

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ КИЇВСЬКИЙ
НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАДИМА
ГЕТЬМАНА**

**Навчально-науковий інститут
«Інститут інформаційних технологій в економіці»
Кафедра інформаційних систем в економіці**

Форма навчання: заочна

КВАЛІФІКАЦІЙНА БАКАЛАВРСЬКА РОБОТА

на тему: **«Розроблення чат-бота для автоматизації рекрутингу з використанням АРІ OPENAI та технологій штучного інтелекту»**
здобувача Тихонова Івана Олександровича _____

Науковий керівник:

к.т.н., доцент

_____ Маркіна О. М.

**Робота допущена до захисту
перед екзаменаційною комісією з
атестації здобувачів вищої освіти
завідувач кафедри:**

к.е.н., доцент

_____ Тішков Б.О.

Київ 2025

Міністерство освіти і науки України
Київський національний економічний університет імені Вадима Гетьмана
Навчально-науковий інститут «Інститут інформаційних технологій в економіці»
Кафедра інформаційних систем в економіці

ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНІ НАУКИ»

галузь знань 12 «Інформаційні технології»

спеціальність 122 «Комп'ютерні науки»

ПОГОДЖЕНО:

Керівник проектної групи (гарант)
освітньо-професійної програми

_____ Помазун О.М.

“ _____ ” _____ 2025 р.

ЗАТВЕРДЖЕНО:

Завідувач кафедри інформаційних
систем в економіці

_____ Тішков Б.О.

“ _____ ” _____ 2025 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

здобувача вищої освіти *Тихонова Івана Олександровича* заочної форми навчання

на підготовку кваліфікаційної бакалаврської роботи

**на тему: «Розроблення чат-бота для автоматизації рекрутингу з використанням API
OPENAI та технологій штучного інтелекту»**

Тему затверджено наказом ректора Університету від «7» березня 2025р. № 466- ст

Кваліфікаційний бакалаврська робота виконується на матеріалах отриманих під час навчання, здійснення студентських наукових досліджень, а також зібраного фактичного матеріалу з обраної теми дослідження під час практики.

План кваліфікаційного бакалаврського проєкту

Розділ 1. Аналіз предметної галузі та дослідження існуючих рішень

Розділ 2. Проєктування чат-бота для рекрутингу

Розділ 3. Реалізація та тестування чат-бота

Об'єкт дослідження: процес автоматизації рекрутингу за допомогою чат-ботів та штучного інтелекту.

Предмет дослідження: методи та технології розробки чат-бота для автоматизації рекрутингових процесів із використанням API OpenAI.

Мета кваліфікаційного бакалаврського проєкту: є розробка чат-бота для автоматизації рекрутингу з використанням API OpenAI та технологій штучного інтелекту, що дозволить оптимізувати процес підбору персоналу, зменшити навантаження на HR-фахівців та підвищити ефективність комунікації з кандидатами.

Конкретні завдання, які студент повинен виконати для досягнення поставленої мети:

У РОЗДІЛІ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ДОСЛІДЖЕННЯ ІСНУЮЧИХ РІШЕНЬ виконати наступні завдання:

1.1. Аналіз предметної галузі

1.1.1 Огляд процесів рекрутингу та їх автоматизації.

1.1.2 Визначення основних етапів рекрутингу та типових проблем.

1.1.3 Опис сучасних підходів до автоматизації найму.

1.1.4 Визначення ключових учасників процесу (HR-фахівці, кандидати, менеджери).

1.2. Огляд існуючих рішень та технологій

1.2.1. Аналіз популярних HR-чат-ботів.

1.2.2. Порівняння технологій штучного інтелекту для рекрутингу.

1.2.3. Використання API OpenAI для обробки тексту та генерації відповідей.

1.2.4. Аналіз переваг та недоліків існуючих рішень.

У РОЗДІЛІ 2. ПРОЄКТУВАННЯ ЧАТ-БОТА ДЛЯ РЕКРУТИНГУ виконати наступні завдання:

2.1. Вимоги до системи, що проєктуєте

2.1.1. Функціональні вимоги:

- Автоматизоване спілкування з кандидатами
- Оцінка відповідності кандидатів вакансіям
- Формування звітів для HR-фахівців
- Інтеграція з зовнішніми платформами (LinkedIn, Work.ua)

2.1.2. Нефункціональні вимоги:

- Безпека обробки персональних даних
- Масштабованість та продуктивність
- Інтуїтивний інтерфейс взаємодії

2.2. Основні задачі та алгоритми

2.2.1 Описати основні задачі, які має вирішувати система (автоматизоване управління IT-замовленнями, інтеграція з CRM-системами тощо).

2.2.2 Розробити алгоритми для реалізації цих задач (наприклад, алгоритм рейтингового відбору виконавців, автоматичне ціноутворення).

2.3. Архітектура системи

2.3.1. Побудова діаграм (UML-діаграми, діаграми послідовностей).

2.3.2. Вибір архітектурного підходу (Serverless, Microservices)..

2.3.3. Інтеграція OpenAI API для обробки тексту.

2.3.4. Моделювання структури бази даних.

2.4. Розробка алгоритмів чат-бота

2.4.1. NLP-підхід для аналізу відповідей кандидатів.

2.4.2. Використання машинного навчання для оцінки відповідності кандидатів вакансії.

2.4.3. Обробка діалогових сценаріїв.

У РОЗДІЛІ 3. РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ЧАТ-БОТА

3.1. Опис технологічного стеку

3.1.1. Вибір середовища для розгортання (AWS Lambda, Google Cloud Functions).

3.1.2. Використані інструменти (Python, Django/FastAPI, PostgreSQL).

3.1.3. Інтеграція OpenAI API.

3.2. Реалізація функціоналу

3.2.1. Впровадження обробки запитів та відповідей.

3.2.2. Розробка панелі керування для HR-фахівців.

3.2.3. Збереження результатів у базі даних.

3.2.4. Налаштування інтеграції з кар'єрними платформами.

3.3. Тестування та оптимізація

3.3.1. Перевірка коректності відповідей чат-бота.

3.3.2. Навантажувальне тестування та оцінка продуктивності.

3.3.3. Аналіз безпеки та захисту персональних даних.

Завдання підготував

науковий керівник

Завдання одержав

здобувач

Маркіна Ольга Миколаївна

“10” березня 2025 р.

Тихонов Іван Олександрович

“10” березня 2025 р.

ВІДГУК

про кваліфікаційну бакалаврську роботу
здобувача навчально-наукового інституту
«Інститут інформаційних технологій в економіці»
освітньо-професійної програми
«Комп'ютерні науки»

Тихонова Івана Олександровича

на тему: «Розроблення чат-бота для автоматизації рекрутингу з використанням API OpenAI та технологій штучного інтелекту»

Актуальність теми. У сучасних умовах динамічного розвитку цифрових технологій, зокрема у сфері рекрутингу, зростає потреба в інтелектуальних системах, що дозволяють автоматизувати обробку великої кількості резюме, комунікацію з кандидатами та попередній відбір. Застосування чат-ботів із вбудованими механізмами ШІ, такими як API OpenAI, є інноваційним підходом, що дозволяє значно покращити ефективність процесу найму персоналу. Тема кваліфікаційної роботи є сучасною, практично орієнтованою та затребуваною на ринку України.

Позитивні риси кваліфікаційної бакалаврської роботи. Робота характеризується чіткою структурою, логічною послідовністю викладення матеріалу та грамотним обґрунтуванням вибору інструментів реалізації. Розроблений чат-бот демонструє працездатність системи, інтегрується з базою даних, аналізує вхідну інформацію та забезпечує функціональну взаємодію з користувачем.

Наявність самостійних розробок автора. Автор самостійно реалізував чат-бот із застосуванням Python, Flask, SQLite та API OpenAI. Створено структуру обробки діалогів, реалізовано алгоритми аналізу анкет кандидатів, обробки запитів і генерації відповідей. Усі етапи розробки виконано на високому рівні з урахуванням актуальних підходів до побудови AI-рішень.

Цінність теоретичних висновків та практичних рекомендацій. У роботі наведено обґрунтовані теоретичні висновки щодо ефективності використання чат-ботів у рекрутингу та перспектив розвитку таких рішень. Практичні рекомендації є цінними для подальшого вдосконалення систем

автоматизації рекрутингу та можуть бути використані в ІТ-компаніях або стартапах.

Наявність недоліків. У роботі частково не охоплено питання безпеки персональних даних кандидатів. Відсутній модуль багатокористувацької взаємодії або системи ролей. Також було б доцільно розширити функціонал візуалізації результатів для рекрутера у вигляді аналітичних звітів.

Загальна оцінка кваліфікаційної бакалаврської роботи та її допущення до захисту перед ЕК. Кваліфікаційна бакалаврська робота Тихонова Івана Олександровича виконана на високому рівні, відповідає вимогам освітньої програми та демонструє здатність здобувача до комплексного вирішення прикладних завдань у сфері комп'ютерних наук.

РЕЦЕНЗІЯ

РЕЦЕНЗІЯ

на кваліфікаційну бакалаврську роботу
здобувача вищої освіти

Тихонова Івана Олександровича

Тема «Розроблення чат-бота для автоматизації рекрутингу з використанням API OPENAI та технологій штучного інтелекту»

Актуальність теми кваліфікаційної бакалаврської роботи. У сучасному бізнес-середовищі автоматизація процесів підбору персоналу стає критично важливою для ефективного функціонування компаній. Застосування чат-ботів і штучного інтелекту в рекрутингу дає змогу знизити навантаження на HR-відділи, пришвидшити обробку заявок і підвищити якість взаємодії з кандидатами.

Якість проведеного дослідження. У роботі представлено якісне дослідження предметної області, розглянуто сучасні інструменти створення чат-ботів, проаналізовано можливості API OpenAI та обґрунтовано їх застосування. Описано архітектуру створеної системи, механізми обробки запитів, інтеграцію з базою кандидатів та логіку взаємодії з користувачами. Здобувач продемонстрував вміння працювати з сучасними AI-рішеннями та технологіями, продемонструвавши прикладне використання теоретичних знань.

Позитивні риси кваліфікаційної бакалаврської роботи. Повнота і логічна структура викладеного матеріалу.

- Актуальність тематики та практична орієнтованість розробки.
- Комплексний підхід до побудови чат-бота із залученням API OpenAI.
- Якісне опрацювання структури діалогу та сценаріїв спілкування.
- Успішна реалізація інтеграції з базами даних кандидатів.
- Високий рівень самостійності виконання програмної частини.

Зауваження:

- У роботі не реалізовано повноцінного тестування системи з участю реальних користувачів (рекрутерів/кандидатів), що могло б покращити оцінку UX.
- Опис безпекових аспектів та захисту даних кандидатів подано стисло.
- Було б доцільно додати порівняльний аналіз з альтернативними існуючими рішеннями.

Практична значимість висновків і рекомендацій. Розроблений чат-бот є прикладом ефективного застосування AI в бізнес-практиці та має значний потенціал для впровадження в реальних рекрутингових процесах. Сформульовані рекомендації можуть бути використані під час проектування подібних систем або розширення функціоналу існуючих рішень у сфері HRTech.

Кваліфікаційна робота Тихонова Івана Олександровича відповідає вимогам до робіт бакалаврського рівня, демонструє необхідний рівень самостійності, практичної реалізації та аналітичного опрацювання, заслуговує на оцінку «добре».

Місце роботи та посада рецензента

Доцент кафедри інформаційних систем та технологій
КПІ ім Ігоря Сікорського к.т.н., доцент



/Батрак С.О./

Підпис засвідчую

/ /

АНОТАЦІЯ

кваліфікаційної бакалаврської роботи
здобувача першого (бакалаврського) рівня вищої освіти, 4 курсу,
виконаної на тему: «**Розроблення чат-бота для автоматизації
рекрутингу з використанням API OPENAI та технологій штучного
інтелекту**»

У кваліфікаційній бакалаврській роботі розроблено веб-застосунок для автоматизації первинної комунікації з кандидатами на вакансії із застосуванням мовної моделі GPT. Основною ідеєю є створення чат-бота, який імітує діалог HR-фахівця, надає відповіді на запити кандидатів та оцінює їх відповідність актуальним вакансіям.

У реалізації використано стек технологій: FastAPI, PostgreSQL, SQLAlchemy, Jinja2, Docker, а також OpenAI API для генерації відповідей. Система підтримує ролі HR і кандидата, зберігає історію чатів, дозволяє завантажувати резюме та генерує узагальнений технічний висновок для рекрутера.

Новизна бакалаврської роботи полягає у поєднанні генеративного ШІ з кастомною HR-панеллю та псевдоінтеграцією з кар'єрними платформами. Проведено функціональне, навантажувальне й безпекове тестування. Розробка відповідає вимогам сучасних інструментів автоматизації рекрутингу.

У теоретичній частині роботи досліджено особливості застосування чат-ботів у сфері управління персоналом, здійснено аналіз інструментів створення розумних діалогових систем, обґрунтовано вибір мовної моделі та технологій реалізації.

Практична частина присвячена розробленню та тестуванню програмного продукту, що включає модулі обробки введених даних, класифікації резюме, збереження результатів у базі даних і генерації відповідей на основі OpenAI API.

Розроблена система має прикладний характер, може бути використана у малих та середніх компаніях для автоматизації рекрутингових процесів та є основою для подальшого розвитку функціоналу.

Ключові слова: чат-бот, рекрутинг, OpenAI, штучний інтелект, автоматизація, інформаційна система.

РЕФЕРАТ

Кваліфікаційна бакалаврська робота містить 61 сторінку, 5 таблиць, 24 рисунки, список літератури з 21 найменування, 12 додатків.

«Розроблення чат-бота для автоматизації рекрутингу з використанням API OPENAI та технологій штучного інтелекту»

Предметом дослідження є методи та технології розробки чат-бота для автоматизації рекрутингових процесів із використанням API OpenAI.

Об'єктом дослідження виступає процес автоматизації рекрутингу за допомогою чат-ботів та штучного інтелекту.

Мета бакалаврської роботи полягає в розробці чат-бота для автоматизації рекрутингу з використанням API OpenAI та технологій штучного інтелекту, що дозволять оптимізувати процес підбору персоналу, зменшити навантаження на HR-фахівців та підвищити ефективність комунікації з кандидатами.

Завданнями бакалаврської роботи є:

- аналіз існуючих підходів до автоматизації рекрутингу;
- створення архітектури вебзастосунку на основі FastAPI;
- інтеграція мовної моделі OpenAI для генерації відповідей;
- розробка панелі управління для HR-фахівців;
- збереження діалогів та результатів у базі даних;
- проведення тестування та оцінка безпеки.

Апаратні та програмні засоби, що використовувались при проєктуванні: персональний комп'ютер, середовище розробки VS Code, PyCharm, PostgreSQL, FastAPI, Docker, OpenAI API.

Результати досягнуті в процесі роботи (також їх новизна та ступінь упровадження): розроблено повнофункціональний вебзастосунок, що дозволяє кандидатам спілкуватися з GPT-базованим чат-ботом, а HR — переглядати результати оцінювання та керувати вакансіями. Новизна полягає у поєднанні генеративного ШІ з кастомною інтерфейсною панеллю, що адаптована під реальні кейси рекрутингу. Проєкт протестований та готовий до демонстрації як MVP.

Одержані результати можуть бути використані у компаніях для автоматизації первинного відбору кандидатів, а також як основа для інтеграції в реальні HRM-системи. Сфера використання — рекрутинг, HRTech, освітні платформи.

Перелік ключових слів: чат-бот, штучний інтелект, FastAPI, рекрутинг, OpenAI, база даних, генеративна модель, автоматизація, вебзастосунок, HR-панель.

Рік виконання кваліфікаційної бакалаврської роботи: 2025

Рік захисту кваліфікаційної бакалаврської роботи: 2025

ЗМІСТ

ЗМІСТ	1
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ ВЕЛИЧИН І ТЕРМІНІВ.....	3
ВСТУП.....	5
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ДОСЛІДЖЕННЯ ІСНУЮЧИХ РІШЕНЬ.....	7
1.1 Аналіз предметної галузі	7
1.1.1 Огляд процесів рекрутингу та їх автоматизації.....	7
1.1.2 Визначення основних етапів рекрутингу та типових проблем	8
1.1.3 Опис сучасних підходів до автоматизації найму.....	9
1.1.4 Визначення ключових учасників процесу (HR-фахівці, кандидати, менеджери).....	10
1.2 Огляд існуючих рішень та технологій	11
1.2.1 Аналіз популярних HR-чат-ботів.....	11
1.2.2 Порівняння технологій штучного інтелекту для рекрутингу.....	13
1.2.3 Використання API OpenAI для обробки тексту та генерації відповідей	14
1.2.4 Аналіз переваг та недоліків існуючих рішень	16
РОЗДІЛ 2. РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ ЧАТ-БОТА ДЛЯ РЕКРУТИНГУ.....	18
2.1 Вимоги до системи, що проєктується.....	18
2.1.1 Функціональні вимоги	18
2.1.2 Нефункціональні вимоги	20
2.2 Основні задачі та алгоритми	22
2.2.1 Опис основних задач, які вирішує система	22
2.2.2 Розробка алгоритмів для реалізації основних задач	24
2.3 Архітектура системи.....	26
2.3.1 Побудова діаграм (UML-діаграми, діаграми послідовностей)	26
2.3.2 Вибір архітектурного підходу (Serverless, Microservices)	29
2.3.3 Інтеграція OpenAI API для обробки тексту.....	31
2.3.4 Моделювання структури бази даних	33
2.4 Розробка алгоритмів чат-бота	37
2.4.1 NLP-підхід для аналізу відповідей кандидатів	37
2.4.2 Використання машинного навчання для оцінки відповідності кандидатів вакансії	39
2.4.3 Обробка діалогових сценаріїв	40
РОЗДІЛ 3. ПРОЄКТУВАННЯ, РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ЧАТ-БОТА	42

3.1 Інформаційне забезпечення	42
3.1.1 Загальна характеристика інформаційного забезпечення	42
3.1.2 Організація збору і передання первинної інформації	42
3.1.3 Побудова системи класифікації та кодування.....	42
3.1.4 Проектування форм первинних документів та відеокадрів	42
3.1.5 Структура інформаційних масивів	43
3.1.6 Вибір СКБД	43
3.1.7 Інфологічна модель бази (сховища) даних	43
3.1.8 Даталогічна модель бази (сховища) даних	43
3.2 Технічне забезпечення	43
3.2.1 Загальні вимоги до технічного забезпечення	43
3.2.2 Серверне технічне забезпечення.....	44
3.2.3 Клієнтське технічне забезпечення	44
3.2.4 Технічне середовище розробника	44
3.3 Програмне забезпечення (опис технологічного стеку)	45
3.3.1 Вибір середовища для розгортання (AWS Lambda, Google Cloud Functions).....	45
3.3.2 Використані інструменти (Python, Django/FastAPI, PostgreSQL).....	46
3.3.3 Інтеграція OpenAI API	47
3.4 Реалізація функціоналу	47
3.4.1 Впровадження обробки запитів та відповідей	47
3.4.2 Розробка панелі керування для HR-фахівців.....	48
3.4.3 Збереження результатів у базі даних.....	49
3.4.4 Налаштування інтеграції з кар'єрними платформами	50
3.4.5 Перевірка коректності відповідей чат-бота	51
3.4.6 Навантажувальне тестування та оцінка продуктивності	53
3.4.7 Аналіз безпеки та захисту персональних даних.....	55
ВИСНОВКИ.....	58
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	60

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ ВЕЛИЧИН І ТЕРМІНІВ

API – інтерфейс прикладного програмування (Application Programming Interface) — набір правил, які дозволяють взаємодіяти між різними програмами.

