

В.В. Городецька, магістрант
В.Л. Левківський, ст. викладач
А.Ю. Левченко, к.т.н., ст. викладач
Державний університет «Житомирська політехніка»

Розробка Telegram-бота для перевірки рівня знань претендента на заміщення вакантної посади

Використання сучасних інформаційних технологій створює нові можливості оперативного отримання інформації. Вчасне отримання інформації на сьогодні є актуальним як ніколи, в цьому можуть допомогти чат-боти. Популярність чат-ботів росте з кожним роком. Їх використовують у різних сферах життя для тестування, реклами, опитування, попереднього інтерв'ювання претендента на роботу. Також з метою раціонального використання часу можна застосувати чат-боти. Також вони завжди на зв'язку 24/7. Метою роботи є програмна реалізація Telegram-бота для проведення попереднього тестування та перевірки рівня знань і навичок претендента на заміщення вакантної посади в організаціях різної сфери діяльності. Програмний продукт реалізовано за допомогою мови програмування Python. Як сховище даних використовується документоорієнтована система керування базами даних з відкритим вихідним кодом MongoDB. У базі даних зберігається інформація для тестування, а також результати пройдених тестів. У роботі було використано методи об'єктноорієнтованого проєктування, теорії документоорієнтованих баз даних. Результатом реалізації поставленого завдання є програмний продукт у вигляді Telegram-бота. Розроблений бот може використовуватися під час проведення співбесіди, для тестування претендента на посаду. За результатами тестування менеджер по набору персоналу в подальшому зможе зробити попередні висновки. Таке тестування дозволить оцінити рівень знань претендента, щоб отримати бачення про його рівень кваліфікації.

Ключові слова: чат-бот; месенджер; дані; Python; MongoDB; штучний інтелект; співбесіда.

Актуальність теми. У сучасному світі мобільні пристрої розвинені як величезна інфраструктура, тому є великий попит на мобільні додатки. Особливо велику популярність мають соціальні мережі та месенджери. Одним із найпопулярніших таких месенджерів є Telegram, який віддалено нагадує соціальну мережу, завдяки своїй платформі. Тому раціональним лишається бути у тренді, користуючись новинками першими. Одним із головних плюсів для бізнесу та організаційної роботи в Telegram є так звані боти. Більшість сучасних компаній, організацій та онлайн-магазинів додали у свою діяльність Telegram-бота для вирішення певних завдань, таких, як знайти ту чи іншу інформацію, або зробити замовлення. Актуальність теми ботів в наш час є настільки велика, що генеральні директори великих ІТ-компаній у світі вже запевнили своїх користувачів, що найближчим часом в їхніх продуктах також з'явиться можливість користування так званими ботами.

Аналіз останніх досліджень та публікацій, на які спираються автори. Існує безліч додатків, які призначені для імітації діалогу, але здебільшого знання діалогового бота зберігаються в базі даних, створеній розробниками. Дуже часто з цією метою використовують документоорієнтовані бази даних. Стаття [1] присвячена проєктуванню та розробці власної документоорієнтованої системи керування базами даних. У ній описано компоненти сучасних систем керування, їх трирівнева архітектура, наведено загальну блок-схему роботи системи.

Але на сьогодні існує досить багато науковців, які досліджують ідею створення чат-ботів зі штучним інтелектом. У [2] описується впровадження системи рекомендацій, здатної розробляти адаптивні туристичні маршрути за допомогою чат-ботів. Автори С.През, М.Лалл, С.Синха [3] описали дизайн та розробку інтелектуального чат-бота з розпізнаванням голосу. У статті представлено демонстраційний зразок технології для перевірки запропонованої структури, необхідної для підтримки такого робота. Розмовний штучний інтелект набуває все більшого значення, оскільки він забезпечує зручний інтерфейс взаємодії між людьми та комп'ютерами через свій перспективний потенціал і привабливу комерційну цінність. Це активна дослідницька область, що становить значний інтерес і описана у [4]. Існує безліч додатків, які використовують людську подоби і призначені для імітації людського діалогу. Однак дуже небагато дослідників досліджували ідею створення чат-бота зі штучним характером і особистістю. У [5] описується підхід до ідеї виокремлення найважливіших фактів у текстах, що описують життя (зокрема і особистість) історичного діяча, для побудови діалогового агента, для використання в навчальних закладах.

