

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«До захисту допущено»
В.о. завідувача кафедри
Ігор ШЕЛЕХОВ

(підпис)

червня 202_ р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня бакалавр

зі спеціальності 122 Комп'ютерні науки,
освітньо-професійної програми «Інформатика»
на тему: «Веб-додаток для покращення якості освітнього процесу»
здобувача групи ІН – 01 Губко Роман Андрійович

Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання на
відповідне джерело.

(підпис)
) Роман ГУБКО

Керівник,
старший викладач кафедри комп'ютерних наук,
к. ф. -м. н

Анна БАДАЛЯН

(підпис)

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«Затверджую»

В.о. завідувача кафедри

Ігор ШЕЛЕХОВ

(підпис)

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ
на здобуття освітнього ступеня бакалавра

зі спеціальності 122 Комп'ютерні науки, освітньо-професійної програми «Інформатика»
здобувача групи ІН-01 Губко, Роман, Андрійович

1. Тема роботи: «Розробка веб-додатку»
затверджую наказом по СумДУ від «22» квітня 2024 р. № 0414-VI
2. Термін здачі здобувачем кваліфікаційної роботи до 01 червня 2024 року
3. Вхідні дані до кваліфікаційної роботи
4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)
1) Аналіз проблеми предметної області, постановка й формування завдань дослідження.
2) Огляд технологій, що використовуються для розробки додатку. 3) Розробка інтелектуальної системи. 4) Аналіз результатів.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання «06» травня 2024 р.

Завдання прийняв до виконання _____ Керівник _____
(підпис) (підпис)
))

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання	Примітка
1	<i>Аналіз проблеми предметної області, постановка й формування завдань дослідження</i>	06.05.24-12.05.24	
2	<i>Огляд технологій,</i>	12.05.24-17.05.24	
3	<i>Розробка інтелектуальної системи</i>	17.05.24-21.05.24	
4	<i>Аналіз отриманих результатів</i>	21.05.24-25.05.24	
5	<i>Оформлення пояснювальної записки до кваліфікаційної роботи</i>	25.05.24-31.05.24	

Здобувач вищої освіти _____
(підпис)

Керівник _____
(підпис)

АНОТАЦІЯ

Записка: 21 стр., 04 рис., 1 додаток, 20 використаних джерел.

Обґрунтування актуальності теми роботи – Тема кваліфікаційної роботи є актуальною, оскільки присвячена розробці веб-додатку для навчання та освіти, особливо під час дистанційного навчання

Об’єкт дослідження — розробка веб-додатку для покращення якості освіти

Мета роботи — розробити додаток для навчання

Методи дослідження — методи штучного інтелекту: машинне навчання, обробка природної мови, аналіз даних.

Результати — розроблений телеграм бот став потужним інструментом для оптимізації взаємодії між вчителями та учнями. Його функціонал дозволяє не лише ефективно передавати навчальний матеріал, але й надає вчителям можливість використовувати інтерактивні та аналітичні інструменти для поліпшення процесу навчання.

Зміст

Вступ	5
1 ІНФОРМАЦІЙНИЙ ОГЛЯД	6
1.1 Інформаційний огляд розробки Telegram-бота для допомоги вчителям та учням у знаходженні інформації за допомогою штучного інтелекту	6
1.2 Постановка задачі	7
1.3 Розбір конкурентів	8
2 ВИБІР МЕТОДІВ РІШЕННЯ ЗАДАЧІ	12
2.1. Прототипування веб-сайта за допомогою PyCharm	12
3 ПРОГРАМНА РЕАЛІЗАЦІЯ	16
4 ВИСНОВКИ	19
СПИСОК ЛІТЕРАТУРИ	20
Додаток А	21

Вступ

Обґрунтування вибору теми роботи: Розробка Telegram-бота з використанням штучного інтелекту для підтримки освітнього процесу є актуальною через потребу в покращенні комунікації між вчителями та учнями, ефективність інструментів штучного інтелекту для індивідуалізації навчання, популярність та доступність Telegram як платформи та потенціал для інновацій та розвитку в освіті.

