

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

Веб-додаток для генерації та аналізу резюме із застосуванням
штучного інтелекту»

08-54.БКР.014.00.000 ПЗ

Виконав: студент 4 курсу, групи 2 КІ-216
спеціальності 123 — «Комп'ютерна інженерія»
освітня програма — «Комп'ютерна інженерія»

Смаковський Б.А.

Керівник к.т.н., доц. каф. ОТ

Кожем'яко А.В.

Рецензент к.т.н., доц. каф. ПЗ

Майданок В.П.


Допущено до захисту
Завідувач кафедри ОТ
д.т.н., проф. Азаров О.Д.

«19» 06 2025 р.

Вінниця ВНТУ — 2025 рік

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки
Освітньо—кваліфікаційний рівень бакалавр
Галузь знань — 12 Інформаційні технології
Спеціальність — 123 Комп'ютерна інженерія
Освітня програма — Комп'ютерна інженерія

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ проф., д.т.н.

Азаров О. Д.

«21» березня 2025 року

З А В Д А Н Н Я

НА БАКАЛАВРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Смаковському Богдану Андрійовичу

1 Тема роботи: Веб-додаток для генерації та аналізу резюме із застосуванням штучного інтелекту, керівник роботи, Кожем'яко А.В., к.т.н., доцент кафедри ОТ, затверджено наказом вищого навчального закладу від «20» березня 2025 року №97.

2 Строк подання студентом роботи 13.05.2025

3 Вихідні дані до роботи: технічний опис комп'ютерної системи генерації резюме, технології розробки: мова програмування Python, бібліотека OpenAI для роботи із штучним інтелектом, бібліотека numpy для роботи із масивами даних, середовище розробки – Visual Studio.

4 Зміст розрахунково-пояснювальної записки: вступ; визначення параметрів вхідних даних, методи параметричної класифікації резюме, аналіз резюме за допомогою штучного інтелекту, порівняльний аналіз інструментів; висновки.

5 Перелік графічного матеріалу: діаграма класів.

Перелік ілюстративного матеріалу: чат-бот консультант.

6 Консультанти розділів роботи приведені в таблиці 1.

Таблиця 1 — Консультанти роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-3	к.т.н., доц. каф. ОТ Кожем'яко А.В.	21.03.25	25.05.25
Нормоконтроль	ас. каф. ОТ Швець С. І.	14.05.2025	30.05.2025

7 Дата видачі завдання 21.03.2025 р.

8 Календарний план приведений в таблиці 2.

Таблиця 2 — Календарний план

№ з/п	Назва етапів дипломної роботи	Строк виконання	Підпис
1	Постановка задачі розробки системи	21.03.25	СБ
2	Аналіз існуючих підходів визначення основних параметрів резюме	21.03.25—30.03.25	СБ
3	Аналіз методів генерації на основі штучного інтелекту обробки вхідного зображення	21.03.25—30.03.25	СБ
4	Визначення методів класифікації та аналізу резюме	31.03.25—13.04.25	СБ
5	Розробка комп'ютерної програми для моделювання резюме та її тестування	14.04.25—27.04.25	СБ
6	Оформлення пояснювальної записки та ілюстративного матеріалу	28.04.25—11.05.25	СБ
7	Перевірка якості виконання бакалаврської кваліфікаційної роботи та усунення недоліків	13.05.25	СБ

Студент

Богдан СМАКОВСЬКИЙ

Керівник роботи

к.т.н., доц. А.В. Кожем'яко

АНОТАЦІЯ

УДК 004.42

Б. Смаковський, веб-додаток для генерації та аналізу резюме із застосуванням штучного інтелекту. Бакалаврська дипломна робота зі спеціальності 123 «Комп'ютерна інженерія», освітня програма «Системне програмування». Вінниця: ВНТУ, 2025. 82 с.

На укр. мові. Бібліогр.: 16 назв, рис.: 20, табл.: 3.

У роботі проведено огляд існуючих веб-додатків з інтегрованим штучним інтелектом. Проведено огляд можливих застосувань штучного інтелекту у веб-додатках. Проведено аналіз існуючих технологій для створення веб-додатків. Проведено аналіз існуючих технологій для інтеграції штучного інтелекту.

Створено та проаналізовано діаграми класів, діаграми послідовностей та діаграми прецедентів.

Для реалізації веб-додатку було використано Flask, HTML, CSS, JavaScript, SQLite та AssistantAPI.

Ключові слова: Штучний інтелект; підбір функцій; прийняття на роботу; управління людськими ресурсами; підбір персоналу; резюме; прийняття рішень; сприятливість результату; чат-бот, веб-додаток; CSS, HTML. JAVASCRIPT, FLASK, ASSISTANTAPI

ABSTRACT

Smakovskii B. bachelor's thesis in the specialty 123 “Computer Engineering,” educational program “System Programming.” Vinnytsia: VNTU, 2025. 96 p.

In Ukrainian. Bibliography: 16, 20 figures, 5 appendices, 3 tables.

The paper reviews existing web applications with integrated artificial intelligence. It reviews possible applications of artificial intelligence in web applications. It analyzes existing technologies for creating web applications. It analyzes existing technologies for integrating artificial intelligence.

Class diagrams, sequence diagrams, and use case diagrams were created and analyzed.

Flask, HTML, CSS, JavaScript, SQLite, and AssistantAPI were used to implement the web application.

Keywords: Artificial intelligence; job selection; hiring; human resource management; staffing; resume; decision making; outcome favorability; chatbot, web application; CSS, HTML. JAVASCRIPT, FLASK, ASSISTANTAPI

ЗМІСТ

ВСТУП.....	4
1 АНАЛІЗ ТЕХНОЛОГІЙ УПРАВЛІННЯ ЛЮДСЬКИМИ РЕСУРСАМИ НА ОСНОВІ ВЕБ-ДОДАТКІВ ДЛЯ ГЕНЕРАЦІЇ ТА АНАЛІЗУ РЕЗЮМЕ	6
1.1 Аналіз літературних джерел управління людськими ресурсами на основі веб-додатків для генерації та аналізу резюме із застосування ШІ	6
1.2 Штучний інтелект в управлінні людськими ресурсами	11
1.3 Процес підбору персоналу за резюме із використанням штучного інтелекту	13
1.4 Можливості в застосуванні штучного інтелекту для управління персоналом	14
1.4.1 Чат-боти та НЛП у рекрутингу	15
1.4.2 Інтелектуальний аналіз даних для штучного інтелекту	15
2 ПРИНЦИПИ УПРАВЛІННЯ ЛЮДСЬКИМИ РЕСУРСАМИ НА ОСНОВІ ВЕБ-ДОДАТКІВ ДЛЯ ГЕНЕРАЦІЇ ТА АНАЛІЗУ РЕЗЮМЕ ІЗ ЗАСТОСУВАННЯ ШІ	17
2.1 Мінімальна довжина опису	18
2.2 Опорний вектор машини	18
2.3 Формування даних	19
2.4 Вибір функцій	22
2.5 Вилучення ознак.....	23
2.6 Модель машинного навчання та експерименти	24
3 ТЕХНОЛОГІЇ СТВОРЕННЯ КОНСТРУКТОРІВ ДЛЯ РЕЗЮМЕ ТА CV ЗІ ШТУЧНИМ ІНТЕЛЕКТОМ.....	28
3.1 Аналіз конструкторів резюме та CV зі штучним інтелектом	28
3.2 Основні параметри засобів штучного інтелекту для генерації резюме	35

4 РЕАЛІЗАЦІЯ ВЕБ-ДОДАТКУ ТА ШТУЧНОГО ІНТЕЛЕКТУ	37
4.1 Пояснення основного коду	37
4.2 Результати дослідження.....	47
ВИСНОВКИ	57
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	59
ДОДАТОК А Технічне завдання	63
ДОДАТОК Б Протокол перевірки кваліфікаційної роботи	67
ДОДАТОК В Графічна частина	68
ДОДАТОК Г Ілюстративна частина	70
ДОДАТОК Д Лістинг програмного забезпечення	72

ВСТУП

Штучний інтелект (ШІ) швидко впроваджується у функціональні модулі та створює значні проблеми та потенціал для управління людськими ресурсами з настанням цифрової ери.

