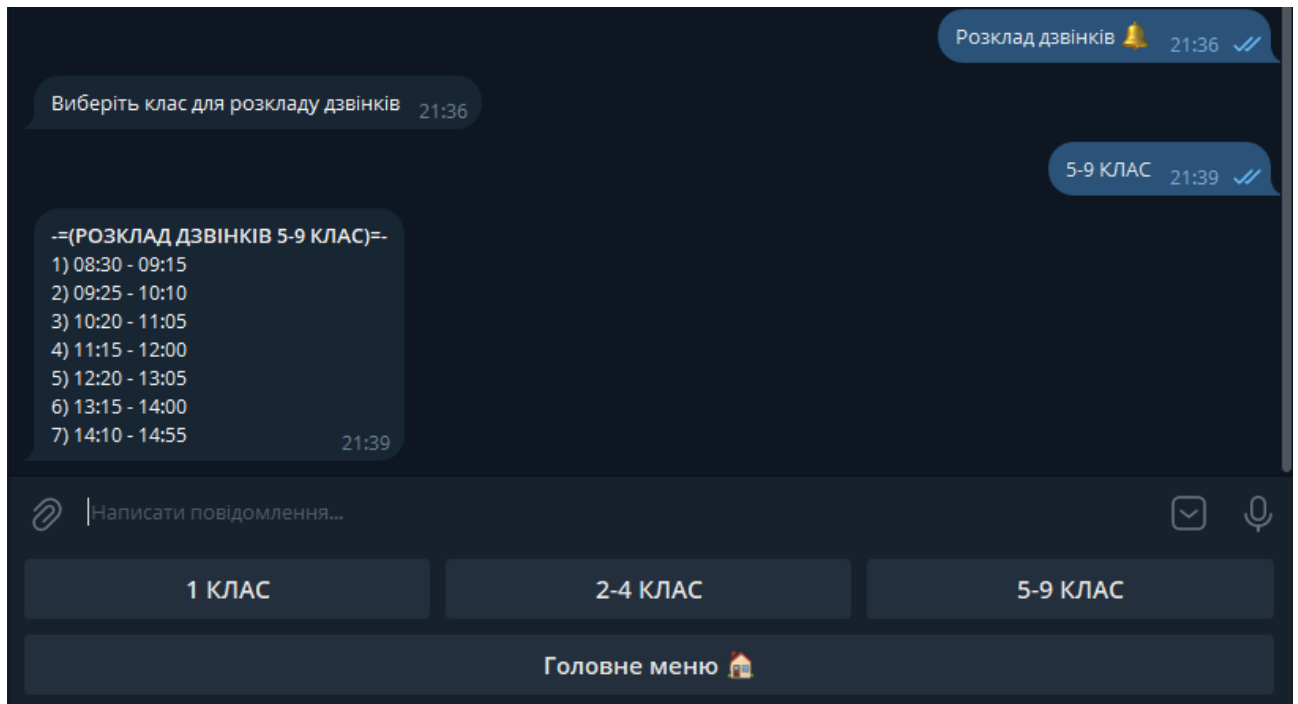
```
def send_main_menu(message):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
    button1 = types.KeyboardButton("Розклад дзвінків 🛎")
    button2 = types.KeyboardButton("Розклад уроків 📅")
    button4 = types.KeyboardButton("Події 🎉")
    button5 = types.KeyboardButton("Погода 🌤")
    if is_admin(message.from_user.id):
        button3 = types.KeyboardButton("Адмін-панель 👤")
    markup.add(button1, button2, button4, button5)
    if is_admin(message.from_user.id):
        markup.add(button3)

    bot.send_message(message.chat.id, text=f"*Оберіть дію  
{message.chat.username}* 😊", reply_markup=markup,
        parse_mode='Markdown')
```