Боти використовуються в різних соціальних мережах. У [6] описується, як реалізувати чат-бот у соціальній мережі «Twitter» для розваги та вірусної реклами з використанням бази даних та простого алгоритму. Успішна реалізація чат-бота не дозволяє людям класифікувати його як СПАМ. У результаті

чат-бот отримав обліковий запис у Twitter (@DonPlaticador), який працює без втручання людини і з кожним днем заробляє все більше передплатників. У [7] було розроблено діалоговий бот для дослідження соціальних процесів у суспільстві. Дослідження того, як люди обробляють висловлювання, коли вони фактично залучені до взаємодії, були вкрай рідкісні.

Відомо, що люди з неврологічними захворюваннями, наприклад, хвороба Паркінсона, відчувають труднощі з мовою та спілкуванням. У [8] подано початкові випробування штучного розмовного агента на ім'я Харлі. Він працює на смартфоні і може спілкуватися з користувачем на різні теми. Опис програми та зразок діалогового вікна ілюструють різні ролі, які чат-боти можуть виконувати в управлінні неврологічними станами.

У [9] авторами створено програмний продукт з використанням методів штучного інтелекту для пошуку вільного місця на парковці. Використовуючи чат-бот, можна здійснювати пошук вільних місць, витрачаючи при цьому мінімум часу. Автори статті [10] демонструють систему оптимізації процесу паркування, застосовуючи Telegram-бот для надання інформації користувачеві про вільні паркомісця.

Метою дослідження є створення мобільного чат-бота для оцінки попереднього рівня знань перед проходженням співбесіди. Для досягнення поставленої мети необхідно реалізувати такі завдання:

- провести аналіз актуальності створення чат-бота для використання різними організаціями;
- виконати реалізацію функціонала чат-бота;
- провести тестування розробленого продукту.

Викладення основного матеріалу. Будь-яка співбесіда – стрес і хвилювання для претендента: окрім відмінного резюме та вміння триматися, завжди є так званий людський фактор. Зазвичай на співбесіди ставлять заздалегідь підготовлені питання для отримання бачення рівня кваліфікації.

Все частіше у великих компаніях усі можливі процеси автоматизують. Проведення тестування – не виняток. Згідно з прогнозами міжнародної аудиторсько-консалтингової компанії, до 2023 року 7 з 10 співробітників працюватимуть поза офісом, а найматимуть їх боти та програми штучного інтелекту. В Україні вже теж є компанії, де тестування кандидата займаються боти (роботи) – спеціально написані програми. Для більшої схожості з людиною їх називають, даючи співзвучні з людськими імена. Бот оцінює рівень знань, які мають відповідати вимогам вакансії. Для автоматизації процесу тестування необхідно створити чат-бота на платформі Telegram, який в режимі реального часу буде оцінювати рівень знань претендента, на результат якого в майбутньому буде спиратися HR-менеджер.

Для початку створення бота потрібен акаунт у месенджері «Telegram». Спілкування з ботом майже нічим не відрізняється від спілкування з реальною особою в чаті. Різниця полягає в тому, що спілкування проходить не з людиною, а з програмним продуктом із застосуванням штучного інтелекту. Взаємодія з ботом може відбуватися в будь-який час доби та день тижня. Але перший крок за людиною, необхідно знайти бот за його іменем – telegram.me/<ім'я бота>.

Мова програмування Python. Для реалізації поставленого завдання було обрано мову програмування Python. Python – є універсальною мовою, яка може бути використана для різних цілей: створення вебдодатків, ігор, чат-ботів, обробки та аналізу даних. Відносний лаконізм мови Python дозволяє створити програму, яка буде набагато коротшого за свій аналог, написаний на іншій мові.