Актуальність: цей проект використовує популярний штучний інтелект

Об'єкт дослідження: Процес розробки веб-додатку для навчання

Предмет дослідження: розробка додатку в Pycharm

Гіпотеза: Впровадження Telegram-бота, який використовує штучний інтелект для підтримки освітнього процесу, сприятиме підвищенню якості навчання та збільшенню доступності освітніх ресурсів для учасників освітнього процесу

Новизна: Новизна даної роботи полягає в поєднанні двох ключових компонентів - використанні штучного інтелекту (ШІ) та платформи Telegram для покращення освітнього процесу. Цей проект виходить за межі звичайних інструментів навчання та комунікації, пропонуючи інтелектуальний та інтерактивний підхід до взаємодії між вчителями та учнями

Структура. Дане робота складається зі вступу, аналітичного огляду, постановки задачі, вибір методу розв'язання поставленої задачі, опису програмного забезпечення інформаційної системи, висновків, списку використаних джерел та додатків

ВЕБ-ДОДАТОК, ШТУЧНИЙ ІНТЕЛЕКТ, РУСНА

1 ІНФОРМАЦІЙНИЙ ОГЛЯД

1.1 Інформаційний огляд розробки Telegram-бота для допомоги вчителям та учням у знаходженні інформації за допомогою штучного інтелекту

У сучасному світі, де інформація стає все більш доступною, вміння її шукати, обробляти та аналізувати стає ключовою навичкою для успіху в будь-якій сфері. Це особливо актуально для вчителів та учнів, адже саме від їхньої здатності ефективно працювати з інформацією залежить якість освітнього процесу. Роль штучного інтелекту:

Штучний інтелект (ШІ) володіє значним потенціалом для покращення навичок пошуку та обробки інформації. ШІ-системи здатні швидко й ефективно опрацьовувати великі обсяги даних, виявляти приховані закономірності та тенденції, а також генерувати нові знання. Це може допомогти вчителям та учням:

Швидше та легше знаходити необхідну інформацію: ШІ-системи можуть шукати інформацію за різними критеріями, включаючи ключові слова, контекст та тип джерела. Це дозволяє значно скоротити час, який витрачається на пошук інформації, та зосередитися на її аналізі та розумінні.

Краще розуміти знайдену інформацію: ШІ-системи можуть допомогти з аналізом інформації, виявляючи зв'язки між різними даними, візуалізуючи складні концепції та надаючи резюме ключових моментів.

Створювати нові знання: ШІ-системи можуть генерувати нові ідеї, гіпотези та теорії на основі аналізу наявних даних. Це може допомогти вчителям та учням у дослідницькій роботі та творчому мисленні.

Опис дослідження:

Дане дослідження зосереджене на розробці Telegram-бота, який використовує ШІ для допомоги вчителям та учням у пошуку та обробці інформації. Бот буде оснащений наступними можливостями:

Інтелектуальний пошук: Бот зможе шукати інформацію в Інтернеті за допомогою ШІ, використовуючи різні критерії, такі як ключові слова, контекст та тип джерела.

Аналіз та обробка інформації: Бот зможе аналізувати знайдену інформацію, виявляючи зв'язки між даними, візуалізуючи складні концепції та надаючи резюме ключових моментів.

Інтерактивний інтерфейс: Бот буде мати зручний інтерфейс для взаємодії з користувачами, що дозволить їм легко ставити запитання, отримувати інформацію та керувати процесом пошуку.

Очікувані результати:

В результаті дослідження очікується отримати Telegram-бота, який стане цінним інструментом для вчителів та учнів, допомагаючи їм:

- Економити час та зусилля на пошук інформації
- Краще розуміти знайдену інформацію
- Розвивати навички критичного мислення та аналізу
- Підвищувати ефективність навчання та самоосвіти

Методи дослідження:

Для досягнення мети дослідження будуть використовуватися такі методи:

- Методи штучного інтелекту: машинне навчання, обробка природної мови, аналіз даних.
- Методи дослідження інтерфейсів користувача: тестування користувачів, інтерв'ю, аналіз зворотного зв'язку.

Етапи дослідження:

Дослідження буде проводитися в кілька етапів: • Розробка алгоритму пошуку інформації в Інтернеті за допомогою ШІ.

- Розробка алгоритму аналізу та обробки інформації, знайденої в Інтернеті.
- Розробка інтерфейсу для взаємодії з ботом.
- Тестування та вдосконалення бота.
- Оцінка ефективності бота в реальних умовах.

1.2 Постановка задачі

Опис проблематики: Зростання обсягів інформації у сучасному світі ставить перед вчителями та учнями виклик знаходження швидкого та ефективного доступу до релевантної інформації для підтримки навчання та викладання.

Мета проекту: Створення Telegram-бота, який використовує штучний інтелект для полегшення процесу пошуку та надання корисної інформації для вчителів та учнів.