Актуальність теми. Все більше організацій використовують штучний інтелект (ШІ) для прийняття рішень щодо набору працівників відділу кадрів на роботу. Проте суб'єктивність прийняття рішення на сьогодні в цьому питанні визначальною. Враховуючи, що на ефективність організації значною мірою впливає суб'єктивність прийняття рішення окремими особами, у цьому дослідженні використовується методологія відбору резюме як приклад для вивчення впливу штучного інтелекту, який допомагає керівникам організацій прийняти правильне рішення щодо зарахування працівника.

Це дослідження направлено на аналіз впливу штучного інтелекту, який допомагає у прийнятті рішень щодо ефективності. Запропонована модель може допомогти організаціям ефективніше використовувати штучний інтелект для перевірки резюме. А майбутні дослідження можуть вивчити співпрацю між людьми та штучним інтелектом для прийняття рішень щодо управління людськими ресурсами.

Впровадження штучного інтелекту в управлінні людськими ресурсами може допомогти бізнесу та створити значну перевагу на ринку. За допомогою штучного інтелекту більшість кадрових обов'язків можна виконати ефективно і в набагато коротші терміни. Для сталого розвитку компаній важливо скоротити процеси, які забирають багато часу і які можна автоматизувати. Особливо в процесі підбору персоналу штучний інтелект може полегшити аналіз невеликої кількості кандидатів та багато іншого. Це дослідження зосереджене на використанні штучного інтелекту для підбору та відбору персоналу як операції з управління людськими ресурсами.

Метою роботи є підвищення ефективності комунікацій між компаніями та клієнтами шляхом створення веб-додатків для генерації та аналізу резюме завдяки впровадженню чат-бота-консультанта на основі штучного інтелекту.

Об’єкт дослідження – процес підвищення ефективності комунікацій між компаніями та клієнтами завдяки впровадженню чат-бота-консультанта на основі штучного інтелекту.

Предмет дослідження – методи та технології створення веб-сайту та інтеграції чат-бота-помічника на основі штучного інтелекту.

Завдання дослідження:

- проаналізувати предметну область;
- оглянути поняття штучного інтелекту;
- провести огляд мови програмування та інших технологій для реалізації візуальної частини веб-додатку;
- оглянути наявні технології створення чат-бота-консультанта;
- спроектувати та реалізувати веб-додаток з використанням чат-бота-консультанта;
- розробити веб-додаток бібліотеку з інтегрованим чат-ботом консультантом.

Методи дослідження: вивчення існуючих аналогів; вивчення існуючих технологій; методи об’єктно-орієнтовного моделювання; побудова тестових програм; відлагодження тестових програм або додатку в цілому.

Запропоновані методи дозволяють ефективно використовувати наявні технології для розробки веб-додатків, що включають чат-боти та інші елементи штучного інтелекту.

Практична значущість отриманих результатів полягає в створенні працюючого веб-додатку, який має усі необхідні функції для взаємодії з користувачами за допомогою чат-бота-консультанта на основі штучного інтелекту. Цей додаток може бути використаний як у комерційних, так і в освітніх цілях, забезпечуючи ефективну комунікацію та підвищуючи задоволеність клієнтів. Розроблені підходи та рішення можуть бути застосовані для подальшого вдосконалення інших веб-додатків, що робить результати дослідження корисними для широкого кола користувачів та розробників.

1 АНАЛІЗ ТЕХНОЛОГІЙ УПРАВЛІННЯ ЛЮДСЬКИМИ РЕСУРСАМИ НА ОСНОВІ ВЕБ-ДОДАТКІВ ДЛЯ ГЕНЕРАЦІЇ ТА АНАЛІЗУ РЕЗЮМЕ

1.1 Аналіз літературних джерел управління людськими ресурсами на основі веб-додатків для генерації та аналізу резюме із застосування ІІІ

Управління людськими ресурсами (УЛР) має довгу історію використання. За матеріалами [1], метод УЛР був вперше винайдений у 1960-х роках і може бути застосований до будь-кого, хто має посаду в компанії. Відділ кадрів (ВК) також керує відділами в організації, якщо вони прямо чи опосередковано пов'язані з працівниками [1]. Управління персоналом має дві основні цілі в організації. Перша — це адміністративні посади, а друга — управління персоналом. Перший є відносно простим, оскільки більшість Положень технічно відомі. Тим не менш, навіть засновані на правилах функції ВК повинні бути переглянуті і виправлені, якщо вони викликають невдоволення у співробітників. Це змушує УЛР постійно контролювати співробітника. Відділ кадрів (ВК) повинен зосередитися на функціях, які є максимально важливими для створення стійкого бізнесу, а не мати справу з процесами, що забирають багато часу.

Нові технологічні розробки завжди впливали на життя людей. В результаті управління персоналом зазнало змін під впливом технологічного розвитку. В [2], основними цілями УЛР є залучення та утримання нових талантів, мотивація співробітників та відбір найкращих кандидатів. Технологічний розвиток змінив функціонування ВК-процесів, управління персоналом, посадові інструкції [2]. Звичайно, більшість удосконалень ВК-процесів стали можливими завдяки збільшенню обсягу даних та знань. Як зазначається в роботі інформаційні технології дозволяють більш децентралізовано приймати рішення, щоб більше осіб, які приймають рішення, могли зробити свій внесок. Це призводить до більш розширених можливостей і кращого перетворення входів (даних) у виходи [3]. Також зазначається, що використання даних та великих даних трансформують ВК-операції. Вони вважають, що головними викликами є отримання, обробка та аналіз даних в інтересах компаній. Однак все ще залишаються такі питання, як розмір, як організувати, як отримати доступ і як розподілити [4]. Складність, розмір і

швидкість потоку даних дозволяють алгоритмам машинного навчання аналізувати і прогнозувати результати на основі даних. Сучасні УЛР-системи можуть збирати, запитувати та аналізувати статистичні дані, щоб мати можливість аналізувати взаємозв'язок між даними, потрібно використовувати алгоритми машинного навчання (МН) [5].

Технологічний прогрес не тільки автоматизує та економить час, але й представляє процес, який є дуже ефективним та справедливим. Будь-яка технологія може розвиватися лише тоді, коли вона знаходиться в потрібний час. Зростання «великих даних» і прогрес в обчислювальній потужності відкрили нові можливості для ІІІ [6]. Все ще існують деякі перешкоди та виклики щодо розвитку ІІІ, такі як дефіцит навчальних даних, регулювання та відсутність стандартів. Сьогодні існують можливості адаптувати штучний інтелект (ІІІ) та технології на основі штучного інтелекту, такі як машинне навчання та обробка природної мови, до різних етапів управління персоналом.

У той же час для є важливим обговорення етичних питань та питань прозорості [7]. Коли мова йде автоматизацію, то це може бути пов'язано із скороченням персоналу, що викликає страх безробіття. Кожен технологічний стрибок змінює людську працю та необхідні навички, які потрібні через перероблені посадові інструкції. Нові технології можуть спричинити технологічне безробіття, яке вважається короткостроковим безробіттям і може бути подолане шляхом належної освіти для здобуття необхідних спеціалізованих компетентностей. Крім того, у статті зазначається, про соціальний аспект підкатегорій штучного інтелекту не повністю розуміє кожна людина. Особливо в залежності від вікових категорій змінюється усвідомлення взаємодії зі штучним інтелектом. Разом з тим, вони зазначили, що оскільки галузь потребує фахівців з нових технологій, вкрай важливо здобути потрібну освіту [8]. Крім того, буде більше викликів і більше сірих зон, оскільки ІІІ формується протягом життя людини у майбутньому.