База даних MongoDB. Для зберігання, аналізу та обробки даних було обрано систему керування базами даних MongoDB. MongoDB – це база даних, яка є документоорієнтована з відкритим вихідним кодом і вона не вимагає опису схем таблиць. Під час проектування базами даних створено такі колекції: Bot, Users. Для збереження даних про питання та відповіді тесту була призначена колекція Bot (рис. 1 і табл. 1).

```

    _id: ObjectId("62dfd768cb2b8ed515f2593f")
    id: 0
    text: "Скільки різних груп крові у людини?"
    ~ answers: Array
      0: "1"
      1: "2"
      2: "3"
      3: "4"
      4: "5"
      5: "6"
    correct: 3

```

Рис. 1. Візуалізація даних колекції Bot в MongoDB

Таблиця 1

Колекція Bot

Назва	Тип даних	Опис
id	Int32	Ідентифікатор запитання
text	String	Текст запитання
answers	Array	Масив даних з відповідями
correct	Int32	Правильна відповідь

Дані у колекцію User генеруються автоматично при старті та змінюються впродовж усього тестування (табл. 2 та рис. 2).

Таблиця 2

Колекція Users

Назва	Тип даних	Опис
chat_id	Int32	Ідентифікатор чату (користувача)
is_passing	Boolean	Статус тестування (проходить)
is_passed	Boolean	Статус тестування (пройдено)
question_index	Int32	Номер запитання на даний момент проходження тестування
answers	Array	Масив з обраними відповідями

```

_id: ObjectId("62e3dcac0428971e9a0f341b")
chat_id: 608881332
is_passing: true
is_passed: false
question_index: 1
▼ answers: Array
  0: 0

```

Рис. 2. Візуалізація даних колекції Users в MongoDB

Створення Telegram-бота та реалізація програмного коду. Перед написанням програмного коду за допомогою офіційного бота Telegram @BotFather потрібно створити нового бота та отримати його API Токен. Початком написання коду є підключення бібліотек ruTelegramBotAPI та pymongo. Наступним кроком підключаємо базу даних MongoDB. Створюємо колекції та клас DataBase, в якому будуть функції для керування базою даних.

Ідентифікація користувача відбувається за chat_id. Функція перевіряє користувача за chat_id, створює нового користувача, якщо його немає в базі даних. Функція, яка повертає текст і відповіді на запитання, наведена на рисунку 3.

```

def get_question_message(user):
    if user["question_index"] == db.questions_count: count = 0
    for question_index, question in enumerate(db.questions.find({})):
        if question["correct"] == user["answers"][question_index]:
            count += 1
    percents = round(100 * count / db.questions_count)
    if percents < 40:
        smile = "😞"
    elif percents < 60:
        smile = "😐"
    elif percents < 90:
        smile = "😄"
    else:
        smile = "😁"
    text = f"Ви відповіли правильно на {percents}% питань {smile}"
    db.set_user(user["chat_id"],
                {"is_passed": True, "is_passing": False}) return {
        "text": text, "keyboard": None
    }
    question = db.get_question(user["question_index"]) if question is None:
    return
    keyboard = telebot.types.InlineKeyboardMarkup()
    for answer_index, answer in enumerate(question["answers"]):
        keyboard.row(telebot.types.InlineKeyboardButton(f"{chr(answer_index +
97)} {answer}"),
                    callback_data=f"?ans&{answer_index}")
    text = f"Питання №{user['question_index'] + 1}\n\n{question['text']}" return {
        "text": text, "keyboard": keyboard
    }

```

Рис. 3. Приклад функції для перевірки тексту

Якщо користувач пройшов тест, тобто коли `question_index` буде дорівнювати кількості запитань, буде рахуватися відсоток правильних відповідей, в інших випадках створюється пуста клавіатура. Переглядаємо всі відповіді та додаємо необхідні кнопки в клавіатуру. Для кращого сприйняття додаємо Емої. Оновлюємо параметри користувача. Створюємо змінну з текстом повідомлення. Повертаємо словник з текстом і клавіатурою.