Основні функціональні можливості: Пошук інформації: Бот використовує алгоритми машинного навчання для ефективного пошуку релевантної інформації з різних джерел та надає її у зручному форматі.

Технічні аспекти розробки: Мови програмування:

Використання мов програмування, таких як Python або Node.js, для створення бота та реалізації алгоритмів штучного інтелекту. - API Telegram:

Взаємодія з Telegram API для обробки повідомлень та надсилання відповідей.

Бібліотеки машинного навчання: Використання бібліотек, таких як TensorFlow або PyTorch, для реалізації алгоритмів пошуку та аналізу інформації.

Безпека та конфіденційність: Забезпечення заходів безпеки для захисту особистої інформації користувачів та дотримання принципів конфіденційності.

Перспективи розвитку: Можливість інтеграції нових функцій та розширення функціональності бота відповідно до потреб користувачів та змін у сфері освіти.

Висновки: Очікується, що розробка цього Telegram-бота принесе значний внесок у полегшення доступу до інформації для вчителів та учнів, сприяючи вдосконаленню процесу навчання та викладання.

1.3 Розбір конурентів

Детальний аналіз:

1. @QuizBot: Переваги:

- Широкий спектр тестів з різних предметів, що робить його універсальним інструментом для перевірки знань.
- Надання результатів та пояснень до кожного питання, що допомагає користувачам навчатися на своїх помилках та покращувати розуміння матеріалу.
- Можливість самостійного тестування, що робить його зручним для використання в будь-який час і в будь-якому місці.

Недоліки:

- Відсутність персоналізованого навчання та адаптивних алгоритмів, що може обмежувати його ефективність для деяких користувачів.
- Не пропонує глибокого вивчення тем, зосереджуючись лише на перевірці знань.
- Може бути не таким цікавим для користувачів, які шукають більш інтерактивний та захоплюючий досвід навчання.

2. @MathBot: Переваги:

- Покрокове пояснення розв'язання задач, що робить його ідеальним для тих, хто хоче зрозуміти математичні концепції.
- Можливість вивчення нових математичних тем, що робить його корисним інструментом для додаткового навчання.
- Допомога у підготовці до іспитів та тестів, завдяки наданню практики з розв'язання задач. Недоліки:
 - Зосередження лише на математиці, що обмежує його сферу застосування.
 - Відсутність інтерактивних елементів та ігрофікації, що може зробити навчання нудним для деяких користувачів.

- Може бути складним для початківців, які потребують більш базових пояснень.

3. @StudygramBot: Переваги:

- Допомога в організації навчального процесу, що робить його незамінним інструментом для учнів та студентів.

- Створення розкладів занять, встановлення нагадувань про домашні завдання та тестування, що допомагає користувачам краще керувати своїм часом.

- Доступ до корисних навчальних матеріалів, таких як посібники та онлайн-курси, що доповнює формальну освіту. Недоліки:
 - Не пропонує персоналізованого навчання та адаптивного контенту.

- Відсутність інтерактивних елементів та практики, що може обмежувати його ефективність навчання.

- Покладається на самодисципліну користувачів, що може бути складно для деяких людей.

4. @LearnEnglishBot:

Переваги:

- Різноманітні вправи з англійської мови, що охоплюють граматику, лексику, читання, аудіювання та письмо.

- Використання аудіо- та відеоматеріалів, що робить навчання більш цікавим та захоплюючим.

- Можливість спілкування з носіями мови, що допомагає покращити навички усного мовлення та подолати мовний бар'єр.

Недоліки:

- Відсутність персоналізації та адаптивного навчання, що може обмежувати його ефективність для деяких користувачів.

- Не пропонує глибокого вивчення англійської мови, зосереджуючись лише на базових навичках.

- Може бути складним для початківців, які потребують більш структурованого підходу до навчання.

5. @ScienceDailyBot: Переваги:

- Надає доступ до свіжих наукових новин та цікавих фактів з різних галузей науки, що робить його чудовим джерелом інформації для допитливих людей.

- Допомогає користувачам залишатися в курсі останніх наукових відкриттів та досліджень.

- Забезпечує щоденну дозу цікавої та корисної інформації, що може розширити кругозір та стимулювати інтерес до науки.

Недоліки:

- Не пропонує глибокого вивчення наукових тем, зосереджуючись лише на коротких новинах та фактах.