Модернізація штучного інтелекту на основі машинного навчання може революціонізувати кадрові відділи на різних рівнях. Відділи підбору персоналу,

управління, навчання та пільг – це приклади, які можуть використовувати штучний інтелект як революцію в ВК. Вони стверджували, що штучний інтелект може забезпечити величезну вигоду для залучення кандидатів і передбачити додану вартість кандидата для компанії [9]. Сила та потужність зростання організацій походить від того, як вони розумно об'єднують робочу силу та обладнання з мінімальними витратами. Для того, щоб залишатися конкурентоспроможними на нинішньому ринку; компаніям слід зрозуміти можливості штучного інтелекту та переосмислити кадрові операції, щоб отримати вигоду від штучного інтелекту. Ці роботи спрямовані на реалізацію мети шляхом створення алгоритму, який постійно вчиться оцінювати претендентів та створювати шорт-лист (CV) кандидатів для економії часу та коштів.

Відбір кандидатів може зайняти більше кількох тижнів, але штучний інтелект не просто вирішує це, але й може усунути несвідомі людські упередження та дозволяє ВК зосередитися на залученні найкращих кандидатів, щоб зберегти конкурентну перевагу для компанії [10]. Проте, оскільки алгоритми AI та МН є рішеннями «чорного ящика», можливо, що ШІ також може прийняти неетичне або дискримінаційне рішення. Крім того, оскільки штучний інтелект може бути успішним лише в тій мірі, в якій він подається з даних, наданих для навчання, при дефіциті даних про навчання не виключено, що ШІ може зробити упереджений вибір, який менеджери з персоналу робили раніше. Штучний інтелект настільки сильний, наскільки сильні наведені дані, тому він може створити той же результат, що і менеджер з персоналу. Ця робота спрямована на вирішення як проблеми витрати часу на створення CV, так і на усунення людських упереджень шляхом їх усунення до відбору даних за вказівками операторів (ВК-менеджерів). Тому в першу чергу створюється набір даних для алгоритмів навчання. Пізніше для алгоритмів будуть готові елементи резюме, які є особливостями алгоритмів навчання. Вибір ознак і виділення ознак висвітлить, які елементи використовувалися до цього часу і повинні використовуватися для алгоритмів навчання. На заключному етапі буде розглянута оцінка продуктивності алгоритму навчання. Цей процес доведе, що алгоритми штучного інтелекту можуть імітувати

ВК і мають здатність робити більше, постійно навчаючись. Існує небагато академічних досліджень, які зосереджені на особистісних тестах. Аналіз літературних джерел про процес підбору персоналу та штучний інтелект у процесі рекрутингу з подальшим відбором даних у вигляді алгоритмів машинного навчання та алгоритмів векторного машинного навчання (ВМН) з результатами на основі даних.

ВМН є основним фактором конкурентних переваг компаній. Вони також заявили, що стійкість ВК може бути досягнута завдяки реалізації довгострокового підходу до управління людьми, що призводить до стійкої ефективності бізнесу та виплат працівникам [11]. Першим кроком до найму довгострокового відданого співробітника є підбір правильного працівника на посаду, що підтримує його мотивацію. Отже, утримання співробітників починається з вибору того, хто підходить на посаду [11]. Тому важливо набрати правильного співробітника як ресурс для компанії. Можна втратити кандидата, поставивши його в довгий і неоднозначний процес відбору. Крім того, якщо процес найму займає занадто багато часу, а відділ терміново потребує робочої сили, їм може знадобитися найняти некомпетентного співробітника, а потім повторити весь процес. Саме тому етап найму потрібно пройти максимально бездоганно і якнайшвидше. Оскільки складання CV листа є найбільш трудомісткою частиною цього процесу та найбільш підходящим процесом для автоматизації, цей проект зосереджений на цій темі.

Досліджень на одну і ту ж тему небагато, але в цілому вони зосереджені на особистісних тестах і на тому, як можна використовувати ці тести для відбору кандидатів. На підставі інформації, отриманої з співбесід з ВК-менеджерами, відповідальними за підбір персоналу, був зроблений висновок, що особистісний тест не є важливою ознакою процесу відбору. Рекрутер заявив, що хоча особистісні тести є обов'язковими перед прийомом на роботу, вони не використовуються для відбору чи визначення пріоритетів. Однак, наскільки відомо, не існує документу про проект зі штучного інтелекту, який би ґрунтувався на особливостях резюме претендентів і ділився результатами продуктивності застосованих алгоритмів і моделей.

Підбір персоналу є одним з найважливіших завдань ВК, адже тільки набравши найбільш підходящих співробітників для потрібної роботи, компанія може продовжувати свою бізнес-діяльність з високою ефективністю. У цифрову епоху процес підбору персоналу кардинально змінився інноваційним чином, оскільки було знайдено найбільш підходящого кандидата на посади, які набагато важливіші для конкурентних переваг компанії. COVID-19 також підштовхнув УЛР до подальшої цифровізації від звичайного найму [12]. Резюме (CV) включає освіту, навички, сильні сторони, досягнення та інші атрибути, пов'язані з роботою. Попередній відбір резюме є важливим кроком у відборі персоналу, оскільки він має можливість продемонструвати навички, мотивацію, можливості та потенційну цінність претендентів. З його допомогою можна зробити висновки про кандидата. Перевірка резюме є найважчою та найтрудомісткішою частиною процесу найму. Чим більшим стає пул резюме, тим складніше та довше його перевірити. Крім того, завжди можна пропустити найкращу відповідність вакансіям, коли резюме для оцінки занадто велике [13]. Крім того, коли є велике резюме, які потрібно розглянути, завжди можна пропустити те, яке найкраще підходить для вакансій. Після розгляду резюме претендентів, якщо ВК схвалює, вони додаються до шорт-листа, і, як правило, наступний етап починається після завершення шорт-листа. Таким чином, заявникам, як правило, доводиться чекати, поки шорт-лист буде повним або, принаймні, достатнім.

Основна мета УЛР полягає в тому, щоб зібрати компетентних людей у компанії, навчити персонал, змусити їх працювати з максимальною віддачею та забезпечити, щоб співробітники зберігали свою продуктивну приналежність на благо організації. Однак основна диференційована цінність УЛР виходить з здатності визнавати особисті якості окремих осіб і передавати знання всередині організації [14]. Щоб реалізувати це завдання, ВК імпортував безліч стратегій і технологій з різних сфер. Психометричні тести, аналітична обробка в Інтернеті та штучний інтелект є прикладами дисциплін, що застосовуються до управління персоналом. УЛР рухається до цифрової епохи та цифрових технологій. Це вимагає від співробітника розуміння машин і роботи з ними для виконання своїх завдань.

Також важливим є навчання працівників та вимога кваліфікованого персоналу до роботи в цифровому середовищі. Таким чином, трансформація УЛР для роботи з машинами вимагає змін у робочій силі, а також внутрішнього навчання та набуття нових навичок цифрової епохи для працівників. Вони повинні адекватно аналізувати або розуміти штучний інтелект в достатній мірі, щоб використовувати зібрані дані на благо організації [15]. Завдяки технологічним розробкам ВК тепер відповідає не лише за посередництво між організацією та окремими людьми, але й за усунення бар'єру відстані та створення віддаленого робочого середовища. Це вимагає глобального процесу рекрутингу для отримання глобальних талантів. Збільшення кількості та різноманітності претендентів вимагає як онлайн, так і автоматизованих ВК-процесів. Тому потреба в управлінні даними є критично важливою операцією для організації. Це одна з причин впроваджувати системи штучного інтелекту для управління персоналом. Штучний інтелект може замінити більшість рутинних ВК-завдань з мінімальним втручанням людини. Впровадження штучного інтелекту також має багато переваг, які дозволяють завершити процес за менший час з підвищеною точністю [16].