Функція, яка викликається, коли користувач обрав відповідь, представлена на рисунку 4. В ній перевіряються параметри `is_passed` та `is_passing`. Далі додається відповідь користувача.

```
@bot.callback_query_handler(func=lambda query: query.data.startswith("?ans"))
def answered(query):
    user = db.get_user(query.message.chat.id)
    if user["is_passed"] or not user["is_passing"]: return
    user["answers"].append(int(query.data.split("&")[1]))
    db.set_user(query.message.chat.id, {"answers": user["answers"]})
    post = get_answered_message(user)
    if post is not None:
        bot.edit_message_text(post["text"], query.message.chat.id, query.message.id,
            reply_markup=post["keyboard"])
```

Рис. 4. Приклад функції, яка викликається, коли користувач натиснув на відповідь

Функція, яка буде повертати відповідь користувача, представлена на рисунку 5. У ній створено текстову змінну. В змінну додаємо номер і текст запитання. Переглядаємо всі відповіді, галочкою відмічаємо правильну відповідь, хрестиком – неправильну. Створимо кнопку «Далі».

```
def get_answered_message(user):
    question = db.get_question(user["question_index"])
    text = f"Питання №{user['question_index'] + 1}\n\n{question['text']}\n"
    for answer_index, answer in enumerate(question["answers"]):
        text += f"{chr(answer_index + 97)} {answer}" if answer_index == question["correct"]:
        text += " ✓"
    elif answer_index == user["answers"][-1]: text += " ✗"
    text += "\n"
    keyboard = telebot.types.InlineKeyboardMarkup()
    keyboard.row(telebot.types.InlineKeyboardButton("Далі", callback_data="?next"))
    return {
        "text": text, "keyboard": keyboard
    }
```

Рис. 5. Приклад функції відповіді користувача

Тестування Telegram-бота. Тестування бота здійснювалося вручну в месенджері «Telegram». Для початку тестування потрібно відкрити чат з ботом, знайшовши його в пошуку за нікнеймом `@MyPersonalWikiBot` або перейшовши за посиланням, яке було видане ботом `@BotFather`. Для того щоб запустити бот, потрібно натиснути клавішу «Розпочати», або відправити команду `/start`. Після запуску бота одразу з'являється перше запитання (рис. 6).

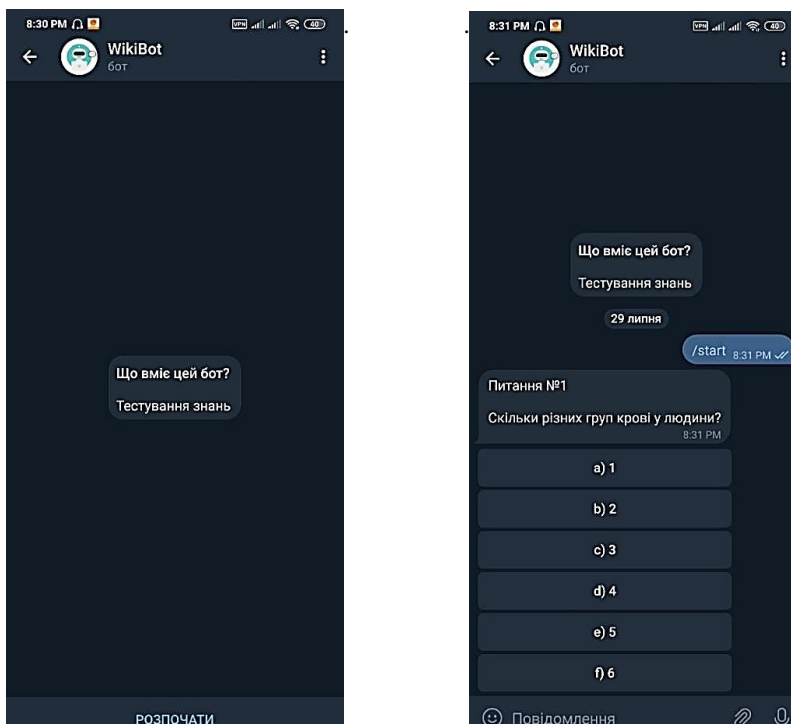


Рис. 6. Вікно чату з ботом та запуск бота

Після вибору відповіді, бот надає підтвердження, в якому відображаються ці відповіді і кнопка «Далі», щоб перейти до наступного питання. Галочкою позначається правильна відповідь, хрестиком – неправильна (рис. 7).