Що я зроблю, щоб розширити свій проект:

1. Додаткові функції:

Додати функцію перекладу, щоб користувачі могли спілкуватися одне з одним та отримувати доступ до інформації різними мовами.

Створити систему нагадувань, щоб допомагати користувачам не пропускати важливі події та дедлайни.

Розробити додаткові матеріали для вивчення різних предметів, щоб задовольнити потреби та інтереси ширшого кола користувачів.

2. Персоналізація:

Надати користувачам можливість налаштовувати свої налаштування, щоб вони могли персоналізувати свій досвід роботи з ботом.

Дозволити користувачам обирати теми, які їм цікаві, щоб отримувати більш релевантну інформацію та рекомендації.

Використовувати машинне навчання, щоб пропонувати користувачам персоналізовані навчальні плани та ресурси.

3. Створення спільноти користувачів:

Створити онлайн-форум або групу в соціальних мережах, де користувачі зможуть спілкуватися одне з одним.

Запропонувати користувачам можливість ділитися своїми знаннями та досвідом одне з одним.

Проводити онлайн-заходи та зустрічі, щоб об'єднати спільноту.

4. Інтеграція з іншими платформами:

Інтегрувати бота з популярними освітніми платформами, щоб користувачі могли отримувати доступ до ширшого кола ресурсів.

Додати можливість ділитися інформацією та контентом з бота в інших соціальних мережах та платформах.

Створити API, щоб інші розробники могли інтегрувати бота у свої програми та послуги.

5. Партнерство з освітніми установами:

Співпрацювати з університетами, школами та іншими освітніми організаціями для розробки нових навчальних програм та ресурсів.

Запропонувати бота як навчальний інструмент для студентів та викладачів. Взяти участь у освітніх заходах та конференціях, щоб підвищити обізнаність про бота.

6. Мультимовність:

Перекласти інтерфейс та контент бота на різні мови.

Додати підтримку голосового вводу та виводу для різних мов.

Створити спільноти користувачів для кожної мови, щоб вони могли спілкуватися одне з одним рідною мовою.

Впровадження цих ідей допоможе мені розширити свій проект, зробити його більш корисним та привабливим для ширшого кола користувачів, а також сприятиме створенню динамічної та зацікавленої спільноти навколо нього.

2 ВИБІР МЕТОДІВ РІШЕННЯ ЗАДАЧІ

2.1. Прототипування веб-сайта за допомогою PyCharm

PyCharm — це високоефективне інтегроване середовище розробки (IDE) для мови програмування Python, яке надає численні переваги, зроблючи процес розробки більш продуктивним та зручним.

Зручний інтерфейс: PyCharm приваблює своїм інтуїтивним та зручним інтерфейсом, що спрощує навігацію та взаємодію з проектами. Він дозволяє розробникам швидко зорієнтуватися у коді та легко переходити між файлами та функціями.

Вбудовані інструменти відлагодження: PyCharm обладнаний потужними інструментами відлагодження, які допомагають виявляти та виправляти помилки у коді з мінімальними зусиллями. Це дозволяє розробникам швидко знаходити та виправляти проблеми, що збільшує ефективність роботи.

Автоматичне доповнення коду: PyCharm надає широкі можливості автоматичного доповнення коду, що робить процес написання та підтримки коду швидшим та зручнішим. Це зменшує кількість помилок та підвищує швидкість розробки.

Підтримка систем керування версіями: PyCharm інтегрується з популярними системами керування версіями, такими як Git, що дає змогу розробникам легко відстежувати та керувати змінами у своїх проектах. Ця функція сприяє спільній роботі команд, дозволяючи їм ефективно співпрацювати та здійснювати колективне управління кодом. Завдяки можливості перегляду історії змін, відмінностей між версіями та гілками, розробники можуть легко координувати свою роботу та впроваджувати нові функції безпечно та без конфліктів.

Автоматичне створення віртуальних середовищ: PyCharm дозволяє створювати та керувати віртуальними середовищами Python з легкістю. Це забезпечує ізольоване середовище для кожного проекту, що дозволяє уникнути конфліктів між різними версіями бібліотек та різними залежностями. Завдяки цьому, розробники можуть ефективно керувати залежностями своїх проектів та гарантувати стабільну роботу додатків навіть у складних середовищах.