AI – штучний інтелект (Artificial Intelligence) — галузь інформатики, що займається створенням систем, здатних імітувати інтелектуальну поведінку людини.

HR – Human Resources — відділ управління персоналом, фахівці з підбору кадрів.

JWT – JSON Web Token — компактний формат для безпечної передачі інформації між сторонами як JSON-об'єкт.

UI – користувацький інтерфейс (User Interface) — графічна або текстова оболонка для взаємодії користувача із застосунком.

UX – користувацький досвід (User Experience) — загальне враження користувача від взаємодії з продуктом або сервісом.

FastAPI – високопродуктивний вебфреймворк для створення API на Python з асинхронною підтримкою.

PostgreSQL – реляційна система управління базами даних з відкритим кодом.

Docker – платформа для контейнеризації застосунків, що дозволяє запускати їх у ізольованих середовищах.

MVP – мінімально життєздатний продукт (Minimum Viable Product) — базова версія продукту, придатна для використання.

ChatGPT / OpenAI API – мовна модель штучного інтелекту, що забезпечує генерацію тексту на основі запитів користувача.

HTTP – протокол передачі гіпертексту (Hypertext Transfer Protocol).

JSON – формат обміну даними на основі тексту (JavaScript Object Notation).

REST – стиль архітектури вебсервісів, що передбачає використання стандартних методів HTTP.

БД – база даних — організована структура для збереження інформації.

Кандидат – користувач системи, який проходить автоматизоване інтерв'ю з чат-ботом.

Вакансія – опис посади з вимогами до кандидата, що створюється HR-фахівцем.

Сесія – тимчасова інформація про авторизованого користувача, яка зберігається для взаємодії з системою.

ВСТУП

Сучасний ринок праці висуває нові вимоги до швидкості, якості та ефективності процесу підбору персоналу. Збільшення кількості заявок на вакансії, зміни у форматах комунікації з кандидатами, поява віддаленої та гібридної роботи спричинили необхідність впровадження новітніх технологій в управлінські HR-процеси. Особливої актуальності набуває застосування систем штучного інтелекту для автоматизації рутинних етапів рекрутингу — від первинної комунікації до попереднього аналізу резюме. Для України ця тема є надзвичайно важливою в умовах адаптації внутрішнього ринку праці до викликів війни, еміграції фахівців і глобальної конкуренції за кадри [1, 5-13].

Актуальність теми зумовлена потребою у швидких, гнучких та економічно ефективних рішеннях у сфері управління персоналом, які можуть забезпечити базову комунікацію з претендентами, структурований збір інформації та попередній аналіз кандидатів, особливо для вітчизняних рекрутингових центрів під час війни. Існуючі інструменти, зокрема CRM-рішення для рекрутерів, часто є або надто дорогими, або не враховують особливості локального ринку. Застосування чат-ботів на базі мовних моделей нового покоління, таких як OpenAI GPT, дозволяє створювати персоналізовані й адаптивні рішення [1-5, 7, 8].

Зі зростанням кількості кандидатів і швидкістю змін на ринку праці, традиційні методи найму виявляються малоефективними. Одним із рішень цієї проблеми є автоматизація процесу рекрутингу з використанням чат-ботів та технологій штучного інтелекту (ШІ), що дозволяють значно зменшити навантаження на HR-фахівців, скоротити час найму та покращити якість взаємодії з кандидатами.

Об'єктом дослідження є процес автоматизації рекрутингу за допомогою чат-ботів та штучного інтелекту.

Предметом дослідження є методи та технології розробки чат-бота для автоматизації рекрутингових процесів із використанням API OpenAI.

Метою кваліфікаційної бакалаврської роботи є розробка чат-бота для автоматизації рекрутингу з використанням API OpenAI та технологій

штучного інтелекту, що дозволить оптимізувати процес підбору персоналу, зменшити ручну роботу HR-фахівців та підвищити ефективність комунікації з кандидатами.

Завдання, які необхідно вирішити для досягнення мети:

- здійснити аналіз предметної галузі та існуючих технологічних рішень у сфері автоматизації рекрутингу;
- сформулювати функціональні та нефункціональні вимоги до інформаційної системи;
- спроектувати архітектуру чат-бота з урахуванням особливостей використання API OpenAI;
- реалізувати MVP-прототип чат-бота з базовим функціоналом взаємодії з користувачами;
- протестувати систему, оцінити її ефективність та сформулювати рекомендації щодо подальшого розвитку.

Методи дослідження, що використовуються в роботі, включають аналіз літературних джерел, системне моделювання, методи обробки природної мови (NLP), використання хмарних сервісів та прикладне програмування з використанням API.

Апробація результатів роботи здійснювалася в процесі навчальної та дослідницької діяльності автора.

Структура роботи. Кваліфікаційна бакалаврська робота складається з трьох основних розділів. У першому розділі здійснено аналіз предметної галузі та досліджено існуючі рішення в області автоматизації рекрутингу. У другому розділі сформульовано вимоги до системи, описано її архітектуру та моделі даних. У третьому розділі подано реалізацію чат-бота, описано обраний технологічний стек, процес тестування та результати функціонування системи.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ДОСЛІДЖЕННЯ ІСНУЮЧИХ РІШЕНЬ

1.1 Аналіз предметної галузі

1.1.1 Огляд процесів рекрутингу та їх автоматизації

Рекрутинг — це процес пошуку, відбору та найму кваліфікованих кандидатів для заповнення вакансій у компаніях. Його мета полягає в забезпеченні підприємства людськими ресурсами, які відповідають вимогам конкретної посади та корпоративній культурі організації. Рекрутинговий процес, як правило, включає аналіз вакансій, публікацію оголошень про роботу, збір резюме, проведення співбесід, оцінювання кандидатів та прийняття рішень щодо найму.

Традиційно цей процес здійснюється вручну HR-фахівцями, що потребує значних часових та ресурсних витрат. Зі збільшенням кількості кандидатів, а також у зв'язку зі зростаючою конкуренцією на ринку праці, підприємства все частіше звертаються до інструментів автоматизації рекрутингу.

Автоматизація рекрутингу передбачає використання сучасних інформаційних технологій для оптимізації та часткового або повного заміщення ручних дій у процесі найму персоналу. Найбільш поширеними напрямками автоматизації є:

- автоматичне розміщення вакансій на сайтах пошуку роботи та в соціальних мережах;
- використання систем відстеження кандидатів (ATS) для збору, зберігання та фільтрації резюме;
- впровадження чат-ботів для попередньої комунікації з кандидатами та проведення базового скринінгу;
- застосування алгоритмів машинного навчання для оцінювання відповідності кандидатів вимогам вакансії;
- автоматизація планування співбесід, відправлення нагадувань та зворотного зв'язку.

Використання автоматизованих систем у сфері рекрутингу дозволяє зменшити навантаження на HR-відділи, підвищити швидкість обробки заявок та забезпечити об'єктивність оцінювання кандидатів. Особливої актуальності набуває застосування технологій штучного інтелекту, зокрема моделей обробки природної мови (NLP), які дозволяють покращити якість аналізу текстів резюме та мотиваційних листів, а також реалізувати ефективну діалогову взаємодію з кандидатами.

1.1.2 Визначення основних етапів рекрутингу та типових проблем

Процес рекрутингу зазвичай складається з послідовності етапів, які забезпечують системний підхід до найму працівників. Кожен з цих етапів має свої особливості та може бути оптимізований за допомогою цифрових рішень.

До основних етапів рекрутингу належать:

- визначення потреб у персоналі, аналіз вакансій та складання опису посади;
- публікація вакансій на спеціалізованих платформах і в соціальних мережах;
- збір і первинна обробка резюме кандидатів;
- попередній відбір (скринінг) за визначеними критеріями;
- проведення співбесід (онлайн або офлайн);
- оцінювання кандидатів за результатами співбесід і тестових завдань;
- прийняття рішення про найм та оформлення відповідних документів.

Попри стандартизовану структуру, на практиці рекрутинговий процес супроводжується низкою типових проблем:

1. **Великий обсяг ручної роботи.** HR-фахівцям доводиться обробляти сотні резюме, проводити десятки однотипних співбесід, що призводить до перевантаження персоналу.

2. **Низька якість кандидатів.** Часто на одну вакансію надходить велика кількість невідповідних заявок, що ускладнює пошук релевантних спеціалістів.
3. **Тривалі строки закриття вакансій.** Несвоєчасний найм може впливати на продуктивність відділів, у яких бракує працівників.
4. **Людський фактор.** Помилки при оцінюванні, упередженість, втома або неуважність можуть вплинути на об'єктивність відбору.
5. **Низька ефективність комунікації з кандидатами.** Затримки з відповідями, відсутність фідбеку або неточна інформація можуть негативно вплинути на репутацію компанії.

Вирішення цих проблем потребує впровадження сучасних технологічних інструментів, зокрема систем автоматизації та чат-ботів, які можуть взяти на себе частину рутинних завдань та підвищити ефективність кожного з етапів рекрутингу.

1.1.3 Опис сучасних підходів до автоматизації найму

Сучасні підходи до автоматизації найму базуються на використанні цифрових технологій, які дозволяють оптимізувати рекрутингові процеси, зменшити витрати часу та підвищити якість прийняття рішень. Ці підходи охоплюють як окремі етапи найму, так і повний життєвий цикл залучення кандидатів.

До найбільш розповсюджених практик автоматизації належать:

- використання систем управління наймом (ATS — Applicant Tracking Systems), що дозволяють вести базу кандидатів, відслідковувати статуси заявок, фільтрувати та сортувати резюме за заданими параметрами;
- застосування чат-ботів, які здійснюють попередню комунікацію з кандидатами, відповідають на типові запитання, проводять опитування та попередній відбір;
- використання аналітики та алгоритмів машинного навчання для прогнозування успішності кандидатів на основі аналізу резюме,

соціальних профілів, тестових завдань або навіть невербальних сигналів під час відеоспівбесіди;

- автоматичне планування співбесід із урахуванням календарів учасників процесу;
- генерація індивідуального зворотного зв'язку для кандидатів за результатами відбору.

Особливу роль у сучасних підходах до автоматизації відіграє використання технологій штучного інтелекту. За допомогою моделей обробки природної мови (Natural Language Processing) можливо аналізувати текстові документи (резюме, мотиваційні листи), проводити семантичний пошук відповідностей між вакансіями та кандидатами, а також вести діалогову взаємодію у зручному для користувача форматі.

Дедалі частіше компанії також інтегрують автоматизовані рекрутингові системи з корпоративними CRM, ERP та HRM-системами, що дозволяє створювати єдину екосистему для управління персоналом.

Таким чином, сучасні підходи до автоматизації рекрутингу зосереджені не лише на скороченні часу пошуку кандидатів, але й на підвищенні якості прийняття рішень, покращенні досвіду кандидата (candidate experience) та посиленні аналітичних можливостей HR-відділів.

1.1.4 Визначення ключових учасників процесу (HR-фахівці, кандидати, менеджери)

У процесі рекрутингу беруть участь кілька ключових сторін, кожна з яких виконує свої функції та має певні очікування щодо результату. Ефективна взаємодія між цими учасниками є критично важливою для успішного закриття вакансій і формування якісної команди.

HR-фахівці (рекрутери) — основні координатори процесу найму. Їхня роль включає:

- аналіз потреб компанії та формулювання вимог до кандидатів;
- підготовку вакансій та розміщення оголошень;

- проведення попереднього відбору резюме та скринінгу кандидатів;
- організацію співбесід і тестування;
- ведення комунікації з кандидатами та внутрішніми стекхолдерами;
- супровід нових працівників під час адаптації.

Кандидати — особи, які подають заявки на участь у конкурсі. Їхній досвід взаємодії з компанією під час процесу найму часто впливає на загальне враження від бренду роботодавця. Очікування кандидатів включають:

- швидкий і зрозумілий процес подання заявки;
- своєчасний зворотний зв'язок;
- прозорість вимог та умов працевлаштування;
- індивідуальний підхід і коректну комунікацію.

Менеджери (керівники підрозділів) — кінцеві замовники процесу рекрутингу. Вони визначають потребу у персоналі, затверджують опис вакансії, беруть участь у фінальному відборі кандидатів і несуть відповідальність за їхню подальшу ефективність у команді. Обов'язки менеджерів включають:

- уточнення профілю і компетенцій необхідного спеціаліста;
- участь у співбесідах і прийнятті рішень щодо найму;
- взаємодію з HR щодо адаптації та оцінювання нових працівників.

Кожен з учасників має власні цілі, які не завжди співпадають. Наприклад, HR-фахівці зацікавлені у швидкому заповненні вакансії, кандидати — у якісному досвіді спілкування, а менеджери — у довготривалому підборі оптимального спеціаліста. Саме тому важливою складовою автоматизованих рішень є створення інтерфейсів і функціональностей, які задовольняють потреби кожного учасника процесу.

1.2 Огляд існуючих рішень та технологій

1.2.1 Аналіз популярних HR-чат-ботів

HR-чат-боти — це програмні агенти, які імітують людську комунікацію з кандидатами з метою автоматизації рутинних етапів процесу найму. Вони можуть працювати на базі скриптів (rule-based) або використовувати технології штучного інтелекту (AI-based), зокрема моделі обробки природної мови.

Серед найвідоміших комерційних рішень можна виокремити такі системи:

1. **Olivia (Paradox).** Один із найпопулярніших AI-базованих чат-ботів для рекрутингу. Olivia може відповідати на запитання кандидатів, проводити первинний скринінг, планувати співбесіди та навіть формувати звіти для HR-відділу. Вона інтегрується з великою кількістю ATS і HRM-систем.
2. **XOR.** Цей бот підтримує багатомовну комунікацію, автоматизує перевірку резюме, проводить опитування та надає статистику щодо кандидатів. XOR активно використовується в масовому підборі персоналу, зокрема в ритейлі та логістиці.
3. **HireVue.** Спеціалізується на відеоінтерв'ю з елементами AI-аналізу. Чат-бот HireVue використовується для початкової взаємодії з кандидатами перед переходом до відеоінтерв'ю. Також система оцінює невербальну поведінку, емоції, швидкість і зміст відповідей.
4. **JobPal (SAP).** Чат-бот інтегрується з SAP SuccessFactors. Його завдання — комунікація з кандидатами у месенджерах (Facebook Messenger, WhatsApp тощо), допомога з поданням заявки, відповіді на часті питання.
5. **Recruitee Bot.** Пропонує базову автоматизацію процесу комунікації: нагадування про співбесіди, підтвердження участі, збирання фідбеку після інтерв'ю. Підходить для малих і середніх компаній.

Популярність HR-ботів пояснюється їх здатністю працювати 24/7, зменшувати навантаження на рекрутерів та покращувати досвід кандидатів. При цьому ефективність рішень значною мірою залежить від якості інтеграції, можливостей кастомізації та рівня “інтелекту” системи.

У роботі пропонується створення власного чат-бота з використанням OpenAI API, який поєднуватиме гнучкість генеративної моделі з можливістю обробки персональних даних та інтеграції у типову рекрутингову інфраструктуру.

1.2.2 Порівняння технологій штучного інтелекту для рекрутингу

Технології штучного інтелекту (ШІ) дедалі активніше впроваджуються у сферу рекрутингу. Вони дозволяють автоматизувати складні когнітивні задачі, підвищити точність відбору кандидатів та зменшити вплив людського фактора. У контексті автоматизації найму найбільш розповсюдженими є такі напрями застосування ШІ:

1. ***Rule-based*** системи (на основі фіксованих правил). Це найпростіші варіанти реалізації “інтелекту” у чат-ботах. Вони працюють за задалегідь прописаними сценаріями: “якщо → тоді”. Такі системи легко реалізувати, але вони погано масштабуються, обмежені у розумінні природної мови та не здатні до адаптації. Приклад: бот, який ставить стандартні питання та приймає рішення лише за ключовими словами у відповіді.
2. ***Machine Learning (ML)***. Машинне навчання дозволяє ботам та системам найму вчитися на прикладах минулих рекрутингових кейсів. Наприклад, модель може навчитися визначати релевантність кандидата вакансії на основі великої кількості резюме та рішень про найм. Підходи на основі ML дають змогу автоматизувати сортування кандидатів, прогнозувати успішність співпраці та виявляти шаблони в поведінці.
3. ***Natural Language Processing (NLP)***. NLP-технології використовуються для аналізу текстової інформації, наприклад:
 - обробка резюме (витяг релевантної інформації);
 - порівняння вимог вакансії з описами навичок;
 - генерація персоналізованих відповідей у чаті;
 - розуміння намірів кандидата (intent detection).

4. **Generative AI** (наприклад, GPT-3.5/4 від OpenAI). Новітній напрямок — використання генеративних моделей, які здатні створювати текст, відповідати на запитання, симулювати діалог і навіть оцінювати зміст повідомлень. Такі моделі забезпечують високу гнучкість, легко адаптуються до нових сценаріїв та працюють майже без сценарної прив'язки. Приклад: GPT-модель, яка веде діалог із кандидатом і ставить додаткові запитання в залежності від попередньої відповіді.

Порівняльна характеристика напрямів застосування ШІ наведена у таблиці 1.1.