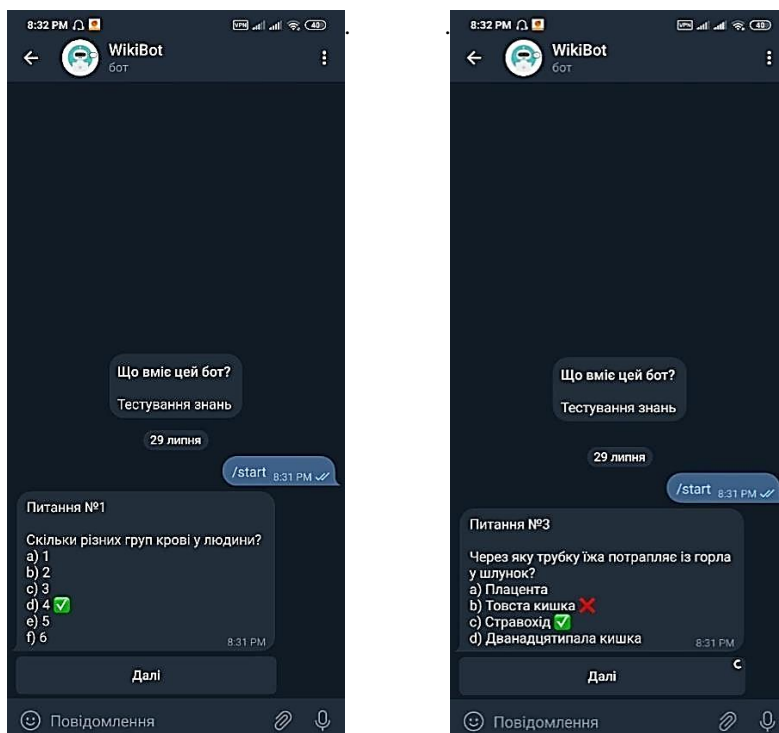


Рис. 7. Повідомлення про правильну і неправильну відповідь

Після проходження усього тесту з'являється повідомлення з кількістю правильних відповідей у відсотках (рис. 8). При повторній спробі пройти тест з'явиться повідомлення «Ви вже пройшли це тестування. Вдруге пройти неможливо» (рис. 8).

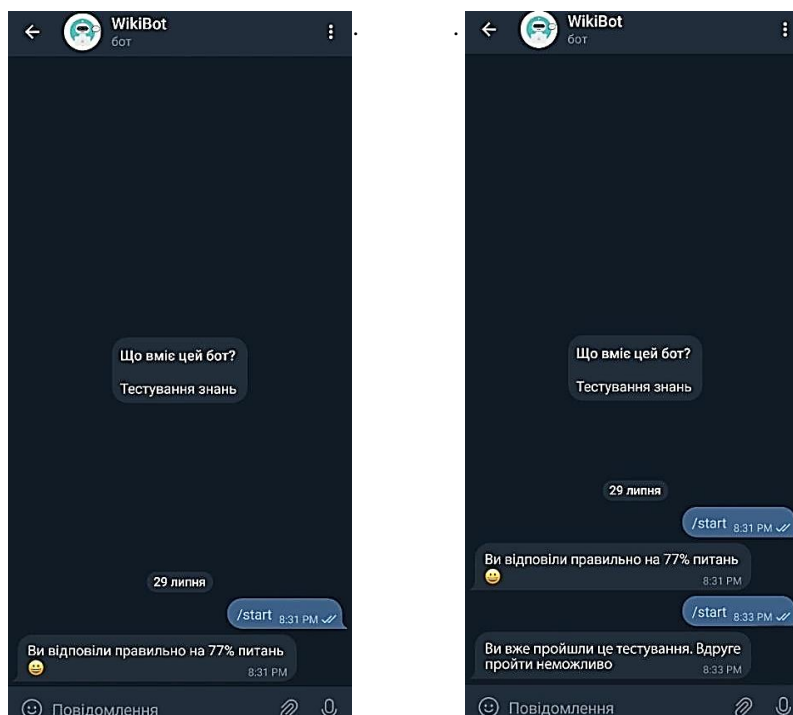


Рис. 8. Результат проходження тесту та відповідь на повторну спробу пройти тест

Висновки та перспективи подальших досліджень. У процесі роботи було проведено аналіз задачі, засобів та методів її розв'язання, в результаті якого визначено необхідність створення чат-бота для перевірки рівня знань на співбесіді. В результаті дослідження було обрано мову програмування Python, джерело даних MongoDB та середовище розробки проекту PyCharm.