1.2 Штучний інтелект в управлінні людськими ресурсами

Словосполучення «штучний інтелект» спочатку було використано в пропозиції літнього семінару Дартмутського коледжу. На семінарі «штучний інтелект» був описаний як «здатність змусити машини поводитися так само, як люди поведуться розумно». Простіше кажучи, штучний інтелект (ШІ) — це широке загальне словосполучення, яке стосується застосування обчислювальних методів для моделювання людського інтелекту. В останні роки все більше організацій впроваджують алгоритми на робочому місці, а деякі навіть почали впроваджувати штучний інтелект (ШІ) для підтримки менеджерів з персоналу в таких завданнях, як управління ефективністю співробітників, просування по службі, співбесіди та перевірка резюме [17]. Одним із ключових факторів, що впливають на зростаюче використання ШІ в прийнятті управлінських та організаційних рішень, є його здатність швидко та ефективно сортувати величезні обсяги інформації [8]. Крім

того, пов'язаний мета-аналіз виявив, що штучний інтелект працює в середньому на 10% точніше, ніж людське судження [18]. Всі ці результати вказують на використання штучного інтелекту в прийнятті рішень УЛР підприємствами як значну тенденцію на майбутнє і показують, що штучний інтелект ефективніше приймає рішення, ніж люди.

В даний час компанії рухаються до Digital Recruitment 3.0 – етапу, який зосереджений на використанні технології штучного інтелекту в процесах підбору та підбору персоналу завдяки прогресу цифрових технологій в управлінні людськими ресурсами [19]. Artificial Intelligence Recruitment – це термін для методів та інструментів, які використовуються підприємствами для відбору талантів шляхом обробки даних та прийняття оцінок і рішень щодо найму за допомогою таких технологій, як машинне навчання, обробка природної мови та розпізнавання емоцій [19]. Згідно з визначенням, інтелектуальний скринінг резюме, онлайн-оцінювання та відеоспівбесіди є основними сценаріями найму для штучного інтелекту (ШІ). Інші важливі технології в рекрутингу зі штучним інтелектом включають обробку природної мови, глибоке навчання, виявлення мови та емоцій тощо. Крім того, деякі вчені зазначають, що існує чотири основні компоненти рекрутингу зі штучним інтелектом: інформаційно-роз'яснювальна робота, скринінг, оцінка та координація [19]. Повний процес, від визначення відповідних претендентів до прийняття на роботу, висвітлюється в цих чотирьох розділах. У цих чотирьох компонентах рекрутинг штучного інтелекту функціонує інакше, ніж звичайні методи вербування людей. Наприклад, штучний інтелект має потенціал змінити спосіб, яким зазвичай оприлюднюється інформація про роботу. Його можна використовувати для націлювання на заявників шляхом збору даних користувачів із платформ соціальних мереж, таких як Facebook і LinkedIn, для обробки природної мови. Штучний інтелект також можна використовувати у віртуальній реальності та іграх для проведення співбесід та оцінювання кандидатів. Тим часом рекрутинг на основі штучного інтелекту може допомогти підприємствам отримати доступ до найкращих талантів як конкурентну перевагу та заощадити значну кількість часу порівняно з традиційним підбором персоналу людьми. Однак попередні

дослідження в основному вивчали переваги штучного інтелекту з точки зору організацій, а не з точки зору людей, на яких впливають рішення, прийняті штучним інтелектом. Це має суттєве значення для визначення того, чи може штучний інтелект бути впроваджений на підприємствах у широкому та тривалому масштабі. На прикладі скринінгу резюме можна сказати, що штучний інтелект (ШІ) може заощадити понад 80% часу, витраченого на традиційні ручні техніки. Однак попередні дослідження не стосувалися того, чи готові претенденти на перегляд їхніх резюме штучним інтелектом, або як змінилися психологічні оцінки заявників щодо них. Тим не менш, це дослідження спрямоване на вирішення цих питань.

1.3 Процес підбору персоналу за резюме із використанням штучного інтелекту

Підбір персоналу є одним із найбільш часто використовуваних процесів, вбудованих у штучний інтелект. Штучний інтелект допомагає у виконанні різних завдань процесу та покращує досвід, заощаджуючи час. Він сканує великі обсяги резюме та автоматично оцінює їх, щоб у найкоротші терміни створити шорт-лист. Це скорочення часу не тільки сприяє скороченню витрат, але й покращує імідж компаній і допомагає захистити конкурентні переваги над ринком. Процес підбору персоналу, орієнтований на кандидатів, є ключем до того, щоб стати гідним кандидатів і отримати конкурентну перевагу над конкурентами. Для досягнення цієї мети процес підбору персоналу має бути простішим, і, що важливіше, слід підтримувати чітку комунікацію між кандидатами та процесом найму. Якщо відхилений претендент все ще зацікавлений у компанії на майбутнє та рекомендує компанію до своєї мережі, процес підбору персоналу пройшов успішно [25]. Традиційне управління наймом відстає від війн талантів у нових сферах. Перемога в цій війні особливо важлива в конкурентних галузях. Ось чому пропонуються стратегічні довгострокові рішення щодо талантів для відділів підбору персоналу. Рішення зі штучним інтелектом можуть вирішити багато проблем за допомогою таких методів, як чат-боти, обробка природної мови (NLP) та алгоритми глибокого навчання.

Було кілька спроб адаптації ШІ для ВК. Наприклад, лише великі компанії впроваджують штучний інтелект у рекрутинг, і наскільки вони його використовують, досі незрозуміло. Отже, незрозуміло, які прийоми або методи використовуються в ШІ для підбору персоналу в даний час [20]. Алгоритм машинного навчання з використанням SVM для профілювання особистості кандидатів на платформах соціальних медіа, отримує 64,5% точності класифікації для п'яти класів (типів особистості) [21]. Оскільки ROC, AUC та матриця плутанини не були надані, точність 64,5% не є достатньою для висновку про те, чи була модель успішною чи ні. Що ще важливіше, використання типів особистості не є достатнім критерієм відбору для підбору персоналу. Дані (про особливості управління персоналом) були зібрані методом анкетного опитування. Для створення нового алгоритму вибору ознак було використано поширення спорідненості (AP) та SVM. Вони зменшили 24 функції з новою моделлю до 12 функцій і підвищили точність від 84,98% до 85,84% [22]. З іншого боку, дослідження для підбору персоналу за допомогою алгоритму навчання SVM. Дані були зібрані за допомогою особистісних опитувань нинішніх співробітників і кандидатів. Незважаючи на те, що традиційне мислення припускає, що оцінка навичок має більше значення, ніж особистість, їхні дослідження показують, що особистість є ключовим фактором в оцінці працівника на посаду [30].

1.4 Можливості в застосуванні штучного інтелекту для управління персоналом

Чат-боти — це програми штучного інтелекту, які широко використовуються в проєктах з управління персоналом. Як показує дослідження Soutar, сектори електронної комерції, охорони здоров'я та страхування зазнали змін завдяки впровадженню чат-ботів, але сектор підбору персоналу має менше 30 відсотків використання. Чат-боти на основі правил не вимагають штучного інтелекту або NLP, а слідують заздалегідь запрограмованому шаблону

1.4.1 Чат-боти та НЛП у рекрутингу

Чат-боти на основі даних можуть створювати довгі та складні розмови. Таким чином, завдяки NLP можна зібрати набагато більше результатів з чат-ботів на основі даних [31]. Двозначні розбори слів та неграматичні проблеми розмовної прози, які вимагають глибшого аналізу та методів машинного навчання для вирішення значення фактури. Алгоритми намагаються вирішити найбільш ймовірний результат (значення) заданих входів, оскільки неможливо виявити значення за допомогою заздалегідь запрограмованих кодів [23,24]. Компанії вважають, що чат-боти можуть мати прямі переваги перед комутацією електронної пошти, яка зараз є бажаною комунікацією з претендентами [25]. Чат-боти – це просунута область NLP-додатків, які полегшують автоматизацію на основі комунікації без необхідності використання оператора. Чат-боти створюють оперативну комунікацію між кандидатами та ВК-відділом [25]. Чат-боти можуть заздалегідь відповідати на запитання про кандидатів, такі як діапазон заробітної плати, умови відпустки, поширені запитання та заохочення щодо робочого простору. Завдяки чат-ботам роботодавці отримують змогу заощадити 74% маркетингових зусиль з підбору персоналу, які включають часові та бюджетні витрати. Крім того, чат-боти позитивно впливають на залученість кандидатів і дозволяють легше справлятися з активностями на борту [26]. Чат-бот може легко отримувати такі вхідні дані, як адреса, рівень освіти та досвід, за допомогою низькорівневих додатків чат-ботів. Однак висококласний чат-бот (для якого потрібен штучний інтелект) може впоратися практично з будь-якою співбесідою з кандидатом.