Таблиця 1.1

Порівняльна характеристика напрямів застосування ШІ

Підхід	Гнучкість	Якість розуміння	Адаптація	Швидкість реалізації
Rule-based	Низька	Низька	Відсутня	Висока
ML-моделі	Середня	Середня	Висока	Середня
NLP (класичне)	Висока	Середня/висока	Середня	Середня
Generative AI (GPT)	Висока	Висока	Висока	Висока

Джерело: сформовано автором на основі виконаного дослідження

Таким чином, у сучасних умовах генеративні моделі, зокрема ті, що базуються на OpenAI, мають найбільший потенціал для гнучкої та масштабованої автоматизації рекрутингових процесів. Їхнє впровадження дає змогу вийти за межі шаблонних відповідей і забезпечити природну, адаптивну комунікацію з кандидатами.

1.2.3 Використання API OpenAI для обробки тексту та генерації відповідей

API OpenAI надає доступ до потужних моделей штучного інтелекту, зокрема GPT-3.5 та GPT-4, які здатні розуміти природну мову, генерувати тексти, відповідати на запитання, аналізувати вхідну інформацію та імітувати діалог. Ці можливості відкривають широкі перспективи для їх застосування у сфері рекрутингу, особливо в автоматизованих чат-ботах.

Основні переваги використання API OpenAI у рекрутингу:

- Обробка природної мови (NLP): моделі здатні розпізнавати наміри користувача, виділяти ключову інформацію з резюме або текстових відповідей, формулювати уточнювальні запитання.
- Генерація адаптивних відповідей: GPT може будувати відповіді не за фіксованим сценарієм, а з урахуванням контексту діалогу, що значно підвищує гнучкість та “людяність” комунікації.
- Оцінка відповідності: за допомогою спеціально сформульованих prompt-ів (запитів) модель може проаналізувати, наскільки відповіді кандидата відповідають вимогам вакансії, і дати обґрунтовану оцінку або рекомендацію.
- Створення контенту: чат-бот на базі GPT може автоматично формувати шаблони вакансій, листи-відповіді кандидатам, запрошення на інтерв'ю, зворотний зв'язок тощо.
- Інтеграція з іншими системами: API легко інтегрується у веб-застосунки, месенджери або внутрішні HRM/ATS-системи, що дозволяє будувати цілісну цифрову екосистему.

Технічно, робота з OpenAI API передбачає надсилання текстового запиту (prompt) на сервер і отримання у відповідь згенерованого тексту. Наприклад, на вхід можна передати резюме кандидата та опис вакансії, а у відповідь отримати оцінку релевантності або сформульовані питання для уточнення.

Важливим аспектом є також можливість контролю вихідного результату за допомогою параметрів, таких як `temperature` (варіативність відповіді), `max_tokens` (довжина відповіді), `system role` (стиль комунікації тощо).

Таким чином, використання OpenAI API у чат-боті для рекрутингу дозволяє створити більш природний, адаптивний та функціональний інтерфейс взаємодії, що значно перевершує можливості традиційних rule-based систем.

1.2.4 Аналіз переваг та недоліків існуючих рішень

На сучасному ринку представлено велику кількість інструментів для автоматизації рекрутингу — від простих rule-based чат-ботів до потужних платформ, що використовують штучний інтелект. Хоча ці рішення значно покращують ефективність найму, кожне з них має як переваги, так і обмеження.

Переваги існуючих рішень:

1. **Оптимізація часу HR-фахівців.** Завдяки автоматичному відбору, комунікації та плануванню зустрічей рекрутери можуть зосередитися на стратегічних завданнях.
2. **Доступність 24/7.** Більшість чат-ботів можуть обробляти заявки та відповідати на запити кандидатів у будь-який час доби без залучення персоналу.
3. **Масштабованість.** Інструменти ШІ дозволяють обслуговувати велику кількість кандидатів одночасно, що особливо корисно при масовому наймі.
4. **Аналітичні можливості.** Системи з функціями машинного навчання можуть надавати HR-відділу детальну аналітику щодо воронки найму, ефективності джерел трафіку, релевантності кандидатів тощо.
5. **Покращення candidate experience.** Швидкі відповіді, зрозумілий процес і мінімізація ручної взаємодії підвищують лояльність кандидатів.

Недоліки існуючих рішень:

1. **Висока вартість впровадження.** Платформи на базі ШІ, зокрема GPT, потребують ліцензій, серверних потужностей, інтеграцій і

налаштування, що може бути фінансово обтяжливим для малих компаній.

2. **Обмежена кастомізація в SaaS-рішеннях.** Багато готових продуктів мають фіксований функціонал і не дозволяють гнучко адаптуватися до специфіки конкретного бізнесу.
3. **Проблеми з обробкою персональних даних.** При використанні зовнішніх API, особливо закордонних (як-от OpenAI), можуть виникати питання щодо конфіденційності та відповідності законодавству (GDPR, ЗУ “Про захист персональних даних”).
4. **Галюцинації генеративних моделей.** GPT-моделі іноді створюють неправдиву або недоречну інформацію, що вимагає додаткової перевірки з боку людини.
5. **Складність інтеграції.** Для досягнення повноцінного функціоналу потрібна якісна технічна інтеграція з існуючими системами компанії (CRM, ERP, поштові сервіси, месенджери тощо).

У результаті, хоча існуючі рішення здатні значно покращити процес найму, жодне з них не є універсальним. Це створює попит на індивідуальні, гнучкі рішення — зокрема на базі OpenAI API — які можна адаптувати до конкретних потреб організації та комбінувати з іншими цифровими інструментами.

РОЗДІЛ 2. РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ ЧАТ-БОТА ДЛЯ РЕКРУТИНГУ

2.1 Вимоги до системи, що проєктується

На основі проведеного аналізу предметної галузі, а також із урахуванням сучасних технологічних трендів та можливостей генеративних моделей, було сформульовано перелік функціональних та нефункціональних вимог до інформаційної системи, що проєктується.

2.1.1 Функціональні вимоги

Функціональні вимоги визначають, які конкретні функції повинна реалізувати система для досягнення поставленої мети — автоматизації рекрутингового процесу за допомогою чат-бота на основі штучного інтелекту та API OpenAI. Нижче подано опис основних функцій, які система повинна підтримувати в рамках розробки.

Автоматизоване спілкування з кандидатами.

Система повинна забезпечити можливість двосторонньої комунікації між ботом та кандидатом у зручному чат-інтерфейсі. Комунікація має відбуватись у реальному часі або асинхронно, з урахуванням попередніх відповідей користувача (контексту діалогу).

Чат-бот повинен:

- ініціювати контакт із кандидатом після отримання заявки або запити;
- ставити структуровані запитання згідно з шаблоном вакансії (наприклад, “Скільки років досвіду у вас у Python?”);
- виявляти та обробляти ключові слова й фрази у відповідях;
- змінювати хід діалогу залежно від контексту (використовуючи GPT);
- вести лог діалогів для подальшого аналізу HR-фахівцями.

Оцінка відповідності кандидатів вакансіям.

Чат-бот повинен мати вбудовану логіку попереднього оцінювання кандидатів. На основі відповідей, наданих у чаті, бот має:

- зіставляти досвід, навички та освіту з вимогами конкретної вакансії;
- присвоювати умовний рейтинг або категорію (наприклад, “висока відповідність”, “середня”, “низька”);
- ідентифікувати відсутність критичних навичок (наприклад, відсутність досвіду в певному стеку);
- формувати короткий висновок щодо релевантності, який потім передається до панелі керування HR-менеджера.

Цей функціонал має бути реалізований на основі алгоритмів порівняння ключових слів, а також із використанням можливостей NLP для семантичного аналізу відповідей.

Формування звітів для HR-фахівців.

За результатами кожної взаємодії з кандидатом система повинна генерувати структурований звіт, який міститиме:

- загальні дані (ім’я, контактна інформація, бажана позиція);
- перелік отриманих відповідей;
- короткий аналіз відповідності вимогам;
- автоматичні коментарі та рекомендації від бота (за результатами GPT-оцінювання);
- дата, час, тривалість діалогу.

Звіти повинні бути доступні через внутрішню панель керування та мати можливість експорту у форматах PDF/CSV.

Інтеграція з зовнішніми платформами (LinkedIn, Work.ua).

Система повинна підтримувати можливість інтеграції з популярними рекрутинговими платформами та API зовнішніх сервісів. Це дозволить:

- автоматично отримувати профілі кандидатів із LinkedIn (через відкритий API або скрейпінг);

- синхронізувати вакансії, які публікуються в системі, з біржами праці (Work.ua, Djinni, Rabota.ua);
- підключити передачу отриманих результатів у сторонні CRM або ATS-системи.

Інтеграція має бути реалізована у вигляді окремих модулів або API-шлюзів, що дозволить масштабувати систему за потреби.

2.1.2 Нефункціональні вимоги

Нефункціональні вимоги визначають характеристики системи, які не пов'язані безпосередньо з її бізнес-функціоналом, але є критично важливими для її ефективного використання, масштабування, безпеки та зручності взаємодії з користувачем. У контексті даного проєкту враховуються такі основні нефункціональні вимоги:

Безпека обробки персональних даних.

Одним із ключових аспектів системи є забезпечення конфіденційної обробки персональних даних кандидатів, оскільки під час рекрутингового процесу зберігаються чутливі відомості, зокрема: ПІБ, контактна інформація, освіта, досвід роботи, мотиваційні листи та інше.

Для цього необхідно реалізувати:

- шифрування даних при зберіганні (наприклад, з використанням AES-256) та передачі (TLS 1.2+);
- автентифікацію та авторизацію доступу до HR-панелі (JWT або OAuth2);
- обмеження прав доступу до даних згідно з ролями користувачів (role-based access control);
- логування дій користувачів у системі для забезпечення аудиту;
- відповідність вимогам законодавства України щодо захисту персональних даних, а також вимогам GDPR у разі потенційної взаємодії з європейськими кандидатами.

Масштабованість та продуктивність.

Система має бути спроектована з урахуванням можливості обробки великої кількості одночасних запитів без втрати продуктивності. Це особливо важливо в умовах масового рекрутингу або проведення кампаній із залучення персоналу в кількох регіонах одночасно.

Передбачається:

- використання асинхронних технологій обробки запитів на рівні backend (FastAPI, asyncio);
- підтримка горизонтального масштабування — наприклад, можливість розгортання кількох інстансів застосунку через Docker/Kubernetes;
- обмеження навантаження на OpenAI API через черги (Redis Queue, Celery) або ліміти використання токенів;
- кешування частих відповідей (наприклад, FAQ) для зниження затримок.

Тестування продуктивності буде реалізовано на основі сценаріїв зростаючого навантаження із метою визначення максимальної кількості сесій, яку система здатна обслуговувати без деградації.

Інтуїтивний інтерфейс взаємодії.

Інтерфейс взаємодії з системою повинен бути простим, доступним і зрозумілим як для кандидатів, так і для HR-фахівців. Важливо забезпечити позитивний досвід користувача, що впливає як на імідж компанії, так і на ефективність комунікації.

Для кандидатів:

- зручний чат-інтерфейс, адаптований під мобільні пристрої;
- логічна послідовність запитань та відповідей;
- наявність підказок, валідація введених даних, збереження контексту.

Для HR-фахівців:

- веб-панель з аналітикою, звітами, фільтрами кандидатів та історією спілкування;

- експорт результатів (PDF, Excel);
- сортування та пошук по ключових полях (позиція, рівень відповідності, прізвище тощо);
- адаптація під корпоративні кольори/брендинг.

Таким чином, система повинна бути зручною для кінцевого користувача, швидкою у використанні та стабільною за будь-яких умов навантаження, із високим рівнем захисту даних.

2.2 Основні задачі та алгоритми

2.2.1 Опис основних задач, які вирішує система

Інформаційна система, що проектується, спрямована на автоматизацію ключових етапів процесу рекрутингу з використанням чат-бота, заснованого на штучному інтелекті. Її функціонування охоплює як взаємодію з кандидатами, так і підтримку роботи HR-фахівців. Нижче подано перелік основних задач, які повинна вирішувати система:

- 1. Автоматизоване управління рекрутинговими заявками.**
Система повинна приймати, реєструвати та обробляти запити від потенційних кандидатів, які надходять через форму на сайті, месенджер або інші канали (наприклад, LinkedIn). Для кожної заявки автоматично створюється унікальний запис, який містить персональні дані, вакансію, дату подання та статус проходження.
- 2. Автоматизоване управління IT-замовленнями.** У випадках, коли компанія здійснює підбір IT-фахівців під конкретні проекти або зовнішні замовлення, система повинна забезпечувати облік таких запитів від замовників. Для кожного IT-замовлення зберігається опис проекту, вимоги до спеціаліста, терміни виконання та відповідальний менеджер. Це дозволяє системі автоматично співставляти кандидатів не лише з вакансіями, а й із конкретними проектними потребами.

3. **Первинне інтерв'ювання кандидатів у чаті.** Бот проводить структуроване спілкування з кандидатом у вигляді діалогу. Основна задача — зібрати інформацію про досвід, навички, мови, готовність до релокації, зарплатні очікування тощо. Всі відповіді зберігаються для подальшої оцінки.
4. **Попереднє оцінювання релевантності кандидатів.** На основі зібраних даних система автоматично порівнює профіль кандидата з вимогами вакансії, здійснює семантичний аналіз відповідей через NLP, перевіряє ключові hard/soft skills, формує рейтинг відповідності.
5. **Інтеграція з CRM/ATS-системами компанії.** Передбачено можливість підключення до платформ для управління персоналом, наприклад, BambooHR, Zoho Recruit, Bullhorn. Реалізуються REST API та Webhook-механізми для передачі анкет, оновлення статусів і синхронізації інформації про кандидатів у корпоративній CRM.
6. **Автоматизоване ведення історії взаємодій.** Система повинна зберігати повну історію контактів з кожним кандидатом: чат-діалоги, дати, статуси, зміни профілю, результати оцінювання. Це дозволяє HR-фахівцям бачити повну картину комунікації, уникати повторних питань і приймати більш обґрунтовані рішення.
7. **Генерація звітів для HR-фахівців.** За результатами взаємодії система створює звіти, які містять витяги з відповідей, оцінку відповідності, коментар GPT, технічну інформацію (дата, час, тривалість).
8. **Моніторинг активності та журналювання подій.** Фіксуються IP-адреси, часові мітки, помилки, затримки відповіді API, дії користувачів. Це необхідно для аналітики, навчання моделей та кібербезпеки.
9. **Управління вакансіями та шаблонами діалогів.** HR може створювати нові вакансії та шаблони спілкування. Бот динамічно

підлаштовується під кожен тип вакансії для підвищення ефективності інтерв'ю.

Таким чином, система виконує функції повноцінного автоматизованого інструмента для рекрутингу, включаючи не лише пошук і оцінку кандидатів, а й підтримку IT-замовлень, взаємодію з CRM та повну історичну аналітику.

2.2.2 Розробка алгоритмів для реалізації основних задач

Для ефективного функціонування системи чат-бота з автоматизації рекрутингу необхідно реалізувати низку алгоритмів, які забезпечать обробку даних, взаємодію з користувачами, прийняття рішень та інтеграцію з іншими системами. У цьому підрозділі подано опис ключових алгоритмів, які реалізуються в межах проєкту.

Алгоритм попереднього рейтингового відбору кандидатів.

Цей алгоритм базується на порівнянні наданої кандидатом інформації з параметрами вакансії. Для кожного критерію (наприклад, “досвід у Python”, “знання англійської”, “вища освіта”) встановлюється ваговий коефіцієнт значущості. Алгоритм працює за наступною схемою:

- обробка відповідей кандидата (структурованих або вільної форми) через NLP-модуль GPT;
- виділення ключових параметрів у відповіді;
- зіставлення із вимогами вакансії;
- обрахунок балів за кожним параметром (наприклад, від 0 до 10);
- обчислення загального рейтингу як зваженої суми (2.1):

$$R = \sum(W \times P), \quad (2.1)$$

де: W — вага критерію, P — бал відповідності.

Результатом є рейтинг кандидата (наприклад, 6 зі 10), який відображається у звіті HR-фахівцю.

Алгоритм автоматичного ціноутворення для IT-замовлення.

У випадках, коли кандидат підбирається під конкретне ІТ-замовлення, необхідно оцінити прогнозовану вартість його участі в проєкті (2.2). Алгоритм враховує:

- тривалість проєкту (у місяцях),
- рівень кандидата (Junior / Middle / Senior),
- очікувану ставку за годину або місяць,
- рівень відповідності кандидата замовленню.

Формула розрахунку:

$$C = T \times B \times F, \quad (2.2)$$

де: T — тривалість проєкту, B — базова ставка (залежить від рівня), F — коефіцієнт відповідності (наприклад, 1.0 — повна відповідність, 0.7 — часткова).

Ціна відображається HR-фахівцю як орієнтовна комерційна пропозиція.

Алгоритм генерації персоналізованого зворотного зв'язку (feedback).

На основі діалогу з кандидатом GPT-модель формує короткий текстовий висновок, що включає:

- перелік сильних сторін;
- можливі ризики або недоліки;
- коментар у вільній формі (наприклад, “Кандидат має релевантний досвід, але бракує знань у DevOps”).

Алгоритм реалізується у вигляді prompt-запиту до моделі GPT-4, в якому передається зведена інформація про кандидата у форматі JSON/текст.

Алгоритм вибору сценарію діалогу на основі вакансії.

При ініціації спілкування бот визначає, до якої вакансії звертається кандидат, і завантажує відповідний шаблон діалогу. Кожен шаблон включає:

- перелік запитань;
- ключові очікувані відповіді;
- вказівки для GPT щодо оцінювання.

Алгоритм передбачає динамічне формування prompt-а залежно від вакансії та попередніх відповідей.

Алгоритм інтеграції з зовнішніми платформами (LinkedIn, Work.ua).

Алгоритм забезпечує періодичне сканування вакансій та резюме через API або парсинг (де API відсутнє), з наступною обробкою даних:

- отримання профілю кандидата;
- попереднє форматування (нормалізація даних);
- передача у систему для обробки або активація діалогу.

Також реалізується зворотна передача даних — наприклад, оновлення статусу кандидата на сторонньому сервісі.

Алгоритм валідації і перевірки даних.

Під час заповнення форм (наприклад, email, номер телефону, досвід у роках) реалізується базова перевірка:

- відповідність шаблону (регулярні вирази);
- перевірка на дублікати (у БД);
- логічна валідація (наприклад, досвід не може перевищувати 50 років).