Код для обробки дій користувача

```
@bot.message_handler(content_types=['text'])
def handle_text(message):
    if message.text == "Розклад дзвінків 🛎️":
        send_bell_schedule(bot, message)
    elif message.text == "Розклад уроків 📅":
        send_lesson_schedule(bot, message)
    elif message.text == "Події 🎉":
        send_events(bot, message)
    elif message.text == "Головне меню 🏠":
        send_main_menu(message)
    else:
        bot.send_message(message.chat.id, text="Ця команда не
розпізнана. Можливо, ви мали на увазі щось інше? 🙄♂")
        send_main_menu(message)
```

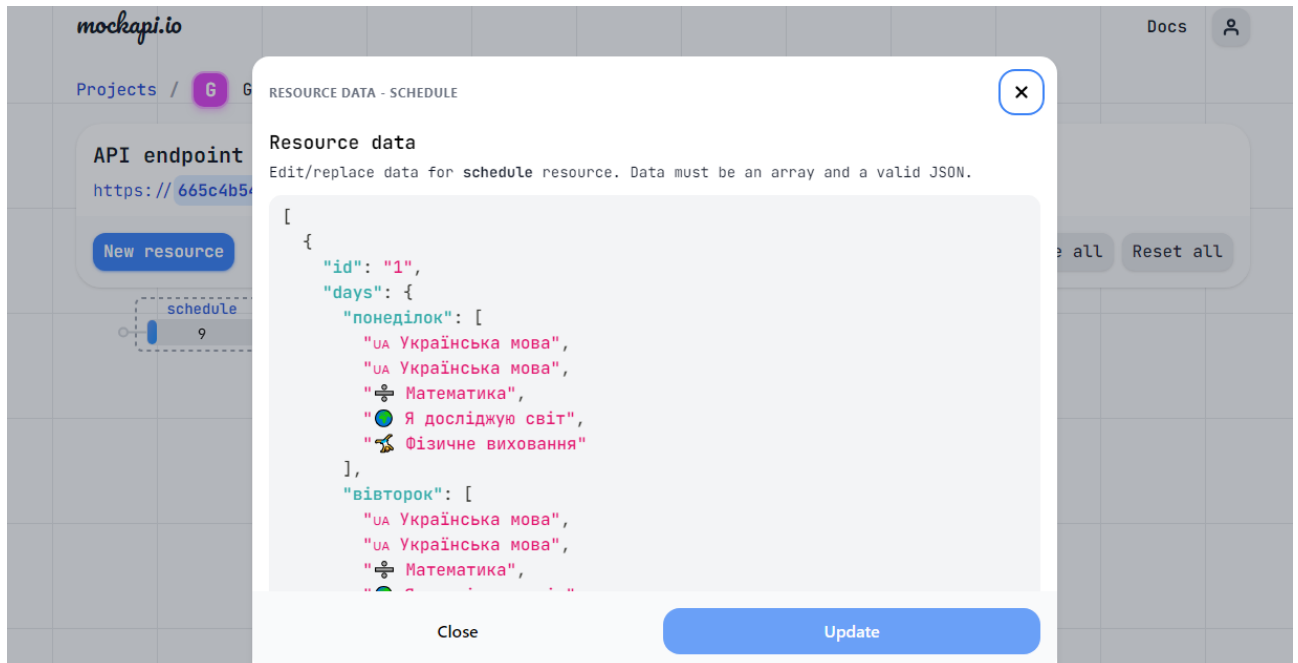
Результат вибору розкладу дзвінків



Код для відправки вибраного розкладу дзвінків

```
def send_bell_schedule_for_class(bot, message, class_number):
    if class_number == "1 КЛАС":
        bot.send_message(message.chat.id,
            "*==(РОЗКЛАД ДЗВІНКІВ 1 КЛАС)--* \n"+
            "1) 08:30 - 09:05\n"+
            "2) 09:15 - 09:50\n"+
            "3) 10:10 - 10:45\n"+
            "4) 10:55 - 11:30\n"+
            "5) 11:40 - 12:15\n",
            parse_mode='Markdown')
    elif class_number == "2-4 КЛАС":
        bot.send_message(message.chat.id,
            "*==(РОЗКЛАД ДЗВІНКІВ 2-4 КЛАС)--* \n"+
            "1) 08:30 - 09:10\n"+
            "2) 09:20 - 10:00\n"+
            "3) 10:20 - 11:00\n"+
            "4) 11:10 - 11:40\n"+
            "5) 11:50 - 12:20\n"+
            "6) 12:30 - 13:10\n",
            parse_mode='Markdown')
    elif class_number == "5-9 КЛАС":
        bot.send_message(message.chat.id,
            "*==(РОЗКЛАД ДЗВІНКІВ 5-9 КЛАС)--* \n"+
            "1) 08:30 - 09:15\n"+
            "2) 09:25 - 10:10\n"+
            "3) 10:20 - 11:05\n"+
            "4) 11:15 - 12:00\n"+
            "5) 12:20 - 13:05\n"+
            "6) 13:15 - 14:00\n"+
            "7) 14:10 - 14:55\n",
            parse_mode='Markdown')
```

Розклад уроків у МоскАРІ



москаpi.io

Projects / 6 6

API endpoint
https://665c4b5...