1.4.2 Інтелектуальний аналіз даних для штучного інтелекту

Інтелектуальний аналіз даних – це область інформаційних технологій, яка поєднує в собі статистику, машинне навчання і штучний інтелект. Інтелектуальний аналіз даних – це процес виявлення значущої та корисної інформації з раніше невідомих або прихованих закономірностей у наборі даних [27]. У штучному інтелекті методи інтелектуального аналізу даних можуть бути використані для отримання результатів чотирьох основних функцій: асоціації, кластеризації,

класифікації та прогнозування. Вони стверджували, що, використовуючи інтелектуальний аналіз даних, можна витягти цінність з біографії абітурієнтів, яка може бути використана в системах штучного інтелекту [28]. Інтелектуальний аналіз даних можливий за допомогою різних методів на заданому наборі даних, таких як дерево рішень, k-найближчий сусід, машини підтримки векторів (SVM), нейронні мережі та моделі підтримки прийняття рішень. Враховуючи, що проблеми класифікації заздалегідь визначають мітки (класи) для даного набору даних, потік алгоритму майнінгу складається з п'яти основних функцій, включаючи збір даних, підготовку даних, інтелектуальний аналіз даних та оцінку інтелектуального аналізу даних (визначення найкращої моделі, яка призводить до найвищої точності). Тоді можна отримати правила класифікації на основі створеної моделі, якщо вона виявилася досить успішною і пройшла оцінку ефективності [29]. За допомогою виявлення аномалій інтелектуального аналізу даних можна досягти вивчення асоціативних правил, кластеризації, класифікації, регресії та узагальнення [30]. Однак для процесу підбору персоналу класифікація є проблемою, яку має вирішити штучний інтелект. По міру зростання організацій при виборі бажаного персоналу в якості співробітників слід враховувати все більшу кількість критеріїв (особливостей машинного навчання). Тому він вимагає більш тривалого часу, а алгоритми інтелектуального аналізу даних дозволяють заощадити час і скоротити витрати [31].

2 ПРИНЦИПИ УПРАВЛІННЯ ЛЮДСЬКИМИ РЕСУРСАМИ НА ОСНОВІ ВЕБ-ДОДАТКІВ ДЛЯ ГЕНЕРАЦІЇ ТА АНАЛІЗУ РЕЗЮМЕ ІЗ ЗАСТОСУВАННЯ ІІІ

В роботі приділено увагу дослідженням створення шорт-листа (CV) з резюме кандидатів. Оскільки НЛП не є предметом даного дослідження, дані CV фіксувалися в системі реляційних баз даних (RDBMS) як в середніх і великих компаніях. Іншими словами, резюме не зчитуються з паперів, а зберігаються в цифровому вигляді в таблицях бази даних, що полегшує підготовку даних. Як згадувалося раніше, існує як мінімум чотири етапи машинного навчання та штучного інтелекту.

Перший етап передбачає підготовку даних, створення моделі та аналіз продуктивності моделі [32], спочатку необхідно отримати дані. Таким чином, доступ до даних та запит до них є першим етапом підготовки даних для всіх проєктів з машинного навчання чи штучного інтелекту. Потім до набору даних може бути застосована деідентифікація з міркувань конфіденційності. Маркування даних, форматування даних і зберігання нових даних також повинні бути реалізовані до використання методів машинного навчання [33]. Після того, як всі препарати в залежності від даних моделі можуть бути очищені, можна заповнити відсутні дані. Вибір або вилучення ознак є одним з найважливіших етапів у підготовці даних.

Вибір ознак набору даних безпосередньо впливає на продуктивність моделі, оскільки деякі функції (входи) не мають зв'язку з бажаним виходом. Вибір ознак може бути складним завданням через велику кількість нерелевантних або зайвих функцій або зашумлених даних, якщо немає можливості дізнатися про функції заздалегідь [34]. Творення нової підмножини з вибором ознак може бути застосовано за допомогою генетичного алгоритму, послідовного плаваючого вперед відбору, послідовного вибору ознак, класифікатора SVM або алгоритмів навчання без учителя.

2.1 Мінімальна довжина опису

Принцип мінімальної довжини опису (MDL), який є відносно недавнім методом, як принцип вибору моделі для індуктивного висновку, який забезпечує рішення проблем вибору, уникаючи проблем перенавчання. Метою MDL є стиснення значущих даних, що називається «регулярністю». Потім він може якнайкраще стиснути значущі функції, поєднавши існуючі функції. Він може використовуватися як простий МДЛ або може бути перетворений в більш високі (більш складні) поліноміальні ступені для опису інформації про дані [49]. Більшість алгоритмів MDL пов'язані з вибором байєсівської моделі. Крім того, MDL мінімізує суму наступного: довжина в бітах опису теорії; довжина в бітах даних при кодуванні за допомогою теорії. Якщо класифікатор позначає цільовий клас, роблячи прогноз (описаний як категоричний), то MDL збільшує позитивні та негативні помилки. Тому алгоритм працював погано і вимагав упередженої стратегії [35].

2.2 Опорний вектор машини

SVM є одним з найбільш значущих алгоритмів навчання для задач класифікації. Класифікувати заданий набір даних можна за допомогою лінійних або нелінійних поверхонь поділу. SVM — це алгоритм контрольованого навчання, тому необхідно позначити цільовий клас, перш ніж обробляти його за допомогою SVM. Якщо лінійна класифікація або регресійний аналіз не увінчалися успіхом, можна перемістити вхідні дані на більш високі розмірності за допомогою трюків ядра. SVM намагається створити максимальний запас, який є максимальною відстанню між площиною і найближчими даними [36]. SVM працює шляхом перетворення вибіркового даних у вищий розмір, ніж вихідні дані. Згодом він робить спробу знайти максимальну межу, що розділяє класи даних. Якщо немає можливості розділити класи, навіть з набагато більшими розмірами, для формалізації SVM слід використовувати змінні slack.

Змінна слабкості являє собою похибку в точці [37]. Графічне представлення SVM показано на рисунку 2.1. Оскільки неможливо ідеально розрізнити два різні

класи (кольори кіл) лише за допомогою лінійної гіперплощини, існує значення похибки слабкості.