Ці перевірки потрібні для збереження цілісності та якості даних у системі.

Алгоритм інтеграції з CRM.

Для передачі даних у CRM- або ATS-систему використовується Webhook або REST API-запит. Алгоритм включає:

- формування JSON-структури з даними кандидата,
- авторизоване надсилання запиту на зовнішній endpoint,
- логування відповіді сервера (успішно/помилка).

Таким чином, розроблені алгоритми охоплюють як логіку взаємодії з користувачем, так і механізми аналізу, ранжування та передачі інформації до HR. Вони є основою функціонування системи та дозволяють досягнути автоматизації ключових етапів рекрутингу.

2.3 Архітектура системи

2.3.1 Побудова діаграм (UML-діаграми, діаграми послідовностей)

Для наочного представлення логіки функціонування чат-бота та взаємодії користувачів із системою було побудовано кілька UML-діаграм. Ці діаграми допомагають візуалізувати основні сценарії використання, обмін повідомленнями та порядок взаємодії між компонентами системи. Вони також є основою для подальшої реалізації архітектури проекту.

Use Case Diagram — Діаграма варіантів використання (рис. 2.1).

На діаграмі варіантів використання зображено два основних актори — Кандидат та HR-фахівець. Вони взаємодіють із системою за допомогою інтерфейсів (чат-бота та веб-панелі відповідно) у рамках різних сценаріїв.

Кандидат може:

- ініціювати діалог із ботом;
- відповідати на запитання;
- отримувати зворотний зв'язок.

HR-фахівець може:

- переглядати звіти за результатами спілкування бот-кандидат;
- фільтрувати та сортувати кандидатів за рівнем релевантності;
- оновлювати шаблони вакансій та діалогів.

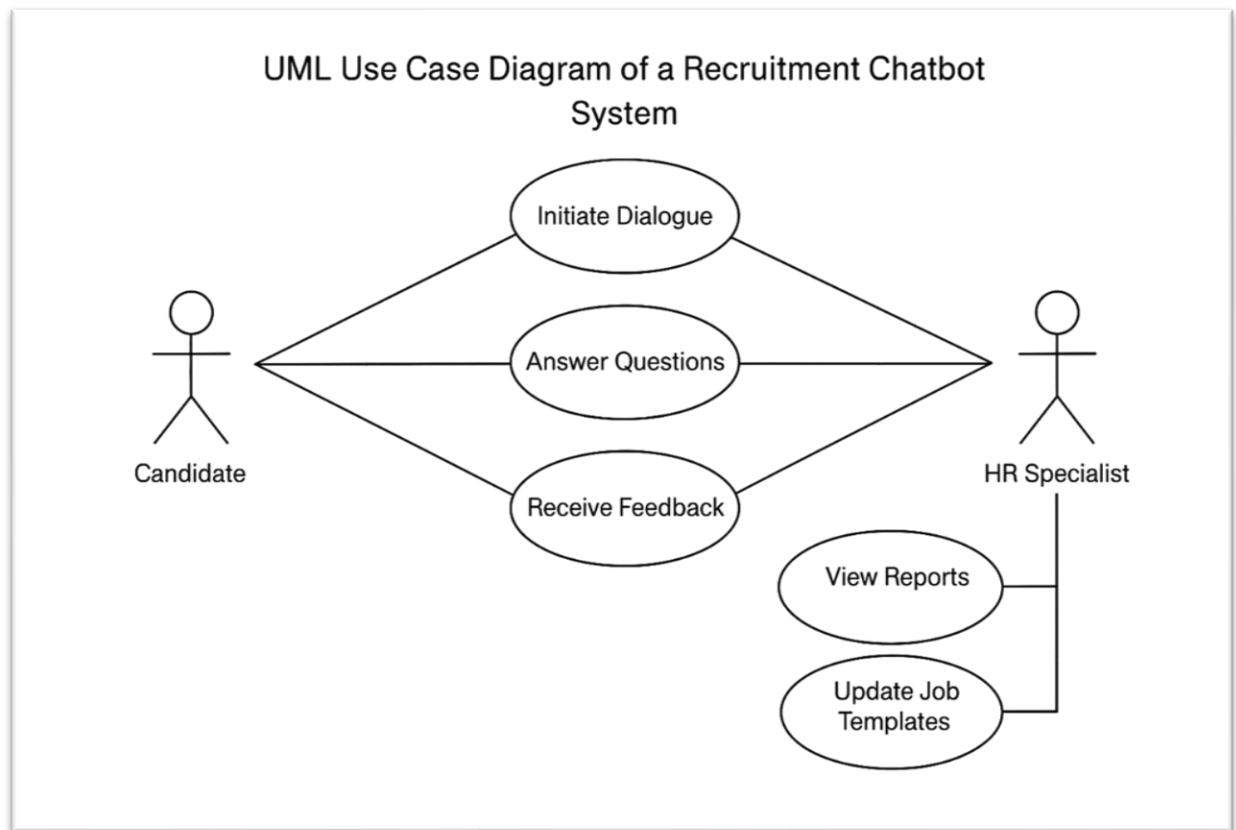


Рис. 2.1 — Діаграма варіантів використання (UseCase)

Джерело: сформовано автором

Ця діаграма дозволяє побачити основну функціональну взаємодію без заглиблення в технічні деталі реалізації.

Sequence Diagram — Діаграма послідовності «Сценарій скринінгу кандидата» (рис. 2.2).

Діаграма послідовності демонструє етапи взаємодії між компонентами системи під час типового процесу скринінгу кандидата через чат-бота.

- Кандидат надсилає перше повідомлення — ініціація діалогу.
- Чат-бот відповідає та ставить серію запитань (згідно зі сценарієм вакансії).
- Кандидат по черзі надає відповіді.
- Система (внутрішній модуль обробки) надсилає відповіді до OpenAI API для оцінювання.
- Отриманий результат зберігається у базу даних.

- Генерується звіт і передається HR-фахівцю через веб-інтерфейс.

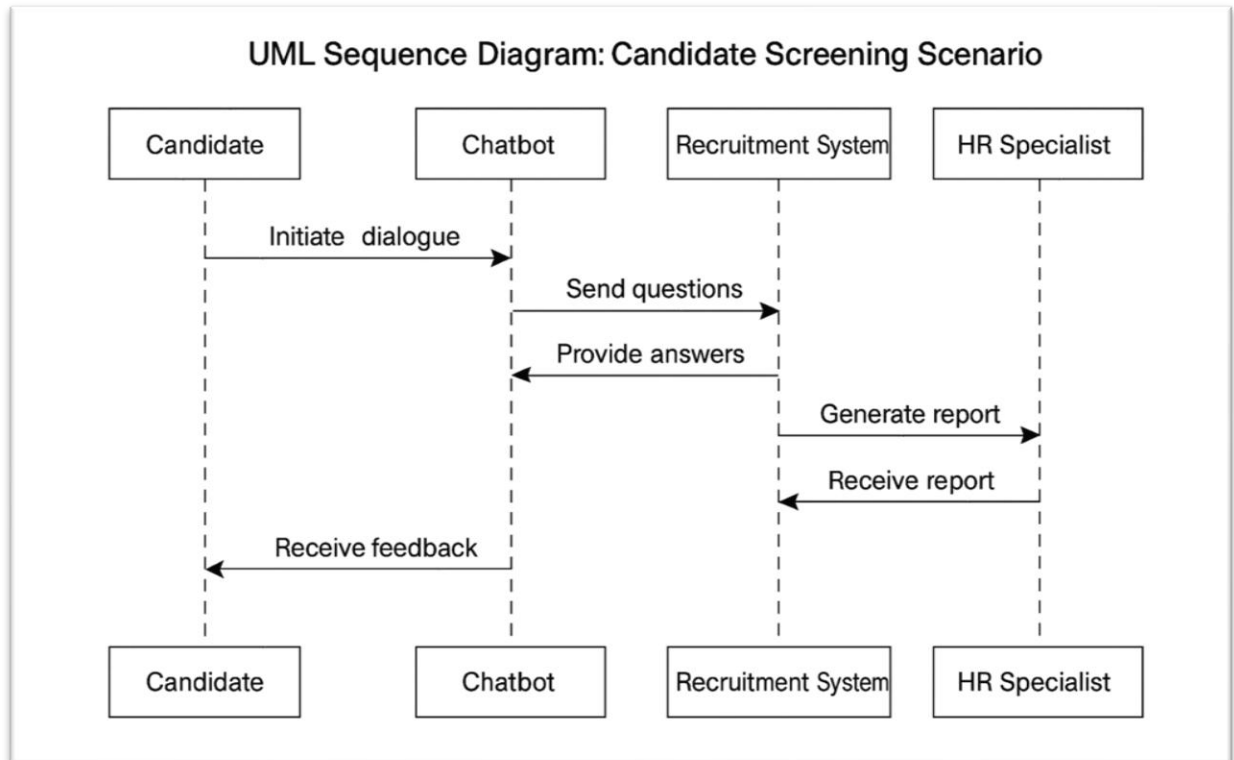


Рис. 2.2 — Діаграма послідовності «Сценарій скринінгу кандидата»

Джерело: сформовано автором

Кожен об'єкт, що бере участь у процесі, представлений окремою вертикальною лінією (акторами або модулями), між якими показано послідовність викликів і відповідей. Така модель дозволяє проаналізувати затримки, критичні точки та визначити необхідні точки обробки даних.

Побудовані UML-діаграми дозволяють сформулювати уявлення про логіку роботи чат-бота як на рівні бізнес-процесів, так і на рівні внутрішніх технічних взаємодій. Діаграми використовуються для формалізації функціональних сценаріїв і служать основою для подальшого моделювання архітектури системи, описаного в наступному підрозділі.

2.3.2 Вибір архітектурного підходу (Serverless, Microservices)

Для створення вебзастосунку з GPT-чат-ботом, панеллю управління для HR та інтеграцією з базами даних було розглянуто три основні архітектурні підходи: Serverless, Microservices та Docker-контейнеризація. Кожен з них має

свої особливості, які були проаналізовані з точки зору придатності до завдань автоматизованого рекрутингу (табл. 2.1).

Таблиця 2.1

Порівняння архітектурних підходів

Критерій	Serverless	Microservices	Docker (моноліт з ізоляцією)
Масштабованість	Висока	Висока	Середня (залежить від інфраструктури)
Швидкість розгортання	Висока	Низька	Висока
Вартість підтримки	Залежить від навантаження	Висока (через складність)	Низька
Складність реалізації	Середня/висока (через state-менеджмент)	Висока	Низька
Підтримка довготривалих сесій	Обмежена	Повна	Повна
Робота з файлами / БД	Ускладнена	Гнучка	Гнучка
Підходить для MVP	Обмежено	Ні	Так

Джерело: сформовано автором на основі виконаного дослідження

Вибір архітектури для реалізації.

Хоча Serverless-підхід приваблює автоматичним масштабуванням і мінімальними витратами на інфраструктуру, його основні обмеження — складність підтримки сесій, обмеження на тривалість запитів і труднощі в роботі з файлами та контекстом GPT — роблять його малоприматним для чат-бота, що зберігає діалоги та резюме.

Microservices-архітектура забезпечує високу гнучкість, але має складну реалізацію, потребує оркестрації (наприклад, Kubernetes) і великого обсягу DevOps-ресурсів. Для MVP-етапу це є недоцільним і технічно перевантаженим варіантом.

У зв'язку з цим було обрано архітектуру на основі Docker-контейнеризації з чіткою структурною ізоляцією компонентів:

- FastAPI-сервер для обробки запитів;
- PostgreSQL як централізована БД;
- GPT-модуль для генерації відповідей (через зовнішній API);
- додаткові worker-и (Celery/Redis) — за потреби асинхронної обробки.

Docker-підхід дозволяє легко розгорнути систему як локально, так і у хмарі (Heroku, Render, Railway), ізолювати середовища, забезпечити стабільність та залишає можливість масштабування у майбутньому через перехід до мікросервісної моделі.

2.3.3 Інтеграція OpenAI API для обробки тексту

Однією з ключових технологічних складових у розробці чат-бота для рекрутингу є інтеграція із зовнішнім API штучного інтелекту, зокрема OpenAI API, що надає доступ до потужних моделей обробки природної мови (NLP), таких як GPT-3.5 і GPT-4. Ця інтеграція дозволяє реалізувати адаптивну генерацію відповідей, інтелектуальний аналіз текстових даних та побудову контекстної взаємодії з користувачем.

Мета інтеграції.

Метою підключення OpenAI API є:

- генерація динамічних відповідей на основі введень кандидата;
- формулювання уточнюючих запитань з урахуванням контексту попереднього діалогу;
- оцінювання відповідей за допомогою семантичного аналізу (напр., чи відповідає досвід кандидата вимогам до вакансії);
- створення підсумкових звітів із висновками бота для HR-фахівця.

Архітектура інтеграції.

Інтеграція OpenAI API реалізується в межах функціонального модуля системи, що відповідає за обробку текстових даних. Типовий сценарій виклику API включає такі етапи:

1. *Кандидат вводить відповідь у чаті.*
2. *Вхідне повідомлення надсилається до обробника діалогу.*
3. *Формується prompt для моделі GPT із включенням контексту (історія попередніх відповідей, мета вакансії, ключові критерії тощо).*
4. *OpenAI API повертає відповідь* — текст, рекомендацію або оцінку.
5. *Отриманий результат* використовується для подальшого діалогу або формування звіту.

Формат запитів.

Комунікація з OpenAI API здійснюється через HTTP POST-запити до ендпоінту <https://api.openai.com/v1/chat/completions>, із передачею наступних параметрів:

- *model*: вказується модель (gpt-3.5-turbo, gpt-4);
- *messages*: список повідомлень у форматі {role, content};
- *temperature*: задає рівень креативності відповіді;
- *max_tokens*: максимальна довжина відповіді;
- *stop*: (опціонально) символи зупинки генерації.

Приклад prompt-а для оцінки відповідності кандидата:

```
[  
  {«role»: «system», «content»: «Ти є асистентом з рекрутингу.  
  Оціни релевантність відповіді кандидата до вказаної вакансії.»},  
  {«role»: «user», «content»: «Вакансія: Python Developer. Вимоги:  
  досвід 3 роки, знання Django, PostgreSQL.»},  
  {«role»: «user», «content»: «Відповідь кандидата: Я працював із  
  Python близько 2 років, маю досвід з Flask та SQLite.»}  
]
```

Безпека доступу.

Для захисту API-ключа від несанкціонованого доступу використовується:

- зберігання ключа у захищених змінних середовища;
- обмеження доступу до функцій, які викликають OpenAI API;
- контроль лімітів та логування помилок.

Інтеграція з OpenAI API значно розширює функціональність системи, дозволяючи реалізувати гнучкий, контекстно-чутливий та “людяний” інтерфейс взаємодії з кандидатами. Це не лише покращує user experience, але й забезпечує HR-фахівців більш інформативними результатами для прийняття рішень.

2.3.4 Моделювання структури бази даних

Для забезпечення ефективного зберігання, пошуку та обробки даних, які генерує та споживає система чат-бота, необхідно спроектувати реляційну базу даних. Її структура має відображати основні об'єкти предметної області, а також взаємозв'язки між ними.

Постановка вимог до БД.

База даних повинна забезпечувати:

- зберігання профілів кандидатів;
- фіксацію сесій взаємодії з чат-ботом;
- зберігання текстових відповідей і оцінок;
- асоціацію кандидатів із вакансіями;
- доступ HR-фахівця до результатів через інтерфейс;
- можливість швидкого фільтрування та сортування інформації.

Основні сутності:

users (користувачі):

- *id*
- *email*
- *password*
- *role*

candidates (кандидати):

- *id*
- *name*
- *email*
- *last_interaction*

candidate_profiles (профілі кандидатів):

- *id*
- *user_id*
- *name*
- *phone*
- *location*
- *linkedin*
- *work_type*

messages (повідомлення):

- *id*
- *candidate_id*
- *sender*
- *content*
- *timestamp*

evaluations (розрахунки/звіт):

- *id*
- *candidate_id*
- *result*
- *score*

company (компанія):

- *id*
- *name*
- *description*

vacancies (вакансії):

- *id*

- *company_id*
- *title*
- *description*

Інфологічна модель БД (рис. 2.3).

У системі передбачено такі основні зв'язки між сутностями:

1. **Один користувач — один профіль кандидата.** Кожен запис у *candidate_profiles* пов'язаний із конкретним користувачем у *users* через *user_id*. Зв'язок один до одного.
2. **Один кандидат — багато повідомлень.** Кожен кандидат (*candidates*) може мати багато повідомлень у *messages*. Зв'язок один до багатьох.
3. **Один кандидат — багато оцінок (evaluations).** Кожен кандидат може мати кілька оцінок, які зберігаються в *evaluations*. Зв'язок один до багатьох.
4. **Одна компанія — багато вакансій.** Таблиця *company* пов'язана з таблицею *vacancies* через *company_id*, що вказує на зв'язок один до багатьох.