Підтримка різних фреймворків та бібліотек: PyCharm надає розширену підтримку для різних фреймворків та бібліотек Python, таких як Django, Flask, NumPy, і багатьох інших. Це робить його універсальним інструментом для розробки на Python, охоплюючи різноманітні проекти від веб-додатків до наукових досліджень. Розширена підтримка дозволяє розробникам працювати з

будь-яким фреймворком або бібліотекою, що вони обирають, не втрачаючи при цьому доступу до всіх можливостей PyCharm.

Загалом, PyCharm є потужним інструментом для розробки на Python, який сприяє підвищенню продуктивності та комфорту програмістів у будьякому розмірі та складності проєктів.

Є багато інших інтегрованих середовищ розробки (IDE) для мови програмування Python, які можна використовувати залежно від ваших вподобань та потреб. Деякі з альтернатив до PyCharm включають:

Розширене описування IDE для Python:

1. Visual Studio Code (VSCode): Переваги:

- Легка та швидка робота.
- Велика кількість розширень для підтримки Python, включаючи linting, автозаповнення коду, інтеграцію з Git та багато іншого.

- Безкоштовний та відкритий код.

- Активна спільнота користувачів та розробників. Недоліки:

- Може здатися складним для початківців.

- Деякі розширення можуть споживати багато ресурсів системи.

2. Atom:

Переваги:

- Розширюваний редактор коду з широким набором пакетів для підтримки Python.

- Інтуїтивно зрозумілий інтерфейс користувача.

- Безкоштовний та відкритий код.

Недоліки:

- Може бути повільнішим, ніж інші редактори коду. • Не так активно підтримується, як VSCode.

3. Sublime Text:

Переваги:

- Мінімалістичний та швидкий інтерфейс.

- Широкий набір плагінів для роботи з Python.

- Безкоштовна пробна версія, платна ліцензія. Недоліки:

- Не така розширювана, як VSCode або Atom.

- Відсутність деяких функцій, які є в інших IDE.

4. Jupyter Notebooks:

Переваги:

- Інтерактивне середовище для виконання коду в клітинках.

- Ідеально підходить для аналізу даних та наукових досліджень.

- Легко ділитися ноутбуками з іншими.

Недоліки:

- Не так добре підходить для розробки великих програмних проектів.
- Може бути складним для початківців.

5. Spyder:

Переваги:

- IDE, спеціально розроблений для наукових обчислень та аналізу даних.
- Вбудований консольний інтерфейс IPython.
- Широкий набір інструментів для роботи з даними. Недоліки:
- Може здатися складним для початківців.
- Не так добре підходить для загальної розробки Python.

6. Thonny:

Переваги:

- Легкий IDE, спроектований спеціально для початківців у програмуванні на Python.
- Візуальне середовище для виконання коду.
- Вбудовані вправи та підказки. Недоліки:
- Не так потужний, як інші IDE. • Не підходить для складних програмних проектів.

7. Komodo IDE:

Переваги:

- Комплексне середовище розробки, яке підтримує Python та багато інших мов програмування.
- Широкий набір функцій для розробки та налагодження.
- Безкоштовна пробна версія, платна ліцензія. Недоліки:
- Може бути складним для початківців. • Досить ресурсомісткий.

8. Eclipse з PyDev Plugin: Переваги:

- Потужна та розширювана IDE.
- Широкий набір функцій для розробки Python.
- Безкоштовний та відкритий код. Недоліки:
- Може здатися складним для початківців.
- Інтерфейс користувача може бути не таким інтуїтивно зрозумілим, як у інших IDE.

Вибір IDE:

Найкращий IDE для вас буде залежати від ваших вподобань, стилю роботи та конкретних потреб у розробці.