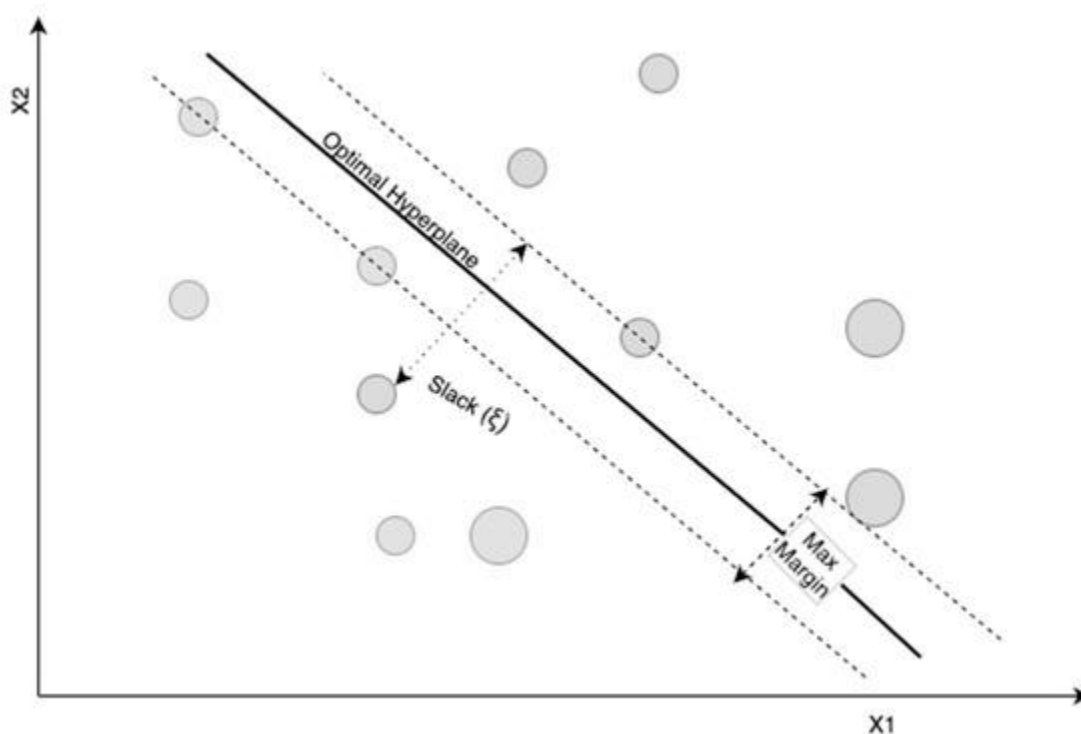


Рисунок 2.1 — Алгоритм SVM

2.3 Формування даних

Першим кроком штучного інтелекту є збір даних і доступ до них. Тому що для алгоритму навчання дані мають бути реальними або реалістичними, щоб ШІ міг навчатися так, як він повинен. У відділі кадрів неможливо зібрати всі аспекти, щоб знайти хорошого співробітника, і не всі дані, які збираються, мають відношення до залучення співробітників. Крім того, не існує стандартизації найму, і кожна компанія має свою організаційну культуру. Таким чином, можна отримати різні характеристики для кожного набору даних, які є важливими або релевантними. Крім того, деякі особливості можна розцінювати і як культуру компанії.

У цьому проекті вік і стать співробітників були відкинуті перед вибором функції через зворотний зв'язок з ВК для процесу найму. Навіть якщо на людську упередженість впливають минулі найми, ці функції більше не вітаються в процесі найму. Реальний опис вакансії був розміщений на сайті для пошуку роботи для

отримання резюме претендентів. Посада в галузі інформаційних технологій (ІТ) була створена для використання аналізу даних з можливостями віддаленої роботи. Виражається в тому, що англійська мова є обов'язковим, і необхідно мати не менше двох років досвіду роботи. Крім того, від кандидатів вимагається мати сильні письмові та усні комунікативні навички, а також бути обережними та дисциплінованими. Загалом із сайту було зібрано 140 резюме.

У цьому проекті використовувалася інформація резюме реальних даних, але для конфіденційності та правових норм вся інформація була анонімізована (анонімізація даних). Закон про захист персональних даних чітко говорить, що необхідно подати заявку на деідентифікацію або що дозвіл має бути підписаний заявниками. Після отримання та зберігання даних необхідно виконати перетворення перед створенням дійсної підмножини. Етапи роботи проілюстровані на рис. 2.2.

Мета полягала в тому, щоб використовувати SVM перед створенням моделі, яка необхідна для перетворення категорійних ознак в числові представлення. Таким чином, деякі ознаки, такі як статус випускника, військовий статус, місто, район, група крові та клас водійського посвідчення, були перетворені в числові представники. Боротися з відсутніми та надлишковими значеннями можна за допомогою алгоритмів вибору ознак; Однак у цьому проекті вони розглядалися як етап підготовки даних перед виконанням будь-якого алгоритму.

У навчанні та тестуванні було використано лише 100 точок даних (претендентів) як для алгоритмів відбору даних, так і для алгоритмів навчання. Всі дані резюме зберігалися в базі даних (особливості алгоритму навчання), але номери телефонів, імена, прізвища, точні адреси та інша інформація, яка може бути використана для ідентифікації людини, були опущені або анонімізовані перед збереженням в локальній базі даних. Оскільки всі дані зберігалися в реляційній базі даних, а всі моделі працювали тільки на одній таблиці, перед моделюванням повинна відбутися денормалізація. У випадку створення резервування, воно повинно розглядатися як етап підготовки даних.

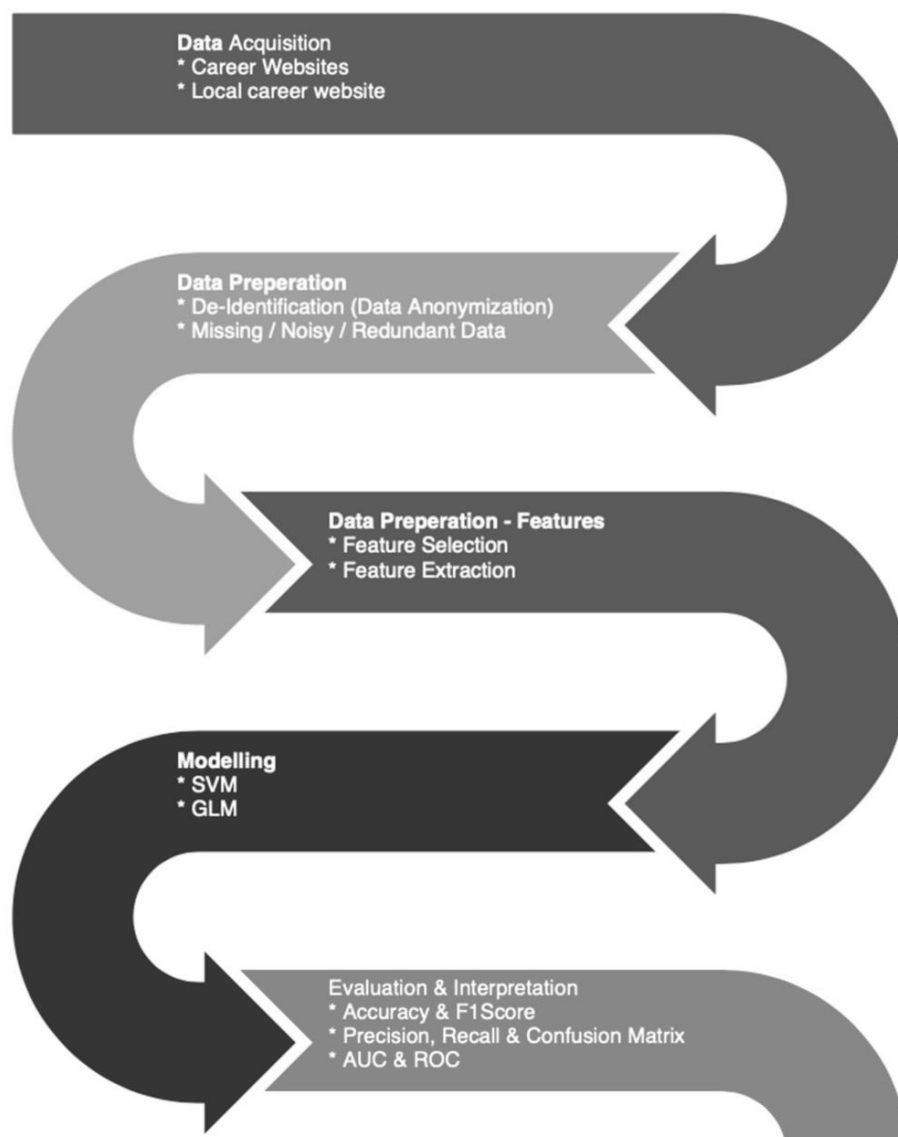


Рисунок 2.2 — Усі етапи навчання штучному інтелекту

Крім того, потрібна додаткова підготовка даних, оскільки деяка інформація зберігається в рядках, а не в стовпцях, оскільки нормалізація в СУБД цього вимагає. Наприклад, минулий досвід роботи кандидата (може бути кілька в резюме) може бути об'єднаний з кандидатом як NxN-зв'язок. Однак кожен кандидат повинен мати лише один рядок у наборі даних для алгоритму машинного навчання. Це множить числа ознак і створює зашумлені дані, як показано на рисунку 2.3, тобто перетворення значень стовпців (рядків) бази даних в стовпці (особливості алгоритму навчання), що створило 18 додаткових стовпців для використання в якості функцій.