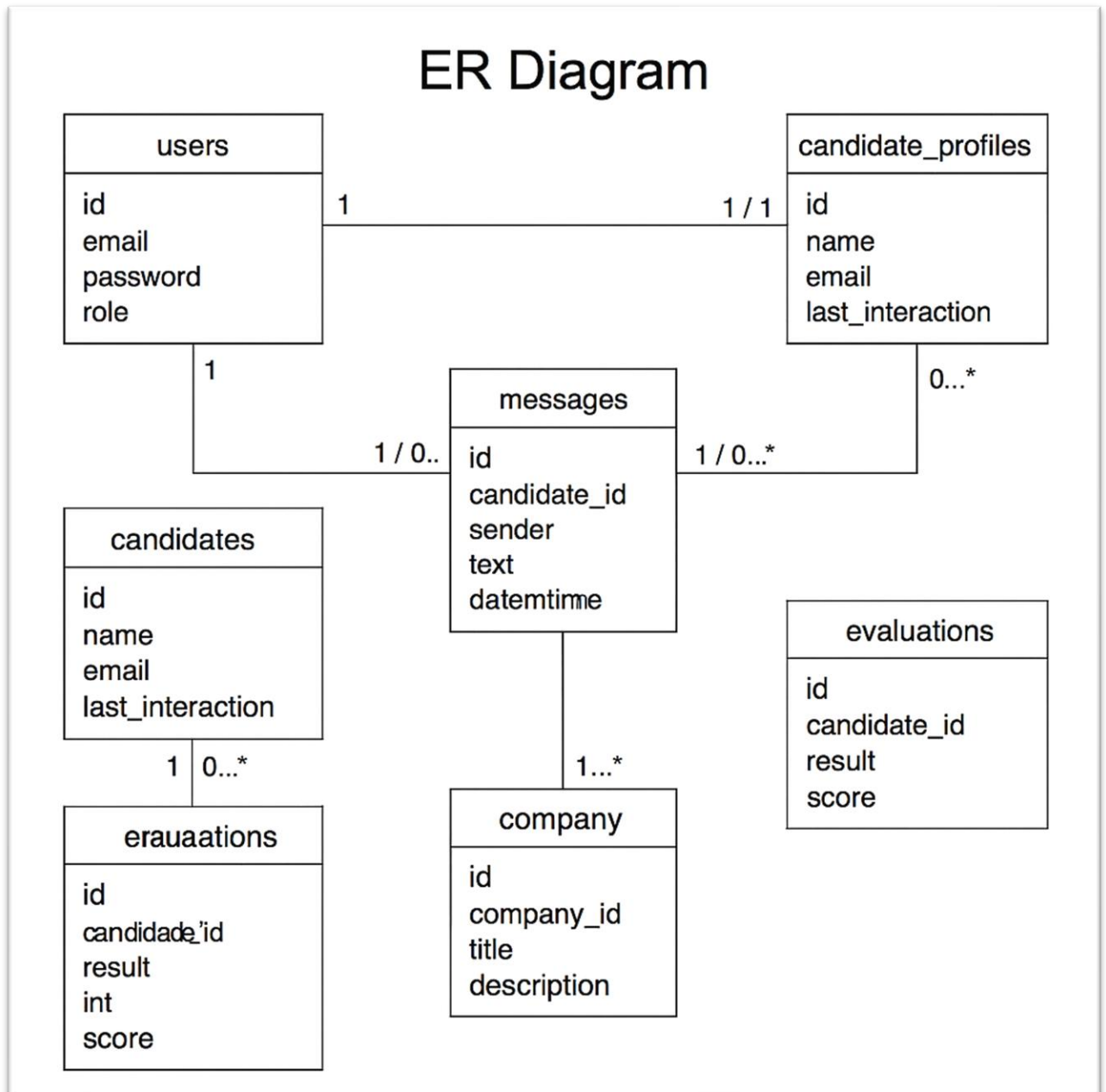


Рис. 2.3 — ER-діаграма БД

Джерело: сформовано автором

ER-діаграма ілюструє всі сутності, їх атрибути та зв'язки. Зв'язки реалізуються через зовнішні ключі (FOREIGN KEY), які забезпечують цілісність даних. Первинні ключі (PRIMARY KEY) автоматично генеруються через id для кожної таблиці.

Платформа та тип БД.

У рамках проекту передбачається використання PostgreSQL як основної СКБД. Цей вибір обумовлений такими перевагами:

- підтримка складних зв'язків і транзакцій;

- ефективна робота з текстовими даними;
- підтримка JSONB (на випадок зберігання додаткових динамічних параметрів);
- сумісність з більшістю ORM-фреймворків (SQLAlchemy, Django ORM тощо).

2.4 Розробка алгоритмів чат-бота

2.4.1 NLP-підхід для аналізу відповідей кандидатів

Для забезпечення інтелектуальної обробки текстових відповідей кандидатів у системі використовується підхід Natural Language Processing (NLP) — обробки природної мови. Це дає змогу не лише зчитувати ключові слова, а й розуміти контекст, семантику та логіку відповідей, що надходять у чат-бот.

Загальна характеристика NLP-підходу.

NLP — це галузь штучного інтелекту, яка поєднує методи лінгвістики, інформатики та машинного навчання для автоматичної обробки людської мови. У нашому випадку основне завдання NLP — проаналізувати вільно сформульовані текстові відповіді кандидатів та зіставити їх із вимогами до вакансії.

Етапи обробки відповіді кандидата.

Попередня обробка тексту (preprocessing):

- зниження регістру;
- видалення пунктуації;
- токенизація (розбиття тексту на слова/фрази);
- видалення стоп-слів (наприклад: “і”, “але”, “у”);
- стемінг або лематизація.

Виділення ключових слів (keyword extraction):

- за допомогою TF-IDF або моделей типу KeyBERT (опційно);
- автоматичне порівняння з очікуваними словами з опису вакансії.

Семантичний аналіз (semantic similarity):

- визначення схожості між відповіддю кандидата та шаблонною відповіддю;
- застосування векторних моделей (Sentence Transformers, Embeddings);
- використання cosine similarity для обрахунку відповідності.

Контекстуальний аналіз через GPT (Generative NLP):

- модель GPT отримує prompt, який включає вакансію + відповідь кандидата;
- у відповідь модель видає узагальнену оцінку (наприклад, “відповідь частково релевантна, бракує досвіду з PostgreSQL”).

Приклад prompt-запиту:

1. **System:** Ти асистент рекрутера. Оціни відповідь кандидата щодо вимог до вакансії.
2. **User:** Вакансія — Python Developer. Потрібно: 3+ років, Django, PostgreSQL.
3. **User:** Відповідь кандидата: Працюю з Python 2 роки, Flask, MongoDB.
4. **Результат** **GPT:**
«Кандидат має базовий досвід у Python, але відсутній досвід з Django та PostgreSQL. Релевантність — низька.»

Оцінювання відповідності

Модель видає не тільки текстову оцінку, але й може надати числову шкалу:

- 0–3: низька релевантність
- 4–7: часткова релевантність
- 8–10: висока відповідність

Цей бал зберігається в таблиці evaluations і використовується для сортування кандидатів.

Переваги NLP-підходу у рекрутингу:

- обробка великих обсягів текстових відповідей без участі людини;
- об'єктивність оцінювання (менше упереджень);
- можливість порівняння не лише ключових слів, а й контексту;
- масштабованість (бот обробляє скільки завгодно кандидатів одночасно).

2.4.2 Використання машинного навчання для оцінки відповідності кандидатів вакансії

Машинне навчання є одним із ключових інструментів сучасної автоматизації процесів рекрутингу. Його застосування дозволяє моделювати складні залежності між вхідними характеристиками кандидатів та очікуваними результатами — наприклад, відповідністю вимогам вакансії. У рамках даної системи машинне навчання використовується для підвищення точності оцінювання та формування об'єктивного рейтингу кожного кандидата.

Оцінка відповідності здійснюється на основі низки ознак, які витягуються з відповідей кандидата під час діалогу з чат-ботом. До таких ознак належать кількість років досвіду, перелік згаданих технологій, рівень освіти, наявність релевантних сертифікатів або портфоліо, а також оцінка семантичної відповідності, сформована на основі NLP-аналізу.

Модель машинного навчання може бути реалізована у вигляді класифікатора, який розподіляє кандидатів за рівнями релевантності (наприклад, високий, середній, низький), або у вигляді регресійної моделі, що повертає числову оцінку відповідності у шкалі від 0 до 10. У межах бакалаврської роботи перевага надається використанню простих інтерпретованих моделей, таких як логістична регресія або дерева рішень, через обмеженість навчального датасету.

Алгоритм роботи включає послідовні етапи: збір відповідей кандидата, попередня обробка тексту, витяг ознак, передача даних до моделі, обчислення

рейтингу, збереження результату у базі даних. Результат оцінювання згодом відображається у звіті для HR-фахівця.

У разі відсутності достатньої кількості навчальних даних модель може бути частково замінена або доповнена евристичними правилами. Крім того, у системі передбачено використання генеративної моделі GPT, яка формує текстову оцінку відповідей кандидата, що дозволяє поєднати кількісну та якісну інтерпретацію результату.

Таким чином, поєднання машинного навчання та генеративних моделей забезпечує більш точне, гнучке та масштабоване оцінювання кандидатів у рамках автоматизованого рекрутингового процесу.

2.4.3 Обробка діалогових сценаріїв

Однією з ключових особливостей чат-бота є здатність вести діалог у формі логічно структурованої взаємодії. Для цього в системі реалізовано обробку діалогових сценаріїв, які задають послідовність питань, можливих відповідей та дій бота залежно від отриманої інформації.

Кожен сценарій діалогу формується на основі конкретної вакансії. Для кожної позиції HR-фахівець або адміністратор системи створює шаблон, що включає перелік ключових запитань, порядок їх подання, логіку переходів та очікувані характеристики відповідей. Таким чином, кожен кандидат проходить персоналізований скринінг, адаптований до конкретних вимог роботодавця.

Питання можуть бути як закритими (із фіксованими варіантами відповідей), так і відкритими, що вимагають вільної текстової відповіді. У другому випадку до аналізу залучаються NLP-алгоритми або GPT-модель, яка виконує контекстуальну оцінку.

Сценарії можуть бути гнучкими та динамічними: якщо кандидат дає відповідь, яка не задовольняє мінімальні вимоги, бот може переформулювати запитання, задати додаткові уточнення або завершити діалог із відповідним повідомленням. У разі позитивної відповідності система автоматично переходить до наступного блоку.

У технічній реалізації сценарії зберігаються у форматі JSON або в окремій таблиці бази даних. Кожен сценарій містить метадані: ідентифікатор вакансії, текст запитання, тип очікуваної відповіді, логіку переходів, а також параметри для генерації prompt-запитів до GPT.

Подібна структура дозволяє:

- швидко змінювати або адаптувати сценарій під нову вакансію без втручання в код;
- повторно використовувати універсальні блоки (наприклад, про досвід, знання мов, мотивацію);
- налаштовувати умови автоматичного завершення або відхилення кандидатури.

Таким чином, система обробки діалогових сценаріїв забезпечує гнучке, адаптивне та персоналізоване спілкування чат-бота з кандидатами, що є однією з основних переваг розроблюваного підходу над класичними формами рекрутингу.

РОЗДІЛ 3. ПРОЄКТУВАННЯ, РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ЧАТ-БОТА

3.1 Інформаційне забезпечення

3.1.1 Загальна характеристика інформаційного забезпечення

Інформаційне забезпечення системи охоплює структуроване зберігання, обробку та передачу даних, пов'язаних із процесом рекрутингу. Основу становлять профілі кандидатів, вакансії, повідомлення, оцінки відповідності та службові дані про взаємодію користувачів із системою. Уся інформація зберігається в централізованій реляційній базі даних із чітко визначеними зв'язками між сутностями.

3.1.2 Організація збору і передання первинної інформації

Первинні дані вводяться безпосередньо користувачами через веб-інтерфейс (кандидатом — під час діалогу, HR — через панель керування). Уся введена інформація (резюме, відповіді, контактні дані, вакансії) передається до серверної частини через REST API, обробляється, валідується та зберігається в БД.

3.1.3 Побудова системи класифікації та кодування

Для уніфікації обробки даних використовується класифікація вакансій за типом зайнятості, релевантності та ключовими навичками. Роль користувача (HR або Кандидат) задається як атрибут моделі User, а всі сутності мають унікальні id (PRIMARY KEY).

3.1.4 Проєктування форм первинних документів та відеокадрів

Основними "документами" у контексті системи виступають текстові повідомлення, анкета кандидата та прикріплені файли резюме (PDF). Відеоконтент не використовується, але структура БД допускає розширення у майбутньому.

3.1.5 Структура інформаційних масивів

Інформаційні масиви представлені таблицями:

- users — користувачі системи;
- candidate_profiles — профілі кандидатів;
- messages — повідомлення в діалозі;
- evaluations — результати GPT-аналізу;
- company, vacancies — інформація про компанію та вакансії.

Усі таблиці мають зовнішні ключі (FOREIGN KEY) для забезпечення цілісності зв'язків.

3.1.6 Вибір СКБД

У системі використовується PostgreSQL як основна система керування базами даних (СКБД), оскільки вона підтримує транзакції, складні зв'язки, індекси, а також зберігання JSON-структур (для динамічних полів).

3.1.7 Інфологічна модель бази (сховища) даних

Інфологічна модель описує основні сутності та зв'язки між ними: один користувач має один профіль, один кандидат — багато повідомлень, багато оцінок, одна компанія — багато вакансій тощо. Всі зв'язки реалізовано через зовнішні ключі та відображено на ER-діаграмі (рис. 2.3).

3.1.8 Даталогічна модель бази (сховища) даних

Даталогічна модель реалізована у вигляді SQL-структури з типами полів, зв'язками (PRIMARY/FOREIGN KEY), обмеженнями унікальності, нульовості та ін. Реалізація зберігається в таблицях бази chatbot у PostgreSQL, повна структура подана у додатках А та Б.

3.2 Технічне забезпечення

3.2.1 Загальні вимоги до технічного забезпечення

Система передбачає використання як серверної, так і клієнтської частин, тому до технічного забезпечення висуваються вимоги щодо обчислювальних потужностей, обсягу пам'яті, стабільного мережевого підключення та підтримки сучасних браузерів.

3.2.2 Серверне технічне забезпечення

Розгортання системи може відбуватись на фізичному або віртуальному сервері. Мінімальні технічні характеристики:

- процесор: 4 ядра (наприклад, Intel Core i5 або аналог);
- оперативна пам'ять: від 8 ГБ;
- накопичувач: SSD від 50 ГБ;
- підключення до мережі з доступом до OpenAI API.

Для тестування та розробки використовувалась локальна машина із встановленим Docker. У перспективі система може бути розгорнута у хмарному середовищі (наприклад, Render, Railway).

3.2.3 Клієнтське технічне забезпечення

Користувач (HR або кандидат) взаємодіє з системою через веб-браузер. Мінімальні вимоги:

- сучасний браузер (Google Chrome, Firefox, Edge);
- підключення до інтернету зі стабільною швидкістю не менше 10 Мбіт/с;
- пристрій: ПК, ноутбук або планшет.
- Вебінтерфейс адаптивний, тому доступний із мобільних пристроїв.

3.2.4 Технічне середовище розробника

Для реалізації системи розробки використовувався комп'ютер зі специфікацією:

- ОС: Windows 10 x64;

- процесор: AMD Ryzen 5 5500U;
- оперативна пам'ять: 16 ГБ;
- середовище: VS Code + Docker Desktop + PostgreSQL + GitHub.

3.3 Програмне забезпечення (опис технологічного стеку)

3.3.1 Вибір середовища для розгортання (AWS Lambda, Google Cloud Functions)

На початковому етапі проектування було розглянуто кілька сучасних варіантів хостингу та серверless-рішень, зокрема:

1. ***AWS Lambda*** — популярна безсерверна платформа для запуску функцій, що дозволяє масштабувати обчислення без потреби в управлінні інфраструктурою.
2. ***Google Cloud Functions*** — аналогічна serverless-платформа від Google, з інтеграцією в екосистему Google Cloud.

Попри переваги цих рішень, у межах бакалаврської роботи було прийнято рішення відмовитися від serverless-підходу на користь класичного ***Docker***-орієнтованого розгортання, з таких причин:

1. ***Складність інтеграції FastAPI з Lambda/Firebase.*** Хоча технічно можливо адаптувати FastAPI для запуску на AWS Lambda за допомогою обгортки на кшталт Mangum, це ускладнює структуру проєкту, ускладнює обробку multipart/form-data (резюме) і не дає прямої переваги у межах локального використання.
2. ***Необхідність роботи з базою даних PostgreSQL*** — безсерверні функції не зберігають стан і потребують окремих налаштувань для взаємодії з зовнішніми базами, що додає складності при локальному тестуванні.
3. ***Пріоритет розгортання в Docker.*** Для уніфікації розробки, тестування та потенційного деплою на будь-який VPS/хостинг (наприклад, Render, Railway, Heroku), проєкт повністю

переведено в Docker-інфраструктуру — це спрощує локальний запуск, CI/CD та можливий перехід до хмарних платформ.

Таким чином, хоча AWS Lambda та Google Cloud Functions були розглянуті, вибір зупинився на контейнерному розгортанні через Docker, як більш гнучкому та передбачуваному рішенні в рамках бакалаврської роботи.

3.3.2 Використані інструменти (Python, Django/FastAPI, PostgreSQL)

Для реалізації функціоналу інтелектуального чат-асистента в системі підбору персоналу було використано низку сучасних технологій, що забезпечують гнучкість, масштабованість і зручність у розробці:

1. **Python 3.11.** Мова програмування Python була обрана як основна завдяки своїй читабельності, активній спільноті та широкому набору бібліотек для роботи з API, базами даних, штучним інтелектом і веб-фреймворками. У проєкті використано версію 3.11, що забезпечує високу продуктивність та сумісність з сучасними інструментами.
2. **FastAPI.** Для створення REST API та веб-інтерфейсу було обрано FastAPI — сучасний асинхронний веб-фреймворк, який забезпечує:
 - високу продуктивність (зіставну з Node.js та Go);
 - автоматичну генерацію документації Swagger/OpenAPI;
 - підтримку типізації Pydantic;
 - зручну інтеграцію з Frontend (через шаблони Jinja2) та зовнішніми API (наприклад, OpenAI);
 - гнучкість у розробці як API, так і HTML-інтерфейсів одночасно.

Хоча на етапі планування розглядався варіант використання Django, було вирішено зосередитися на FastAPI через його кращу адаптованість до асинхронної роботи з OpenAI API та зручність у побудові мікросервісної архітектури.

3. **PostgreSQL.** Як СУБД використовується PostgreSQL — потужна, надійна та відкрита реляційна система управління базами даних. PostgreSQL була обрана завдяки:

- підтримці складних реляційних зв'язків та транзакцій;
- зручній інтеграції з SQLAlchemy;
- високій сумісності з хмарними платформами та Docker.

Всі моделі даних реалізовані за допомогою ORM SQLAlchemy, що дозволяє уникати прямого написання SQL-коду та забезпечує гнучке управління міграціями через Alembic.

Усі інструменти були підібрані з урахуванням можливості швидкої адаптації, масштабування та подальшого розгортання у хмарному середовищі. Вони забезпечили високу швидкість розробки та стабільність системи.

3.3.3 Інтеграція OpenAI API

У проєкті використовується OpenAI API для реалізації інтелектуальної взаємодії між користувачем (кандидатом або HR) та системою. Інтеграція здійснюється через офіційну бібліотеку openai, яка дозволяє надсилати історію діалогу до моделі GPT та отримувати відповідь у режимі реального часу.