New resource

schedule
9

Docs

RESOURCE DATA - SCHEDULE

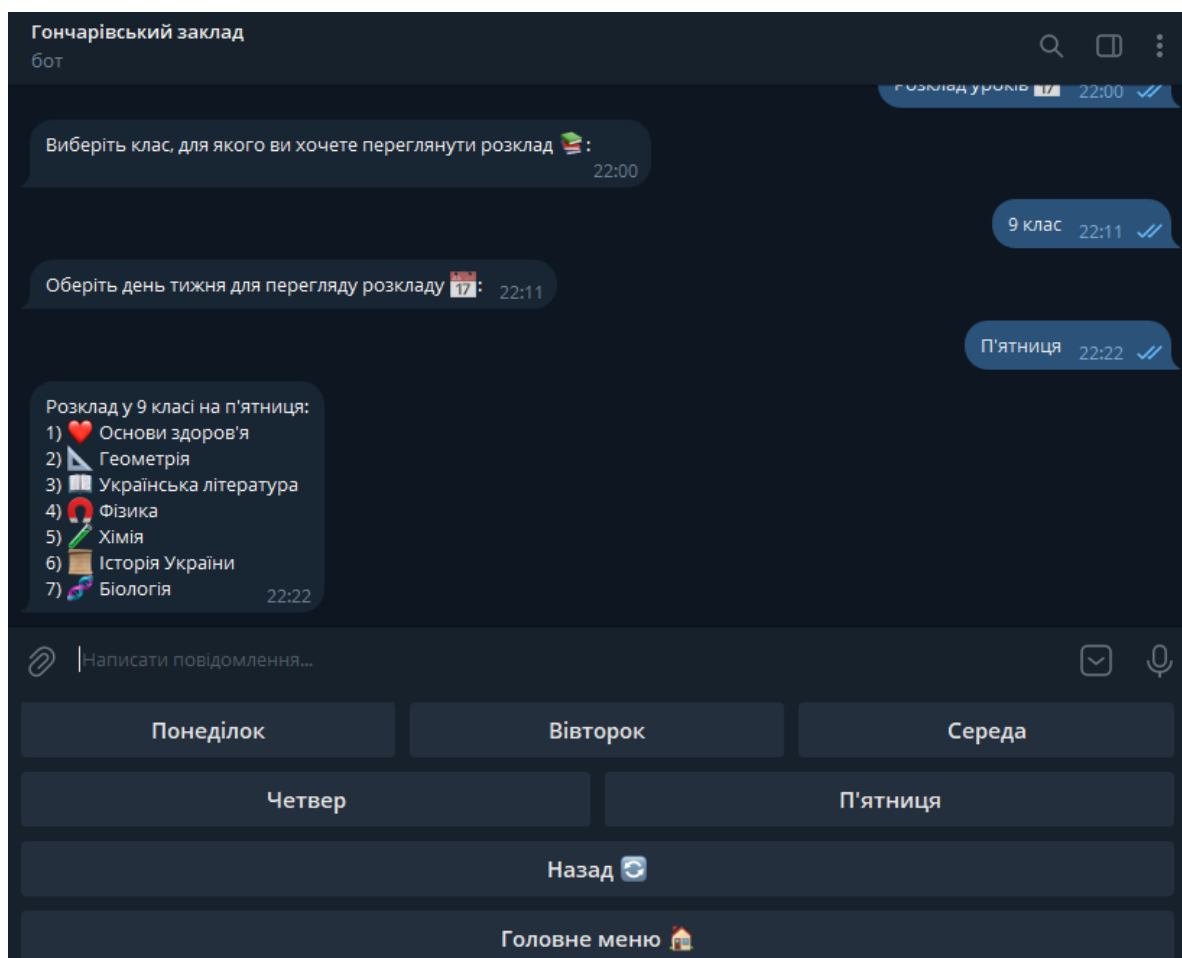
Resource data

Edit/replace data for `schedule` resource. Data must be an array and a valid JSON.

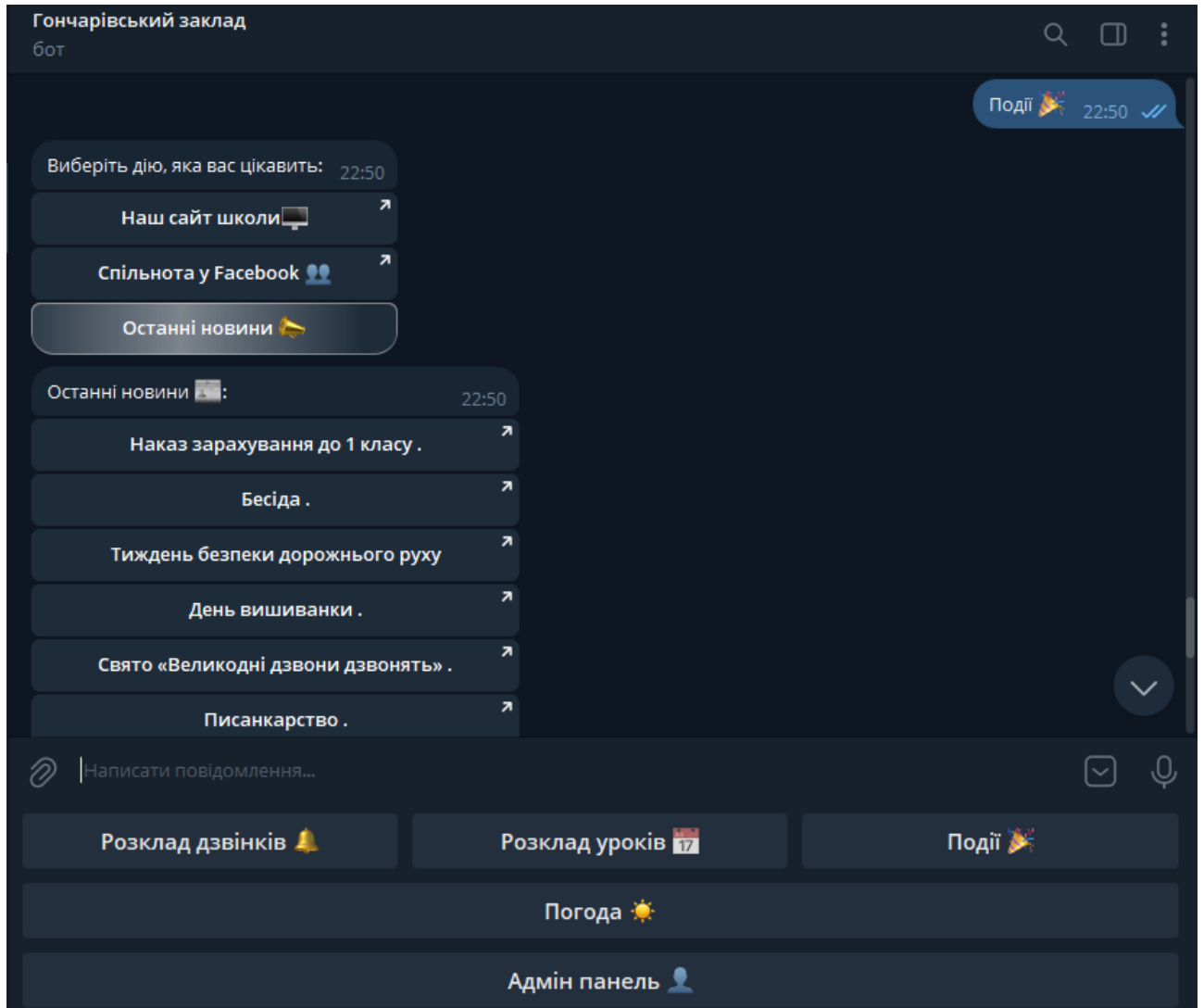
```
[
  {
    "id": "1",
    "days": {
      "понеділок": [
        "UA Українська мова",
        "UA Українська мова",
        "Математика",
        "Я досліджую світ",
        "Фізичне виховання"
      ],
      "вівторок": [
        "UA Українська мова",
        "UA Українська мова",
        "Математика",
        "Я досліджую світ"
      ]
    }
  }
]
```