Telegram-бота створено за допомогою офіційного додатка Telegram @BotFather, код було написано на багатомовній мові програмування Python, з використанням таких бібліотек, як pyTelegramBotAPI та pymongo.

Результатом проведеного дослідження стало створення Telegram-бота для перевірки рівня знань перед проходженням співбесіди. Було проведено тестування розробленого Telegram-бота. Проведено аналіз різних сценаріїв, наприклад, проведення повторного тестування для перевірки знань.

Список використаної літератури:

1. *Коротун О.В.* Проектування та розробка документоорієнтованої системи керування базами даних / *О.В. Коротун, Г.В. Марчук, В.В. Медведєв* // Технічна інженерія. – 2021. – № 2 (88). – С. 63–71.
2. CHAT-Bot: a cultural heritage aware teller-bot for supporting touristic experiences Pattern Recogn / *M.Casillo, F. Clarizia, G.D'Aniello and other* // Lett. – 2020. – № 131. – P. 234–243. DOI: 10.1016/j.patrec.2020.01.003.
3. *Preez du S.J.* An intelligent web-based voice chat bot / *S.J. du Preez, M.Lall, S.Sinha* // IEEE EUROCON 2009. – 2009. – P. 386–391. DOI: 10.1109/EUROCON.2009.5167660.
4. *Rui Yan* Chitty-chitty-chat bot : deep learning for conversational AI / *Yan Rui* // In Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI'18). – AAAI Press, 2018. – P. 5520–5526.
5. *Haller E.* Designing a Chat-bot that Simulates an Historical Figure / *E.Haller, T.Rebedea* // 19th International Conference on Control Systems and Computer Science, 2013. – P. 582–589. DOI: 10.1109/CSCS.2013.85.
6. *Rodrigo S.M.* Development and Implementation of a Chat Bot in a Social Network / *S.M. Rodrigo, J.G.F. Abraham* // Ninth International Conference on Information Technology – New Generations, 2012. – P. 751–755. DOI: 10.1109/ITNG.2012.147.
7. *Holtgraves T.* A procedure for studying online conversational processing using a chat bot / *T.Holtgraves, T.L. Han* // Behav Res. – 2007. – № 39. – P. 156–163. DOI: 10.3758/BF03192855.
8. Hello Harlie: enabling speech monitoring through chat-bot conversations / *D.Ireland, C.Atay, J.J. Liddle and other* // Studies in health technology and informatics. – 2016. – № 227. – P. 55–60 [Electronic resource]. – Access mode : <https://eprints.qut.edu.au/128687/>.
9. Алгоритмічно-програмне забезпечення обробки та аналізу потоку кадрів відеоданих, що надходять з камер міста : комп'ютерна програма / *В.Л. Левківський, Г.В. Марчук, В.В. Ципоренко, Д.К. Марчук*. – 2021 [Electronic resource]. – Access mode : <https://cutt.ly/jMxy1sq>.
10. Available parking places recognition system / *V.Levkivskyi, D.Marchuk, N.Lobanchykova and other* // CEUR Workshop Proceedings 4th Workshop for Young Scientists in Computer Science & Software Engineering. – 2022. – Vol. 3077. – P. 123–134 [Electronic resource]. – Access mode : <http://ceur-ws.org/Vol-3077/paper07.pdf>.