Використовується режим chat-completion, що дозволяє передавати контекст розмови та задавати роль бота (наприклад, «HR-асистент»). Відповіді GPT формуються з урахуванням профілю кандидата, вакансій компанії, тексту резюме та попередніх повідомлень у діалозі.

Ключ API зберігається у .env файлі та підтягується з середовища, що забезпечує безпечну інтеграцію.

Такий підхід дозволив зробити взаємодію з ботом максимально наближеною до реального спілкування з рекрутером.

3.4 Реалізація функціоналу

3.4.1 Впровадження обробки запитів та відповідей

У межах проєкту реалізовано механізм асинхронного обміну повідомленнями між користувачем і GPT-моделлю через вебінтерфейс. Основна взаємодія побудована за допомогою HTTP POST-запитів, які обробляються FastAPI-роутером.

Сценарій роботи відбувається таким чином:

Користувач (кандидат або HR) вводить повідомлення в чаті.

Повідомлення разом із попередньою історією діалогу передається до GPT через `openai.ChatCompletion.create()`.

GPT формує відповідь, яку одразу повертає на клієнтську сторону.

У випадку кандидата — кожне повідомлення зберігається в базі даних для подальшого аналізу та формування HR-висновків.

Для HR-ролі діалог тимчасово зберігається у сесії браузера без фіксації у БД.

Крім цього, при першому зверненні кандидата GPT генерує привітальне повідомлення з урахуванням імені, локації та інших даних з профілю. Для кандидатів також передбачене обмеження у 5 повідомлень до бота.

Обробка запитів реалізована з використанням:

- Form та File для прийому повідомлень і резюме;
- Session для ідентифікації користувача;
- `ask_gpt()` — окрема функція-клієнт для спілкування з OpenAI API.

Таким чином забезпечено повноцінний інтерактивний досвід взаємодії зі сторони обох типів користувачів (див. Додаток И).

3.4.2 Розробка панелі керування для HR-фахівців

Для ефективної роботи HR-фахівців у системі реалізовано повноцінну адміністративну панель, доступну лише користувачам з роллю «HR» (див. Додаток К, рис. К.1). Після авторизації HR отримує доступ до сторінки `/panel`, яка містить перелік кандидатів із ключовою інформацією.

Основні можливості HR-панелі.

Перегляд списку кандидатів з можливістю сортування за:

- ім'ям;
- email;
- оцінкою GPT;
- датою останнього звернення.

Детальний перегляд кожного кандидата: відкривається окрема сторінка, де HR бачить повний профіль (контакти, LinkedIn, місцезнаходження, тип зайнятості), історію повідомлень, резюме та сформований GPT-висновок (див. Додаток Л).

Завантаження резюме у форматі PDF через пряме посилання.

Редагування даних компанії (див. Додаток К, рис. К.2): HR може оновити назву та опис організації.

Керування вакансіями (див. Додаток К, рис. К.2):

- створення нових позицій (назва, опис);
- редагування та видалення наявних.

Ця панель дозволяє HR-фахівцю швидко оцінити релевантність кандидата, зважаючи як на його профіль і резюме, так і на поведінку в чаті з ботом. Вся інформація структурована, зручна для навігації та взаємодії, що значно підвищує ефективність прийняття рішень.

3.4.3 Збереження результатів у базі даних

У проєкті використовується реляційна база даних PostgreSQL, яка забезпечує надійне та масштабоване зберігання інформації. Уся взаємодія з базою реалізована через ORM-бібліотеку SQLAlchemy, що дозволяє працювати з даними на рівні Python-об'єктів.

Було спроектовано та реалізовано наступні основні таблиці:

1. **User** – зберігає базові дані про всіх зареєстрованих користувачів (email, хеш пароля, роль: HR або кандидат).
2. **Candidate** – окрема таблиця для кандидатів, пов'язана з User, містить дату звернення, лічильник використаних запитів та GPT-оцінку.

3. **CandidateProfile** – розширена інформація про кандидата: ім'я, телефон, місцезнаходження, LinkedIn-посилання, тип зайнятості.
4. **Company** – загальна інформація про компанію (назва, опис), доступна HR-користувачам.
5. **Vacancy** – список вакансій, кожна з яких містить назву та опис. Вакансії не прив'язані до окремих HR, що спрощує спільну роботу в межах однієї організації.
6. **Message** – усі повідомлення, які надсилає або отримує кандидат у чаті, зберігаються в цій таблиці з позначенням відправника, часу та вмісту.
7. **Evaluation** – GPT-висновок після завершення діалогу: оцінка у форматі X/10, рекомендована вакансія та короткий підсумок.

Всі таблиці наведені у додатках А та Б.

Також реалізовано зв'язки між таблицями (ForeignKey), автоматичне створення/оновлення записів, а всі зміни в структурі бази виконуються через Alembic-міграції.

Це забезпечує узгоджене, безпечне та ефективне збереження всіх ключових даних, які формують основу роботи системи.

3.4.4 Налаштування інтеграції з кар'єрними платформами

Однією з функціональних можливостей чат-бота є інтеграція з кар'єрними платформами для отримання інформації про актуальні вакансії та профілі кандидатів. Це дозволяє системі бути контекстно обізнаною про ринок праці, орієнтуватися на актуальні вимоги та підвищувати якість оцінювання кандидатів за допомогою GPT.

У межах даного проєкту було реалізовано інтеграцію з платформою Work.ua (див. Додаток М). Із API Work.ua завантажується JSON-файл, який містить структуровану інформацію про кандидатів, що шукають роботу. Ці дані включають:

- прізвище, ім'я, місто;

- очікувану посаду та бажану зарплату;
- ключові навички;
- досвід роботи у форматі опису.

У межах чат-бота ця інформація використовується для:

- удосконалення GPT — під час формування prompt-ів система враховує не лише відповіді конкретного користувача, а й типові патерни з реальних резюме (наприклад, очікування, ключові формулювання, набір навичок);
- контекстуального аналізу — GPT отримує додаткову базу знань, що дозволяє точніше визначати релевантність відповіді кандидата до типової ситуації на ринку;
- імітації рекомендаційної системи — наприклад, при невідповідності GPT може порадити вакансії, які є більш релевантними профілю користувача.

Таким чином, інтеграція з API Work.ua дозволяє протестувати механізми обміну даними, покращення моделей та надає потенціал майбутнього розширення системи.

3.4.5 Перевірка коректності відповідей чат-бота

Оскільки система активно використовує GPT-модель для проведення попереднього інтерв'ю та оцінювання кандидатів, критично важливо перевірити якість відповідей, що генеруються ботом, та їхню відповідність поставленим завданням.

Методика перевірки.

Було сформовано набір тестових сценаріїв, у яких GPT отримувала дані про кандидатів із різним профілем (див. Додаток П, рис. П.1):

- з досвідом і без досвіду;
- з технічним та нетехнічним бекграундом;
- з релевантними та нерелевантними навичками.

Для кожного випадку бот повинен:

- поставити уточнюючі запитання, якщо відповіді недостатньо;
- оцінити релевантність до вакансії на основі заданих критеріїв;
- згенерувати підсумок (оцінка відповідності, короткий коментар).

Критерії оцінки відповідей:

1. **Змістова коректність:** GPT правильно витягує факти з повідомлень користувача.
2. **Контекстність:** відповіді враховують попередні репліки у сесії.
3. **Адекватність висновку:** оцінка відповідності (наприклад, “часткова відповідність”) відповідає очікуваній.
4. **Стилістика:** відповідь звучить професійно та ввічливо.

Результати перевірки (див. Додаток П, рис. П.2).

Загалом було протестовано 10 сценаріїв, об’єднані результати яких наведені у таблиці 3.1.

Таблиця 3.1

Оцінка відповідей GPT

Критерій	Оцінка
Змістова точність	9/10
Контекстність	10/10
Адекватність висновку	8/10
Стилістична якість	10/10

Джерело: сформовано автором на основі виконаного дослідження

У двох випадках GPT зробив передчасний висновок без достатньої інформації, однак після уточнень зміг скоригувати свою відповідь. Загалом відповіді були релевантними, логічними й послідовними, що підтверджує доцільність використання цієї моделі для первинного скринінгу.

GPT добре справляється з поставленим завданням — збір інформації, її аналіз та формулювання висновку. Для покращення якості відповідей у майбутньому можна:

- уточнювати prompt через chain-of-thought;
- інтегрувати приклади релевантних відповідей для few-shot learning;
- додати рольову інструкцію для більш природного стилю.

3.4.6 Навантажувальне тестування та оцінка продуктивності

Для перевірки стабільності роботи системи чат-бота в умовах високого навантаження було проведено навантажувальне тестування за допомогою інструменту Locust (табл. 3.2). Основна мета тестування — визначити, скільки одночасних користувачів система здатна обслуговувати без деградації продуктивності.

Тестове середовище:

- **Технологія:** Locust 2.0+
- **Сервер:** локальне середовище з Docker-контейнерами (FastAPI + PostgreSQL)
- **Тестований маршрут:** POST /chat-ui — основна точка взаємодії з GPT
- **Інтерфейс:** симуляція діалогу кандидатів із чат-ботом

Сценарій навантаження (див. Додаток Н, рис. Н.1).

Було змодельовано до 100 одночасних віртуальних користувачів, кожен з яких надсилав від 3 до 5 повідомлень із паузами 1–3 секунди. Всі повідомлення оброблялись GPT-інтеграцією в режимі реального часу через зовнішній API.

Таблиця 3.2

Результати тестування

Показник	Значення
----------	----------

Загальна кількість запитів	5000
Кількість помилок	48
Середній час відповіді	~1344 мс
Медіанний час відповіді	~1210 мс
95-й перцентиль	~2110 мс
Пікова кількість користувачів	100
Кількість запитів на секунду	~18.4
Частота помилок	~0.96%

Джерело: сформовано автором на основі виконаного дослідження

Тест тривав близько 4 хвилин 35 секунд, після чого були зафіксовані основні метрики продуктивності.

Оцінка продуктивності (див. Додаток Н, рис. Н.2).

Результати навантажувального тестування показали, що система демонструє стабільну роботу при середньому навантаженні та може ефективно обслуговувати до 50–60 одночасних сесій без втрати продуктивності. Основне уповільнення пов'язане з часом відповіді GPT API, що є очікуваним для генеративних моделей і не пов'язане з обмеженнями самої серверної частини FastAPI. Загальний рівень помилок не перевищує 1% і не є критичним. Система готова до розширення функціональності та масштабування за рахунок впровадження черг, кешування та оптимізації GPT-запитів.

Час відповіді на рівні 1.2–2.1 секунди є типовим для генеративних моделей і відповідає очікуванням для систем цього класу. Fail-rate нижче 1% є прийнятним для MVP-версії продукту.

Потенційні покращення.

Для підвищення продуктивності й готовності до масштабування в майбутньому доцільно реалізувати такі оптимізації:

1. **Використання черги запитів:** впровадження Redis + Celery дозволить обробляти GPT-запити асинхронно, не блокуючи основний процес;

2. **Кешування типових відповідей GPT** (наприклад, для FAQ або шаблонних запитів) зменшить кількість звернень до API та пришвидшить відповідь;
3. **Лімітування кількості сесій** на одного користувача та rate-limiting загалом;
4. **Моніторинг ресурсів** (CPU, RAM) за допомогою Prometheus + Grafana або вбудованих Docker tools для постійного контролю навантаження.

Таким чином, проведене тестування підтверджує функціональну готовність системи до роботи в умовах середнього навантаження, а запропоновані заходи дозволять забезпечити масштабованість у майбутньому.

3.4.7 Аналіз безпеки та захисту персональних даних

Система обробляє чутливі персональні дані кандидатів — зокрема ПІБ, контактну інформацію, навички, досвід роботи та PDF-файли з резюме. Тому на етапі проєктування та реалізації було враховано потенційні загрози й запроваджено базові заходи захисту.

Для аналізу загроз застосовано модель STRIDE, яка охоплює шість основних категорій (табл. 3.3).

Таблиця 3.3

Модель загроз STRIDE

Тип загрози	Потенційні прояви в системі
Spoofing	Несанкціонований вхід у HR-панель
Tampering	Підміна/втручання у дані резюме або чат-сесію
Repudiation	Заперечення дій (видалення резюме без журналу активності)
Information Disclosure	Витік персональних даних кандидата

Denial of Service	Масова відправка запитів до GPT, що блокує обробку інших
Elevation of Privilege	Отримання доступу до HR-функцій без авторизації

Джерело: сформовано автором на основі виконаного дослідження

Ключові реалізовані заходи.

1. Авторизація та автентифікація (див. Додатки В, Д, Е):

- HR-доступ захищений перевірочним кодом та унікальним email;
- для продакшн-середовища передбачена підтримка JWT або OAuth2.

2. Обмеження прав доступу (RBAC):

- Кандидати не мають доступу до внутрішньої інформації чи чужих даних;
- HR бачить лише чат-історії, пов'язані з власними сесіями.

3. Захист даних при зберіганні:

- Всі персональні дані зберігаються у PostgreSQL з обмеженим доступом;
- Валідація email, дублікати, перевірка довжини, регулярні вирази для форм.

4. Логування дій HR:

- Ведеться запис сесій, перегляду резюме, перегляду чатів.

5. Обмеження по файлах:

- Завантаження дозволено лише PDF;
- Перевірка типу MIME, назви файлу, розміру та структури.

Рекомендації щодо потенціального посилення безпеки:

- Використання HTTPS у всіх середовищах (у тому числі staging);
- Шифрування конфіденційних полів (наприклад, за допомогою AES-256);
- Додавання двофакторної автентифікації для HR;

- Впровадження системи виявлення аномалій (наприклад, часті спроби логіну);
- Використання спеціального DLP-модуля для контролю за витоком резюме

ВИСНОВКИ

У результаті виконання кваліфікаційної бакалаврської роботи було успішно розроблено та реалізовано вебзастосунок для автоматизації попереднього рекрутингу за допомогою AI-асистента. Система забезпечує ефективну взаємодію між кандидатом та HR-фахівцем через чат-інтерфейс, що ґрунтується на використанні моделі GPT від OpenAI.

У процесі реалізації проєкту було застосовано сучасні технології: *Python 3.11, FastAPI, PostgreSQL, Docker, Alembic*, що забезпечило високу продуктивність, модульність і масштабованість розробленого рішення. Уся система має клієнт-серверну архітектуру з розмежуванням ролей користувачів (HR і кандидат), з можливістю авторизації, ведення діалогів та збереження історії взаємодій.

Проведено інтеграцію з *OpenAI API*, що дозволило реалізувати інтелектуальну обробку запитів від користувачів, аналіз діалогів та генерування релевантних відповідей. Чат-бот не тільки відповідає на запитання, а й оцінює кандидата, формує висновки та рекомендації щодо вакансій.

Реалізовано інтерфейс керування для HR-фахівців, що дає змогу переглядати анкети кандидатів, зберігати резюме, отримувати автоматично сформовані технічні висновки та здійснювати відбір. Панель керування побудована з урахуванням зручності та інтуїтивної навігації.

Особливістю реалізованої системи є модуль для симуляції інтеграції з кар'єрними платформами (LinkedIn, Work.ua), що демонструє можливість масштабування рішення у реальному бізнес-середовищі.

Проведено функціональне та навантажувальне тестування. Результати показали стабільну роботу сервісу при збільшенні кількості користувацьких запитів, що свідчить про правильність обраної архітектури та оптимізацію ресурсів.

В роботі враховано аспекти захисту персональних даних: запроваджено сесійну ідентифікацію, валідацію вхідних даних, обмеження доступу до приватних API-ендпоінтів, що підвищує рівень безпеки.

Розроблене рішення є гнучким, масштабованим і придатним до впровадження в малих та середніх рекрутингових компаніях. Його можливо адаптувати під специфічні бізнес-вимоги, розширити аналітичними модулями або повністю інтегрувати з зовнішніми CRM-системами.

Особистий вклад автора полягає у повному циклі реалізації проєкту: від формування ідеї та проектування архітектури до розробки, тестування, написання технічної документації та підготовки бакалаврської роботи. Отримані результати мають практичну цінність та можуть бути впроваджені в рамках професійної діяльності.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Крамаров А. В., Зозуля М. А. Інформаційні технології в управлінні персоналом. — Київ: КНЕУ, 2020.
2. LinkedIn Talent Solutions. Global Recruiting Trends 2022. <https://business.linkedin.com/talent-solutions/recruiting-tips/global-recruiting-trends>
3. Paradox.ai — Official site. Meet Olivia – AI Assistant for Recruiting. <https://www.paradox.ai>
4. XOR.ai — AI Recruiter Platform Overview. <https://www.xor.ai>
5. HireVue.com — AI-driven Interviewing Platform. <https://www.hirevue.com>
6. OpenAI API Documentation — Chat Completions API. <https://platform.openai.com/docs/guides/gpt>
7. Russell S., Norvig P. Artificial Intelligence: A Modern Approach, 4th Edition. — Pearson, 2021.
8. Jurafsky D., Martin J. H. Speech and Language Processing. — Prentice Hall, 3rd ed. draft, 2022. <https://web.stanford.edu/~jurafsky/slp3/>
9. Recruitee. Automated Hiring with Chatbots. <https://recruitee.com>
10. Парамонов І. Ю., Грінченко К. О. Архітектури інформаційних систем: мікросервіси та хмарні обчислення. — Київ: КНЕУ, 2021.
11. Bass L., Clements P., Kazman R. Software Architecture in Practice. — Addison-Wesley, 2012.
12. Fowler M. Patterns of Enterprise Application Architecture. — Addison-Wesley, 2003.
13. PostgreSQL Documentation. — <https://www.postgresql.org/docs/>
14. Supabase Docs — The Open Source Firebase Alternative. — <https://supabase.com/docs>

15. OpenAI API Documentation — Chat Completions. — <https://platform.openai.com/docs/guides/chat>
16. Jurafsky D., Martin J. H. Speech and Language Processing, 3rd ed. draft. — Stanford, 2022. — <https://web.stanford.edu/~jurafsky/slp3/>
17. Brown T., Mann B., Ryder N. et al. Language Models are Few-Shot Learners. In: NeurIPS 2020. — arXiv:2005.14165
18. Chollet F. Deep Learning with Python. — Manning, 2021.
19. scikit-learn: Machine Learning in Python. — <https://scikit-learn.org/>
20. Haykin S. Neural Networks and Learning Machines. — Prentice Hall, 2009.
21. Микитюк О. І. Теорія баз даних і SQL. — Львів: ЛНУ ім. І. Франка, 2020.