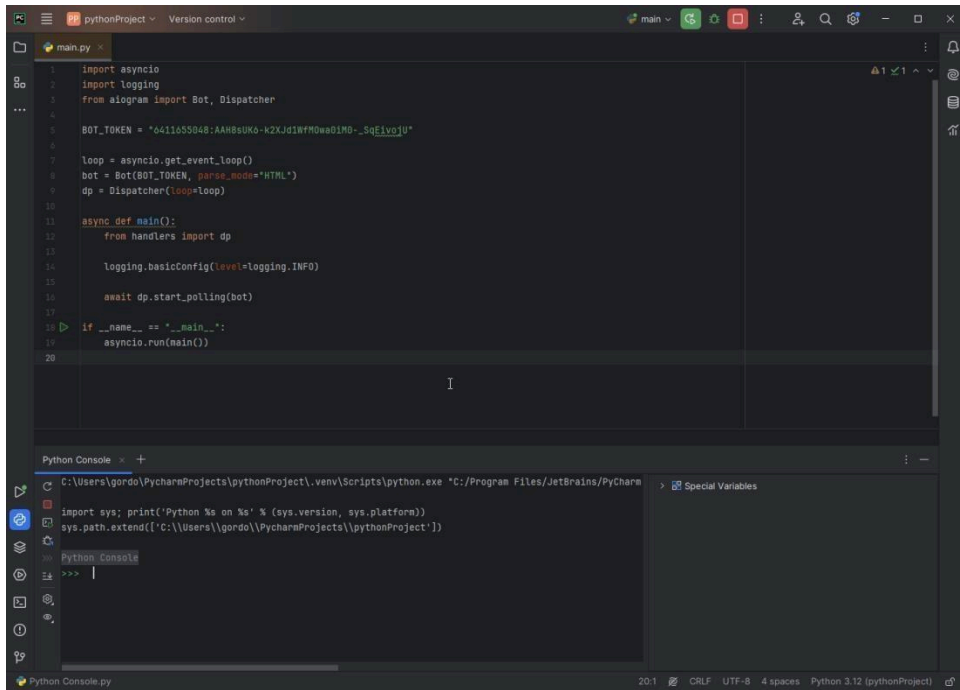
Для початківців: Thonny або Visual Studio Code з розширеннями для Python.

Для аналізу даних та наукових досліджень: Jupyter Notebooks або Spyder.
Для загальної розробки Python: Visual Studio Code, Sublime Text, або PyCharm.

Для складних програмних проєктів: PyCharm або Komodo IDE.

Важливо спробувати кілька різних IDE, щоб знайти ту, яка вам найбільше підходить.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ



```
1 import asyncio
2 import logging
3 from aiogram import Bot, Dispatcher
4
5 BOT_TOKEN = "6411655048:AAH8sUK6-k2XJd1WfMwaD1M6-SqE1vo1U"
6
7 loop = asyncio.get_event_loop()
8 bot = Bot(BOT_TOKEN, parse_mode="HTML")
9 dp = Dispatcher(loop=loop)
10
11
12 @asyncio.coroutine
13 from handlers import dp
14
15 logging.basicConfig(level=logging.INFO)
16
17 await dp.start_polling(bot)
18
19 if __name__ == "__main__":
20     asyncio.run(main())
```

Python Console

```
C:\Users\gordo\PycharmProjects\pythonProject\venv\Scripts\python.exe "C:/Program Files/JetBrains/PyCharm
import sys; print('Python %s on %s' % (sys.version, sys.platform))
sys.path.extend(['C:\\Users\\gordo\\PycharmProjects\\pythonProject'])
>>> |
```