Close Update

Перегляд розкладу уроків для 9 класу на п'ятницю



Останні новини



Код для Адмін-панелі

```

@bot.message_handler(func=lambda message: message.text ==
"Адмін-панель 👤" and is_admin(message.from_user.id))
def handle_admin_panel(message):
    send_admin_menu(bot, message)
@bot.callback_query_handler(func=lambda call: call.data ==
"edit_lesson_schedule" and is_admin(call.from_user.id))
def handle_edit_lesson_schedule(call):
    edit_lesson_schedule(bot, call.message)
def process_global_message(message):
    text = message.text
    send_global_message(bot, text)
    bot.send_message(message.chat.id, text="Повідомлення успішно
надіслано всім користувачам 📧📧")
@bot.callback_query_handler(
    func=lambda call: call.data == "send_global_message" and
is_admin(call.from_user.id))
def handle_send_global_message(call):
    bot.send_message(call.message.chat.id, text="Введіть ваше
повідомлення для надсилання всім користувачам 📧: ")
    bot.register_next_step_handler(call.message,
process_global_message)
@bot.callback_query_handler(func=lambda call: call.data ==
"send_news")
def handle_send_news(call):
    send_news(bot, call.message)
@bot.callback_query_handler(func=lambda call:
call.data.startswith("class_") and is_admin(call.from_user.id))
def handle_edit_class_schedule(call):
    class_number = str(call.data.split("_")[1])
@bot.callback_query_handler(func=lambda call:
call.data.startswith("day_") and is_admin(call.from_user.id))
def handle_edit_day_schedule(call):
    _, class_number, day = call.data.split("_")
    edit_day_schedule(bot, call.message, str(class_number), day)
@bot.callback_query_handler(func=lambda call:
call.data.startswith("lesson_") and is_admin(call.from_user.id))
def handle_edit_lesson(call):
    _, class_number, day, lesson_index = call.data.split("_")
    bot.send_message(call.message.chat.id, text="Будь ласка,
вказіть нову назву уроку 🗨️: ")
    bot.register_next_step_handler(call.message,
process_new_lesson_name, str(class_number), day,
int(lesson_index))
def process_new_lesson_name(message, class_number, day,
lesson_index):
    new_lesson = message.text
    update_lesson(bot, message, class_number, day, lesson_index,
new_lesson)

```