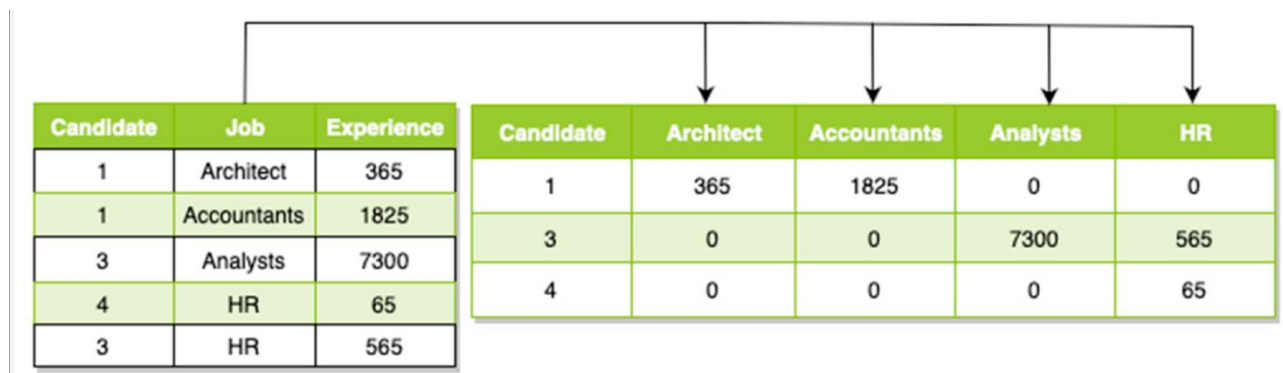


Рисунок 2.3 — Перетворення значень стовпців (рядків) бази даних в стовпці (особливості алгоритму навчання)

Саме тому кожен минулий досвід роботи (аналітик, розробник програмного забезпечення, менеджер тощо) має бути окремою особливістю, а не цінністю функцій. Для цього прикладу необхідно помножити роки на 12 і додати місяці як вартість посадових місць.

2.4 Вибір функцій

У роботі функції, які можуть ідентифікувати кандидата, були видалені або приховані перед збереженням даних. Крім того, деякі характеристики, такі як стать або вік, були усунені оператором, оскільки це було прийнято як неетичне використання в процесі найму в новому підході до управління персоналом. Це не було враховано в попередніх дослідженнях, які стежили за даними в тому вигляді, в якому вони були. Однак концепції УЛР змінюються, і не завжди достатньо для того, щоб штучний інтелект відтворив спеціалістів з підбору персоналу в минулому компанії. Останнім етапом підготовки даних є відбір даних. У цьому дослідженні була використана модель на основі MDL. Всього було створено 107 атрибутів, але 85 залишилися після очищення даних. Очищення передбачає видалення всіх NULL (порожніх) атрибутів, атрибутів, які не можна класифікувати, і нечислових атрибутів. Наприклад, оскільки атрибут академічного звання порожній у всіх кандидатів, він був очищений як підготовка даних, тому він не буде використовуватися в алгоритмі навчання. Крім того, з фічі були видалені "пояснення", так як немає можливості категоризувати, а НЛП в проекті не введено.

Функція "Важливість атрибутів" з бази даних Oracle використовує MDL за замовчуванням. Алгоритм MDL успішно видалив шумні функції з 85 до 6. Деякі функції, такі як адреси кандидатів (місто або навіть країна), були видалені алгоритмом, оскільки це можливість працевлаштування онлайн. Дивно, однак, освітній статус і загальний досвід також не були обрані. Функції автоматичної підготовки даних були відключені з функції важливості атрибутів для спостереження за реальними результатами MDL. Ранги показують важливі атрибути роботи (рисунк 2.4).

variable	importance	rank
BD_10059	0.1855	1
BYD_1	0.1563	2
BD_10021	0.0223	3
BSS_26	0.0162	4
B_CALISMA_DURUMU	0.0112	5
BD_GECICI_TABLO_EVT__	0.0109	6

Рисунок 2.4 — Результати найвищого рангу відбору ознак за допомогою алгоритму MDL [35]

. Функції автоматичної підготовки даних були відключені з функції важливості атрибутів для спостереження за реальними результатами MDL. BD_10059: Вказує на те, що вони мають досвід роботи з посадою «Аналіз». BY_1: Знання мови «англійська» є важливим. BD_10021: Вони мають минулий досвід роботи «розробником програмного забезпечення». BSS_26: Володіє знаннями за спеціальністю «Аналіз». B_CALISMA_DURUMU: Працюють вони зараз чи ні. BD_GECICI_TABLO_EVT__: Вказує на те, що у кандидата є досвід роботи на будь-якій роботі.

2.5 Вилучення ознак

Перед виконанням алгоритму навчання часто необхідно розрізняти релевантні і нерелевантні дані. Отже, позбавлення від нерелевантних даних

покращує продуктивність алгоритму навчання. Вилучення ознак полягає в тому, щоб знайти високо репрезентативний набір ознак з оригіналів. На відміну від відбору ознак, виділення ознак є спробою знайти значуще значення при менш розмірному представленні оригіналу [38].

Видалення нерелевантних ознак підвищує продуктивність моделі навчання. Однак недоліком вилучення ознак є лінійна комбінація ознак, яка в цілому не піддається інтерпретації, тому не завжди можна розрахувати, скільки вихідних даних було втрачено. Аналіз головних компонент (PCA) є одним із широко використовуваних методів, який створює матрицю коваріації та її власні вектори (кожен з яких представляє певну частку дисперсії в даних).

Для перевірки результатів алгоритму PCA в якості алгоритму навчання замість SVM була використана узагальнена лінійна модель (GLM). З них 60% було використано для навчання алгоритму GLM, а 40% набору даних було використано для перевірки продуктивності. Після підготовки даних 98 ознак резюме залишилися як початкові виміри алгоритму навчання.

Алгоритм PCA перетворив початкові 98 функцій на 25 нових. Крім того, відсоток перших трьох нових функцій у розпізнаванні всього набору даних, відповідно, такий: 60,19%, 22,15% і 8,68%. Коли всі 98 функцій були протестовані (з GLM) без застосування запропонованих функцій вилучення ознак, точність становила 65,0%. Однак, якщо вилучення ознак застосовувалося за допомогою алгоритму PCA, точність знижувалася до 37,5%.

2.6 Модель машинного навчання та експерименти

В якості алгоритму навчання використовується SVM. Використовуючи SVM, він успішно класифікував, чи підходять резюме кандидатів для включення до шорт-листа. Для навчання алгоритму навчання були протестовані різні підмножини функцій для досягнення найкращого результату для посадки.

Дані були розділені на 60% на 40%, щоб заповнити SVM як набір даних. Якщо вибір ознак не був застосований, точність моделі була виявлена на рівні 62,5%. Початкові параметри параметрів наведені в таблиці 2.1.

Таблиця 2.1 — Без вибору функцій — лінійний SVM

Модель	Параметр	Цінність
Без вибору функцій	Значення коефіцієнта складності	10
Без вибору функцій	Кількість ітерацій	30
Без вибору функцій	Розв'язувач SVM	SVMS_SOLVER_IPM
Без вибору функцій	Толерантність до конвергенції	0.0001
Без вибору функцій	Ядра	SVMS_LINEAR
Без вибору функцій	Кількість функцій	82

Оскільки було збережено багато функцій, такий результат здається нормальним. Якщо використовуються тільки ті ознаки, які алгоритм MDL виявляє як значущі, то при лінійній точності ядра становить 0,78. Початкові параметри параметрів представлені в таблиці 2.2.