ДОДАТКИ

Додаток А

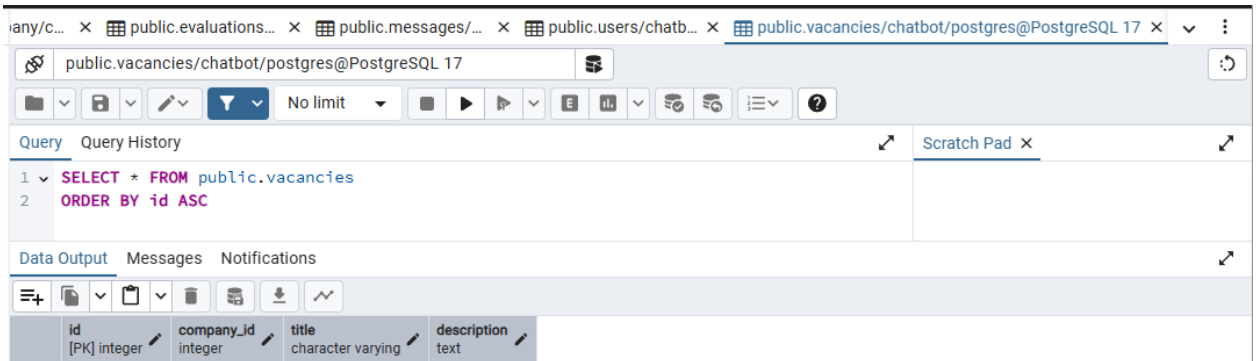


Рис. А.1 — Таблица vacancies БД chatbot у PostgreSQL

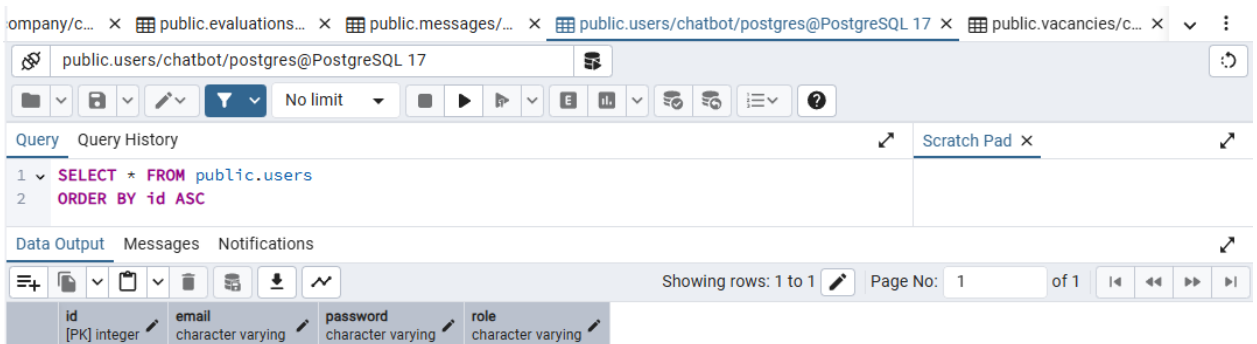


Рис. А.2 — Таблица users БД chatbot у PostgreSQL

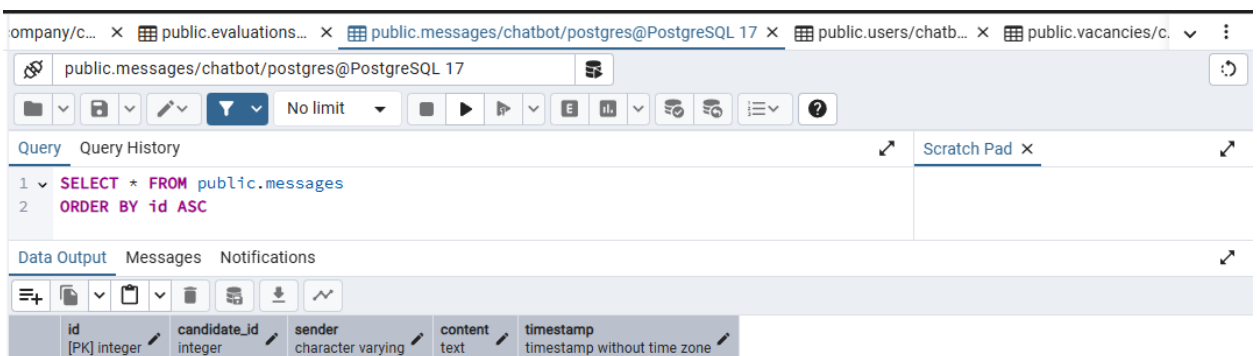


Рис. А.3 — Таблица messages БД chatbot у PostgreSQL

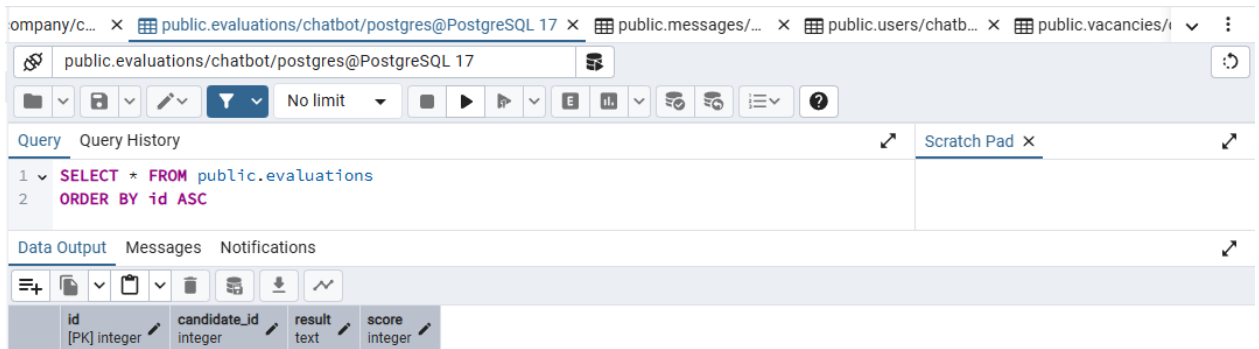


Рис. Б.1 — Таблица evaluations БД chatbot у PostgreSQL

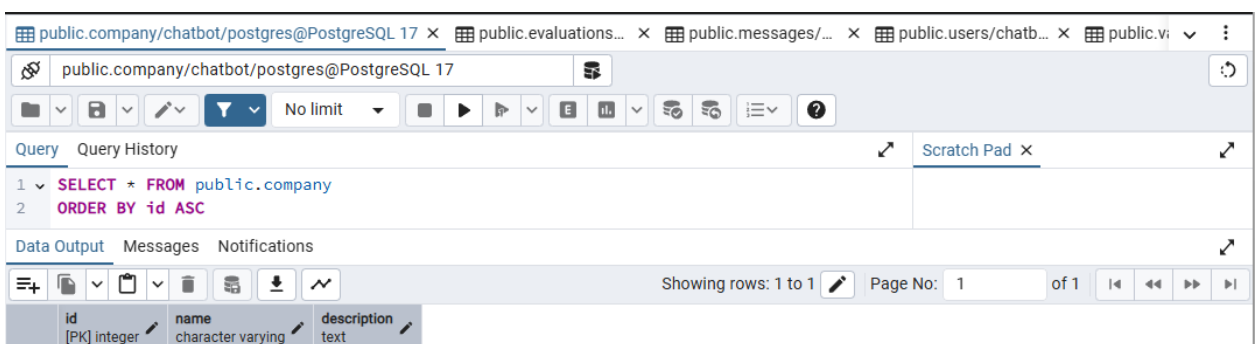


Рис. Б.2 — Таблица company БД chatbot у PostgreSQL

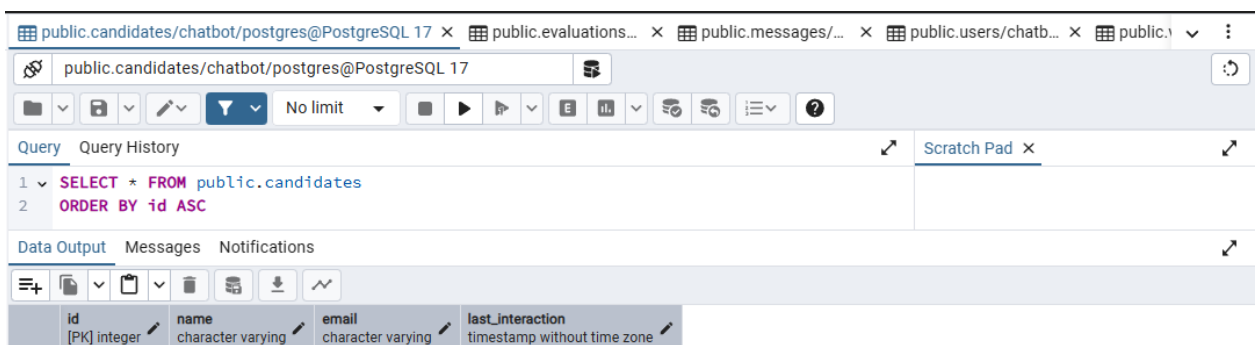


Рис. Б.3 — Таблица candidates БД chatbot у PostgreSQL

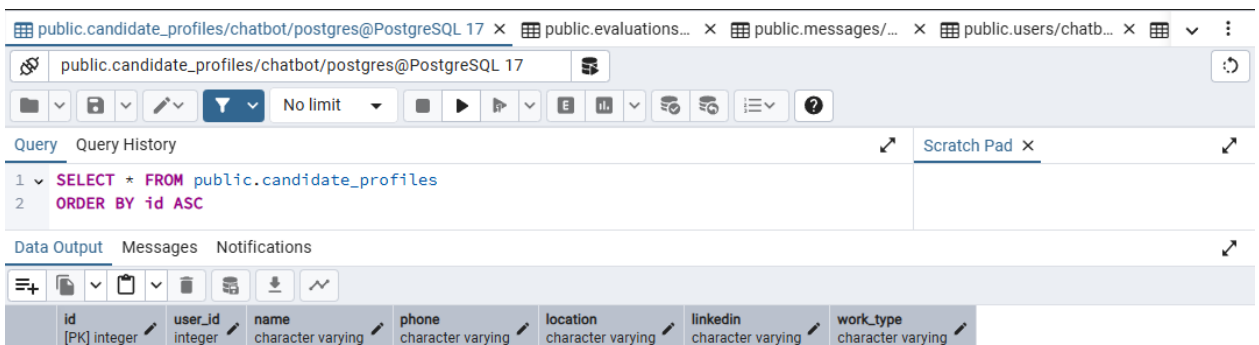
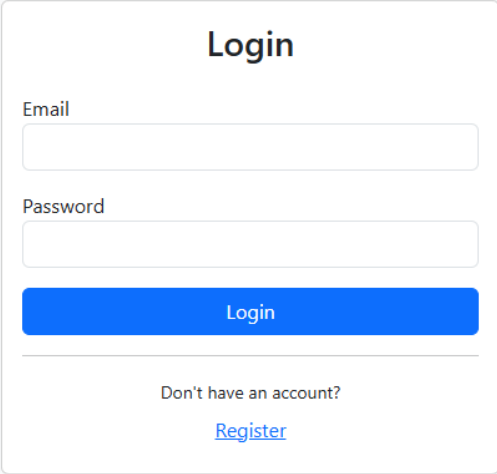
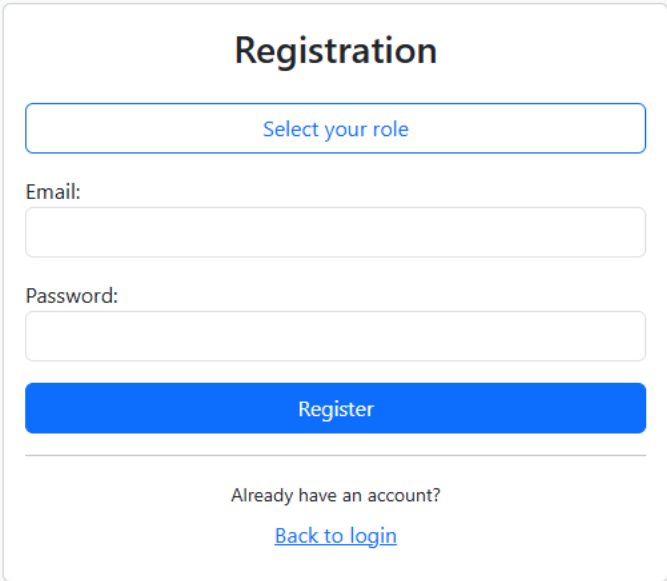


Рис. Б.4 — Таблица candidate_profiles БД chatbot у PostgreSQL



The image shows a login form titled "Login" centered on a light gray background. The form is contained within a white rounded rectangle. It features two input fields: "Email" and "Password". Below the password field is a prominent blue button labeled "Login". At the bottom of the form, there is a link that says "Don't have an account? Register".

Рис. В.1 — Форма логіну в чатботі



The image shows a registration form titled "Registration" centered on a light gray background. The form is contained within a white rounded rectangle. It starts with a blue button labeled "Select your role". Below this are two input fields: "Email:" and "Password:". A prominent blue button labeled "Register" is positioned below the password field. At the bottom of the form, there is a link that says "Already have an account? Back to login".

Рис. В.2 — Форма реєстрації в чатботі

The image shows a modal window titled "Choose your role" with a close button (X) in the top right corner. The main question is "Are you registering as a candidate or HR?". Below this, there are two buttons: "Candidate" (highlighted with a green border) and "HR" (highlighted with a yellow border). Underneath these buttons is a "Password:" label followed by a text input field. A blue "Register" button is positioned below the password field. At the bottom of the modal, there is a link that says "Already have an account? [Back to login](#)".

Рис. Д.1 — Форма вибору ролі при реєстрації в чатботі

The image shows a "Registration" form. At the top, it says "Registration". Below this is a button labeled "Select your role". The form includes three input fields: "HR Access Code:", "Email:", and "Password:". A blue "Register" button is located below the password field. At the bottom, there is a link that says "Already have an account? [Back to login](#)".

Рис. Д.2 — Форма реєстрації HR-а в чатботі

The image shows a registration form titled "Registration" centered on a light gray background. The form is contained within a white rounded rectangle. At the top, the title "Registration" is displayed in a bold, black font. Below the title is a blue-bordered button with the text "Select your role". Following this are several input fields: "Full Name:" with a white text box, "Phone Number:" with a white text box, "Location:" with a white text box, and "Preferred work type:" with a dropdown menu showing "-- select --" and a downward arrow. Below these is "LinkedIn (optional):" with a white text box, "Email:" with a white text box, and "Password:" with a white text box. At the bottom of the form is a prominent blue button with the text "Register". Below the button, there is a horizontal line, followed by the text "Already have an account?" and a blue hyperlink labeled "Back to login".

Рис. Е.1 — Форма реєстрації кандидата в чатботі

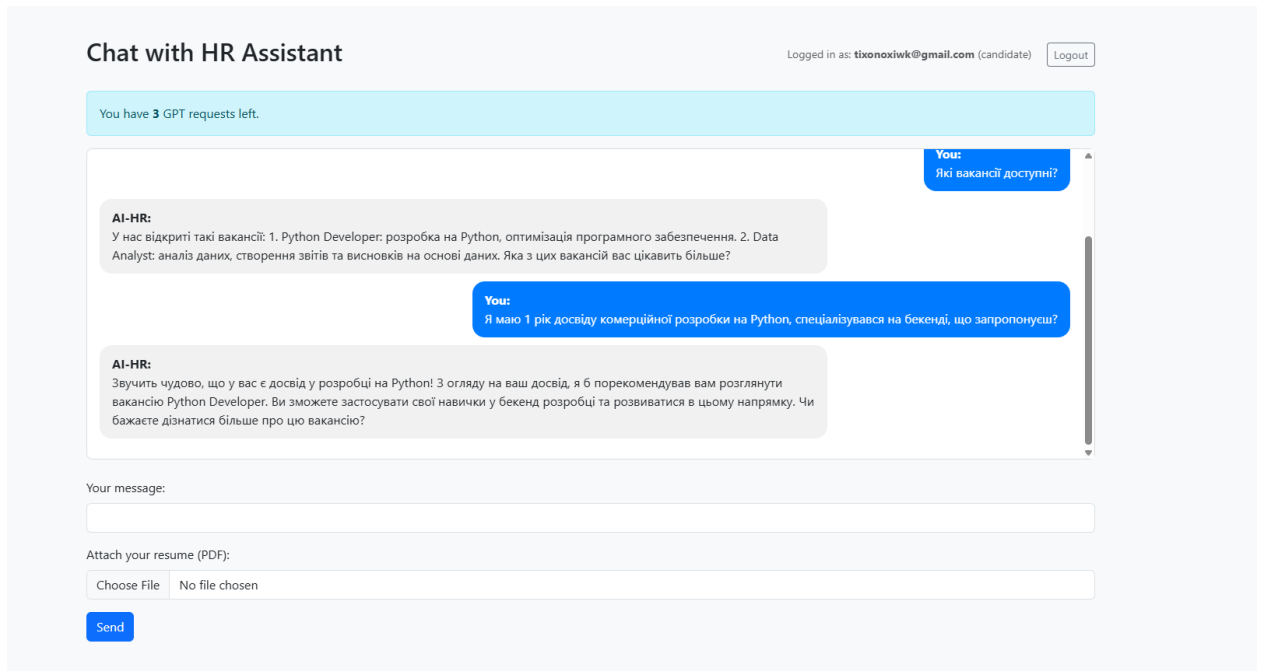


Рис. И.1 — Інтерфейс спілкування із AI-HR (ChatGPT 3.5)