Рисунок 1 - Код в програмі

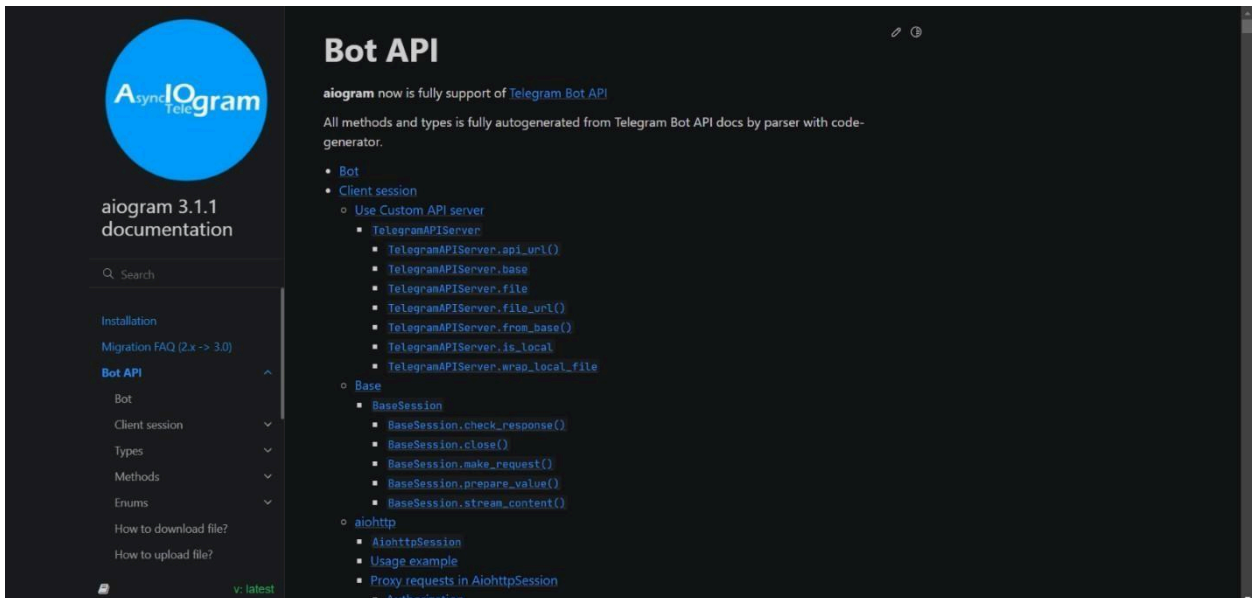


Рисунок 2. Aiogram Bot API для розробки телеграм бота

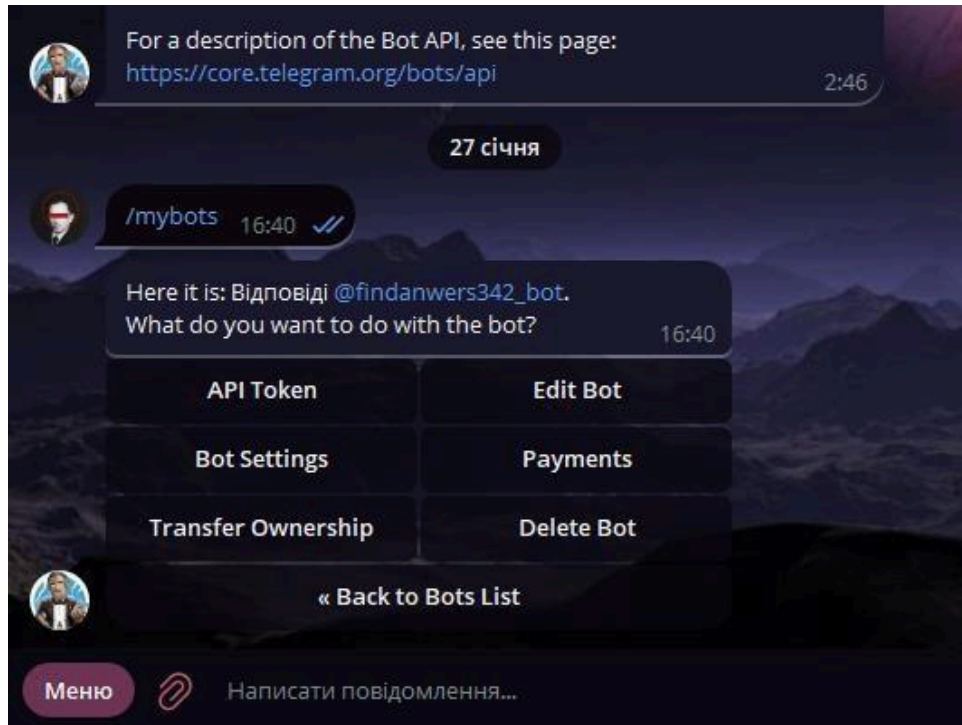


Рисунок 3 - Головний бот BotFather потрібний для створення ботів в телеграмі

```
import asyncio
import logging
from aiogram import Bot, Dispatcher
```

```
BOT_TOKEN = "6411655048:AAH8sUK6-k2XJd1WfMOwa0iM0-_SqEivojU"
```

```
loop = asyncio.get_event_loop()
bot = Bot(BOT_TOKEN,
        parse_mode="HTML")
dp = Dispatcher(loop=loop)
```

```
async def main():
    from handlers import dp
```

```
    logging.basicConfig(level=logging.INFO)
    await
    dp.start_polling(bot)
```

```
if __name__ == "__main__": asyncio.run(main())
```