References:

1. Korotun, O.V., Marchuk, G.V. and Mjedvjedjev, V.V. (2021), «Projektuvannja ta rozrobka dokumentoorijentovanoi systemy keruvannja bazamy danyh», *Tehnichna inzhenerija*, No. 2 (88), pp. 63–71.
2. Casillo, M., Clarizia, F., D'Aniello, G. et al. (2020), «CHAT-Bot: a cultural heritage aware teller-bot for supporting touristic experiences Pattern Recogn», *Lett*, No. 131, pp. 234–243, doi: 10.1016/j.patrec.2020.01.003.
3. Preez, du S.J., Lall, M. and Sinha, S. (2009), «An intelligent web-based voice chat bot», *IEEE EUROCON 2009*, pp. 386–391, doi: 10.1109/EURCON.2009.5167660.
4. Rui, Yan (2018), *Chitty-chitty-chat bot, deep learning for conversational AI*, In Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI'18), AAAI Press, pp. 5520–5526.
5. Haller, E. and Rebedea, T. (2013), *Designing a Chat-bot that Simulates an Historical Figure*, 19th International Conference on Control Systems and Computer Science, pp. 582–589, doi: 10.1109/CSCS.2013.85.
6. Rodrigo, S.M. and Abraham, J.G.F. (2012), *Development and Implementation of a Chat Bot in a Social Network*, Ninth International Conference on Information Technology – New Generations, pp. 751–755, doi: 10.1109/ITNG.2012.147.
7. Holtgraves, T. and Han, T.L. (2007), «A procedure for studying online conversational processing using a chat bot», *Behav Res*, No. 39, pp. 156–163, doi: 10.3758/BF03192855.
8. Ireland, D., Atay, C., Liddle, J.J. et al. (2016), «Hello Harlie: enabling speech monitoring through chat-bot conversations», *Studies in health technology and informatics*, No. 227, pp. 55–60, [Online], available at: <https://eprints.qut.edu.au/128687/>
9. Levkivskyj, V.L., Marchuk, G.V., Cyporenko, V.V. and Marchuk, D.K. (2021), *Algorytmichno-programne zabezpechennja obrobky ta analizu potoku kadriv videodanyh, shho nadhodjat' z kamer mista, komp'juterna programa*, [Online], available at: <https://cutt.ly/jMxy1sq>
10. Levkivskyi, V., Marchuk, D.N., Lobanchykova, N. and other (2022), «Available parking places recognition system», *CEUR Workshop Proceedings 4th Workshop for Young Scientists in Computer Science & Software Engineering*, Vol. 3077, pp. 123–134, [Online], available at: <http://ceur-ws.org/Vol-3077/paper07.pdf>

Городецька Вікторія Валеріївна – магістрант групи ICTM-21-1 Державного університету «Житомирська політехніка».

Наукові інтереси:
– аналіз даних.

Левківський Віталій Леонідович – старший викладач кафедри комп'ютерних наук Державного університету «Житомирська політехніка».

<https://orcid.org/0000-0001-8318-3794>.
Наукові інтереси:
– аналіз даних

Левченко Антон Юрійович – старший викладач кафедри інженерії програмного забезпечення Державного університету «Житомирська політехніка».

<https://orcid.org/0000-0002-4411-6465>.
Наукові інтереси:
– аналіз даних.

Gorodetska V.V., Levkivskyi V.L., Levchenko A.Yu.

Telegram bot development to check the level of knowledge of an applicant for filling a vacant position

Modern information technologies provide wide opportunities for quickly obtaining information. The problem of timeliness of obtaining information remains relevant today. Chatbots are one of the effective means for its solution. The popularity of chatbots is growing every year. They are used in various spheres of human activity for testing, advertising, surveys, interviewing of job applicants. The availability and speed of modern chatbots are decisive for the rational use of time. The purpose of our work is the software implementation of a Telegram bot for conducting preliminary testing and checking the level of knowledge and skills of an applicant for filling a vacant position in organizations of various fields of activity. The software product is written in Python and uses MongoDB, the document-oriented open source database management system as a data store. The database stores information for testing, as well as the results of passed tests. Methods of object-oriented design, theory of document-oriented databases were used in the work. The result of the development is a software product in the form of a Telegram bot for conducting an interview or testing an applicant for a position. Based on the results of the testing, the recruitment manager has the opportunity to draw preliminary conclusions. This test makes it possible to assess the applicant's qualification level and suitability for the position.

Keywords: chatbot; messenger; data; Python; MongoDB; artificial intelligence; interview.

Стаття надійшла до редакції 27.09.2022.