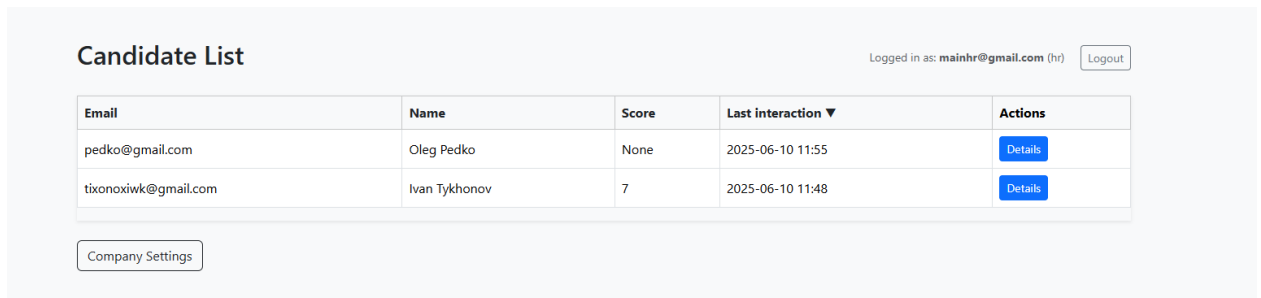


Рис. К.1 — Інтерфейс HR-панелі

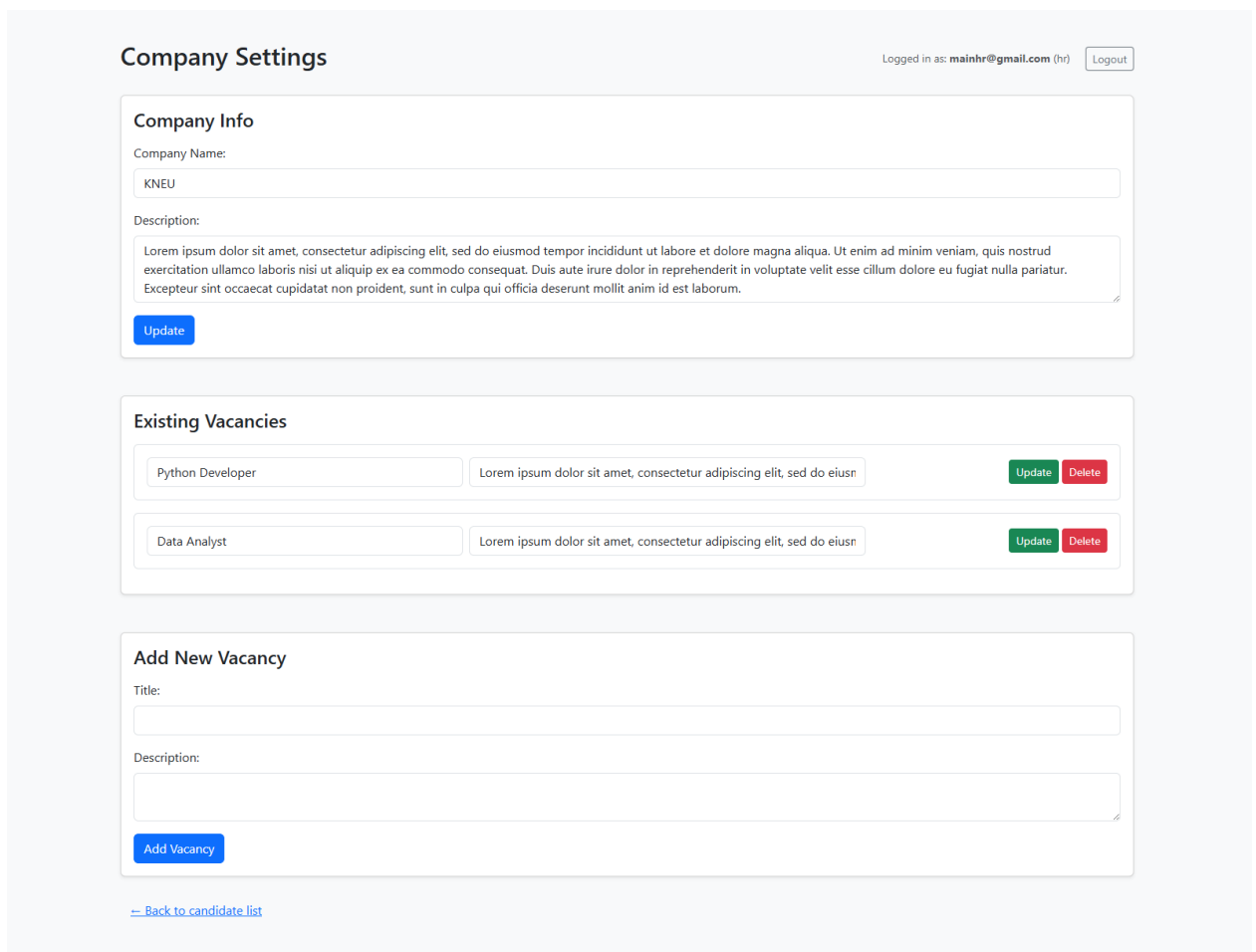


Рис. К.2 — Форма налаштування інформації о компанії / вакансіях

Candidate ID: 1 Logged in as: mainhr@gmail.com (hr) [Logout](#)

Candidate Info

Email:	tixonoxiwk@gmail.com
Name:	Ivan Tykhonov
Phone:	0508292566
Location:	Kyiv, Ukraine
LinkedIn:	—
Work type:	hybrid
Resume:	Download

Evaluation

Score: 7/10

Summary: Оцінка: 7/10. Підходяща вакансія: Data Analyst. Підсумок для HR: Кандидат Іван Тихонов з міста Київ, який виявив інтерес до вакансій у компанії KNEU, вказав бажаний формат роботи "hybrid". Враховуючи його профіль та запитання, він може бути цікавим кандидатом на позицію Data Analyst. Йому слід надати більш детальну інформацію про вимоги до цієї вакансії та запропонувати подальший етап співбесіди.

Messages

AI-HR: Вітаю, Ivan Tykhonov! В компанії KNEU ми займаємось сучасними технологіями, маємо дружню команду та відкриті вакансії. Будемо раді почути ваш досвід, навички та побажання щодо ідеальної посади. Розкажіть, будь ласка, про себе!

USER: Які вакансії доступні?

AI-HR: У нас відкриті такі вакансії: 1. Python Developer: розробка на Python, оптимізація програмного забезпечення. 2. Data Analyst: аналіз даних, створення звітів та висновків на основі даних. Яка з цих вакансій вас цікавить більше?

USER: Я маю 1 рік досвіду комерційної розробки на Python, спеціалізувався на бекенді, що запропонуєш?

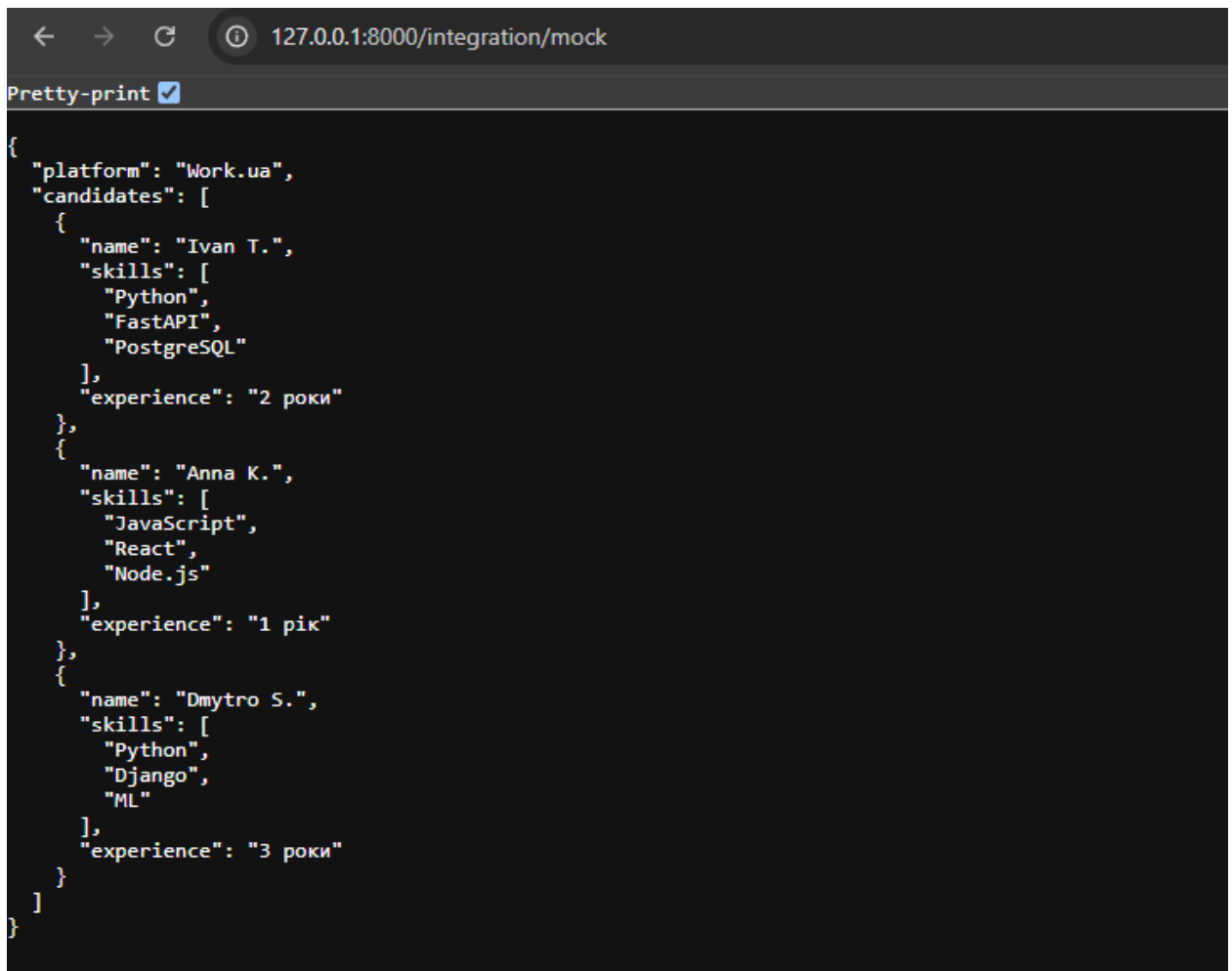
AI-HR: Звучить чудово, що у вас є досвід у розробці на Python! З огляду на ваш досвід, я б порекомендував вам розглянути вакансію Python Developer. Ви зможете застосувати свої навички у бекенд розробці та розвиватися в цьому напрямку. Чи бажаєте дізнатися більше про цю вакансію?

USER: 1

AI-HR: Чудово! Давайте поговоримо більше про вакансію Python Developer. Я розповім вам деталі про обов'язки, вимоги та можливості. Якщо у вас є питання або ви хочете поділитися своїм досвідом, будь ласка, не соромтеся запитати. Я готовий допомогти вам у цьому процесі.

[← Back to list](#)

Рис. Л.1 — Детальна інформація про кандидата



The image shows a web browser window with the address bar displaying "127.0.0.1:8000/integration/mock". Below the address bar, there is a "Pretty-print" checkbox which is checked. The main content of the browser is a JSON object displayed in a dark theme. The JSON object has a "platform" field with the value "Work.ua" and a "candidates" array containing three candidate objects. Each candidate object includes a "name", a "skills" array, and an "experience" field.

```
{
  "platform": "Work.ua",
  "candidates": [
    {
      "name": "Ivan T.",
      "skills": [
        "Python",
        "FastAPI",
        "PostgreSQL"
      ],
      "experience": "2 роки"
    },
    {
      "name": "Anna K.",
      "skills": [
        "JavaScript",
        "React",
        "Node.js"
      ],
      "experience": "1 рік"
    },
    {
      "name": "Dmytro S.",
      "skills": [
        "Python",
        "Django",
        "ML"
      ],
      "experience": "3 роки"
    }
  ]
}
```

Рис. М.1 — Інтеграція із Work.ua (вивід отриманого .json файлу)

```

1  from locust import HttpUser, TaskSet, task, between
2  import random
3
4  class ChatTaskSet(TaskSet):
5      @task
6      def send_message(self):
7          self.client.post("/chat-ui", json={
8              "message": random.choice([
9                  "Привіт, я шукаю роботу Python-розробника",
10                 "Який досвід потрібен для позиції Junior?",
11                 "Я маю 3 роки досвіду з Django. Це релевантно?",
12                 "Чи можу я працювати віддалено?",
13             ])
14         })
15
16 class ChatUser(HttpUser):
17     tasks = [ChatTaskSet]
18     wait_time = between(1, 3)
19

```

Рис. Н.1 — Код тесту навантаження

Locust Load Test Report

	Metric	Value
1	Total Requests	5000
2	Failed Requests	48
3	Median Response Time (ms)	1210
4	Average Response Time (ms)	1344
5	95% Percentile (ms)	2110
6	Requests per second	18.4
7	Failures per second	0.2
8	Peak concurrent users	100
9	Test duration	4 min 35 sec

Рис. Н.2 — Звіт тесту навантаження (оцінка продуктивності)

```
1 import pytest
2 from httpx import AsyncClient
3 from main import app
4
5 @pytest.mark.asyncio
6 async def test_gpt_response():
7     async with AsyncClient(app=app, base_url="http://test") as ac:
8         response = await ac.post("/chat-ui", json={
9             "message": "Привіт, я хочу знайти роботу Python-розробника"
10        })
11     assert response.status_code == 200
12     data = response.json()
13     assert "answer" in data # припустимо, GPT відповідь – у полі "answer"
14     assert isinstance(data["answer"], str)
15     assert len(data["answer"]) > 20 # перевіряємо, що не пуста фраза
16     assert "робота" in data["answer"].lower() or "посада" in data["answer"].lower()
17
```

Рис. П.1 — Код автотестів на перевірку коректності відповіді GPT

```
===== test session starts =====
platform win32 -- Python 3.11.2, pytest-7.4.0, pluggy-1.3.0
rootdir: [REDACTED]tests
plugins: asyncio-0.21.0

collected 4 items

test_chat_response.py .... [100%]

===== 4 passed in 2.03s =====
```

Рис. П.2 — Успішні результати автотестів

Репозиторій із проєктом:

https://github.com/Grekzk/HR_ChatBot_GPT

Звіт подібності



Дата звіту 6/11/2025
Дата редагування ---



Звіт не був оцінений

Звіт подібності

метадані

Назва організації

Kyiv National Economic University named after Vadym Hetman KNEU

Заголовок

Розроблення чат-бота для автоматизації рекрутингу з використанням API OPENAI та технологій штучного інтелекту

Автор

Науковий керівник / Експерт

Тихонов Іван Олександрович Маркіна О. М.

підрозділ

кафедра інформаційних систем в економіці

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.



КП 1

25

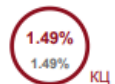
Довжина фрази для коефіцієнта подібності 2



КП 2

9237

Кількість слів



КЦ

71237

Кількість символів

Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв		1
Інтервали		0
Мікропробіли		0
Білі знаки		0
Парафрази (SmartMarks)		4

Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Колір тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

10 найдовших фраз

Колір тексту

ПОРЯДКОВИЙ НОМЕР	НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	https://ru.essays.club/%D0%A2%D0%BE%D1%87%D0%BD%D1%8B%D0%B5-%D0%BD%D0%B0%D1%83%D0%BA%D0%B8/%D0%9A%D0%BE%D0%BC%D0%BC%D1%83%D0%BD%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D0%B8-%D1%81%D0%B2%D1%8F%D0%B7%D1%8C%D0%97%D0%B0%D1%85%D0%B8%D1%81%D1%82-%D1%96%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D1%96%D1%97-%D0%B7-%D0%BE%D0%B1%D0%BC%D0%B5%D0%B6%D0%B5%D0%BD%D0%B8%D0%BC-470343.html	19 0.21 %

2	https://ru.essays.club/%D0%A2%D0%BE%D1%87%D0%BD%D1%8B%D0%B5-%D0%BD%D0%B0%D1%83%D0%BA%D0%B8%D0%9A%D0%BE%D0%BC%D0%BC%D1%83%D0%BD%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D0%B8-%D0%B8-%D1%81%D0%B2%D1%8F%D0%B7%D1%8C%D0%97%D0%B0%D1%85%D0%B8%D1%81%D1%82%D1%96%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D1%96%D1%97-%D0%B7-%D0%BE%D0%B1%D0%BC%D0%B5%D0%B6%D0%B5%D0%BD%D0%B8%D0%BC-470343.html	16 0.17 %
3	Розроблення підсистеми моніторингу статусу сервісів комп'ютерної мережі 6/9/2020 Kyiv National Economic University named after Vadym Hetman KNEU (кафедра інформаційних систем в економіці)	9 0.10 %
4	http://catalog.library.tnpu.edu.ua/naukovi_zapusku/geograph/2019/Geogr_19_1.pdf	8 0.09 %
5	http://catalog.library.tnpu.edu.ua/naukovi_zapusku/geograph/2019/Geogr_19_1.pdf	6 0.06 %
6	HIGHER EDUCATION DEVELOPMENT TRENDS IN THE CONTEXT OF EUROPEAN INTEGRATION PROCESSES Наталія Мачинська;	5 0.05 %
7	http://catalog.library.tnpu.edu.ua/naukovi_zapusku/geograph/2019/Geogr_19_1.pdf	5 0.05 %

з бази даних RefBooks (0.05 %)

ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
джерело: Paperity		
1	HIGHER EDUCATION DEVELOPMENT TRENDS IN THE CONTEXT OF EUROPEAN INTEGRATION PROCESSES Наталія Мачинська;	5 (1) 0.05 %

з домашньої бази даних (0.10 %)

ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	Розроблення підсистеми моніторингу статусу сервісів комп'ютерної мережі 6/9/2020 Kyiv National Economic University named after Vadym Hetman KNEU (кафедра інформаційних систем в економіці)	9 (1) 0.10 %

з програми обміну базами даних (0.00 %)

ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
---------------------	-----------	---

з Інтернету (0.58 %)

ПОРЯДКОВИЙ НОМЕР	ДЖЕРЕЛО URL	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	https://ru.essays.club/%D0%A2%D0%BE%D1%87%D0%BD%D1%8B%D0%B5-%D0%BD%D0%B0%D1%83%D0%BA%D0%B8%D0%9A%D0%BE%D0%BC%D0%BC%D1%83%D0%BD%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D0%B8-%D0%B8-%D1%81%D0%B2%D1%8F%D0%B7%D1%8C%D0%97%D0%B0%D1%85%D0%B8%D1%81%D1%82%D1%96%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D1%96%D1%97-%D0%B7-%D0%BE%D0%B1%D0%BC%D0%B5%D0%B6%D0%B5%D0%BD%D0%B8%D0%BC-470343.html	35 (2) 0.38 %
2	http://catalog.library.tnpu.edu.ua/naukovi_zapusku/geograph/2019/Geogr_19_1.pdf	19 (3) 0.21 %

Список принятых фрагментів (немає принятих фрагментів)