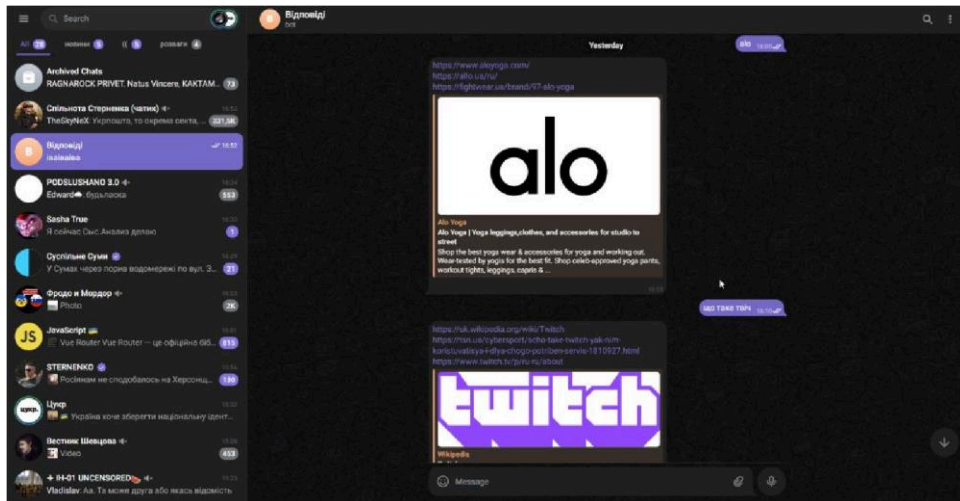


Рисунок 4 - Тестування прототипу у браузері

4 ВИСНОВКИ

У ході виробничої практики був обраний інструмент для первинної розробки телеграм бота, що успішно відповідає поставленим завданням з надання допомоги вчителям. Вибір даного інструменту був обумовлений його зручністю у використанні та можливістю наочного представлення інформації для вчителів.

Розроблений телеграм бот став потужним інструментом для оптимізації взаємодії між вчителями та учнями. Його функціонал дозволяє не лише ефективно передавати навчальний матеріал, але й надає вчителям можливість використовувати інтерактивні та аналітичні інструменти для поліпшення процесу навчання.

Наочність та простота використання розробленого інструменту роблять його дієвим засобом для підтримки вчителів у їхній повсякденній роботі. Відзначається можливість розширення функціоналу та інтеграції з іншими платформами для максимальної зручності користувачів.

Загальний висновок полягає в тому, що обраний інструмент для розробки телеграм бота успішно відповідає вимогам та очікуванням, що виникли під час виробничої практики, і має великий потенціал для покращення процесів навчання та взаємодії між вчителями та учнями.

СПИСОК ЛІТЕРАТУРИ

1. Create a Telegram Bot using Python [Create a Telegram Bot using Python](#)
2. Head-First Python, 2nd edition Paul Barry (*O'Reilly, 2016*)
3. Telegram Bot API
4. python-telegram-bot. Github
5. Python Data Science Handbook. Essential Tools for Working with Data
6. Fluent Python by Luciano Ramalho
7. Python for Everybody Specialization by University of Michigan
8. Official Python Documentation: <https://docs.python.org/>
9. Real Python Tutorials: <https://realpython.com/>
10. Automate the Boring Stuff with Python by Al Sweigart
11. "Python Cookbook" by David Beazley and Brian K. Jones (O'Reilly Media, 2013)
12. "Natural Language Processing with Python" by Steven Bird, Ewan Klein, and Edward Loper (O'Reilly Media, 2009)
13. "Data Science for Business" by Foster Provost and Tom Fawcett (O'Reilly Media, 2013)
14. "Deep Learning" by Ian Goodfellow, Yoshua Bengio, and Aaron Courville (MIT Press, 2016)
15. "Machine Learning Yearning" by Andrew Ng (Self-published, 2018)
16. "Introduction to Machine Learning with Python" by Andreas C. Müller and Sarah Guido (O'Reilly Media, 2016)
17. "Pattern Recognition and Machine Learning" by Christopher M. Bishop (Springer, 2006)
18. "Artificial Intelligence: A Modern Approach" by Stuart Russell and Peter Norvig (Pearson, 2016)
19. "Reinforcement Learning: An Introduction" by Richard S. Sutton and Andrew G. Barto (MIT Press, 2018)
20. "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" by Aurélien Géron (O'Reilly Media, 2019)

Додаток А

```
import asyncio import logging from aiogram import Bot, Dispatcher
```

```
BOT_TOKEN = "6411655048:AAH8sUK6-k2XJd1WfMOwa0iM0-_SqEivojU"
```

```
loop = asyncio.get_event_loop() bot = Bot(BOT_TOKEN, parse_mode="HTML")  
dp = Dispatcher(loop=loop)
```

```
async def main(): from
```

```
handlers import dp
```

```
logging.basicConfig(level=logging.INFO)
```

```
await dp.start_polling(bot)
```

```
if __name__ == "__main__": asyncio.run(main())
```