Таблиця 2.2 — Лінійний SVM з вибором MDL

Модель	Параметр	Цінність
Без вибору функцій	Значення коефіцієнта складності	10
Без вибору функцій	Кількість ітерацій	30
Без вибору функцій	Розв'язувач SVM	SVMS_SOLVER_IPM
Без вибору функцій	Толерантність до конвергенції	0.0001
Без вибору функцій	Ядра	SVMS_LINEAR
Без вибору функцій	Кількість функцій	6

Із загальної кількості вибірок 80% були використані як тренувальні дані, а решта 20% — як тестові дані. Якщо використовувалося 70–30%, то алгоритм реагував найгірше і знаходив лише п'ять атрибутів як важливі. Алгоритм SVM спочатку використовувався з усіма функціями та рекомендованими MDL функціями з лінійним ядром. Було використано ядро Гауса. Допустиме відхилення було встановлено на рівні 0,001 (початкові налаштування параметрів представлені в таблиці 2.3), а точність склала 0,923. Незважаючи на те, що цей результат показує задовільну точність, недостатньо для того, щоб оцінити і прийняти модель, просто

подивившись на точність.

Таблиця 2.3 — Gaussian SVM з вибором функцій MDL

Модель	Параметр	Цінність
Без вибору функцій	Розв'язувач SVM	SVMS_SOLVER_IPM
Без вибору функцій	Кількість ітерацій	11
Без вибору функцій	Значення стандартного відхилення	1.7320
Без вибору функцій	Толерантність до конвергенції	0.001
Без вибору функцій	Ядра	Гауссів
Без вибору функцій	Кількість функцій	6

Після спостереження за кожною мірою алгоритмів машинного навчання для оцінки моделі, ядро Гаусса та значення допуску 0,001 було визнано найкращим. Незважаючи на те, що деякі проекти вимагають високого запам'ятовування, наприклад, медичні проблеми, ця проблема не змушує отримувати високе значення. Однак відкликання проекту склало 0,875, що достатньо для прийняття моделі. З розподілом графіка прогнозування можна ознайомитися нижче (рисунок 2.5).

Оскільки ймовірність 0,5 розділяє два класи, істинно негативні (TN) прогнози вищі, ніж інші, а істинно-позитивні (TP) прогнози також високі. За даними Park et al., крива робочих характеристик приймача (ROC) — це чутливість і специфічність TP/позитивних значень над TN/негативними значеннями. Це також можна визначити як ставки ТП і помилково-позитивні (ФП). Що ще важливіше, площа під кривою roc (AUC) вважалася мірою загальної продуктивності тренувальної моделі.

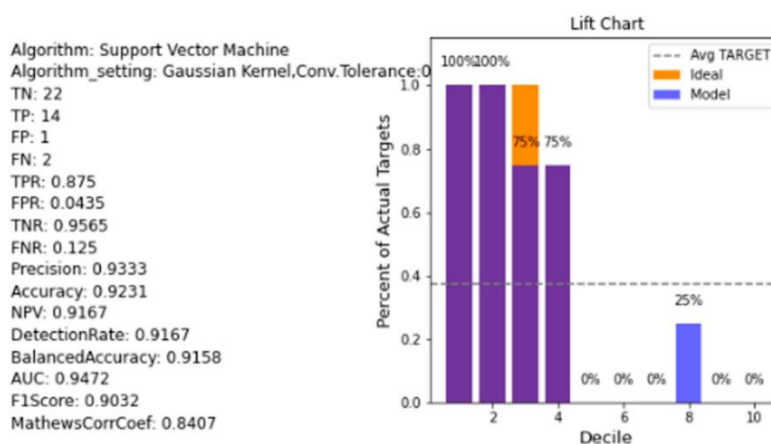


Рисунок 2.5 — Результати SVM з гаусовим ядром і толерантністю до збіжності

Таким чином, необхідно створити модель з ROC-графіком вище опорної лінії, щоб вважатися успішною. Що ще важливіше, AUC є глобальним показником здатності оцінювати здатність моделі розрізняти двійкові класи [39]. Модель SVM у проекті дала 0,9472. Значення AUC показує, що модель може розділяти претендентів як важливих і створювати шорт-лист для найму. Криві AUC і ROC показані на рисунку 2.6.

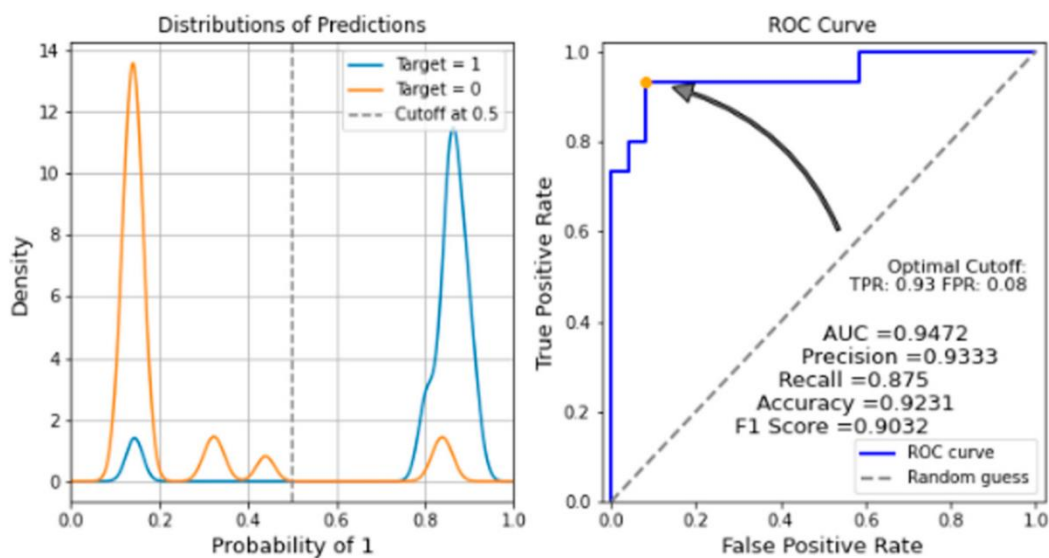


Рисунок 2.6 — Результати ROC та AUC SVM з гаусовим ядром та толерантністю до конвергенції 0,001

Лінійна гранична лінія/опорна лінія ($X = Y$) представляє частоту помилково-позитивних значень, яка така ж, як і швидкість істинно-позитивних значень.

3 ТЕХНОЛОГІЇ СТВОРЕННЯ КОНСТРУКТОРІВ ДЛЯ РЕЗЮМЕ ТА CV ЗІ ШТУЧНИМ ІНТЕЛЕКТОМ

3.1 Аналіз конструкторів резюме та CV зі штучним інтелектом

Інструменти на основі штучного інтелекту дозволяють модифікувати спосіб підготовки кандидатів до подачі заявок на роботу, пропонуючи розумніші, швидші та більш індивідуальні рішення, ніж традиційні методи. Ці платформи надають індивідуальні рекомендації, оптимізують використання ключових слів для систем відстеження кандидатів (ATS) і допомагають шукачам роботи узгодити свої резюме з галузевими стандартами — і все це за допомогою кількох кліків.

Нами проведено аналіз найкращих рішень на основі штучного інтелекту, які допомагають користувачам створювати привабливі резюме та резюме, поєднуючи інтелектуальну автоматизацію та зручний дизайн, щоб покращити пошук роботи.

VisualCV — це універсальна онлайн-платформа для пошуку роботи, яка пропонує набір інструментів для створення професійних резюме, супровідних листів і онлайн-портфолію. Його визнають за комплексний підхід до професійної презентації.

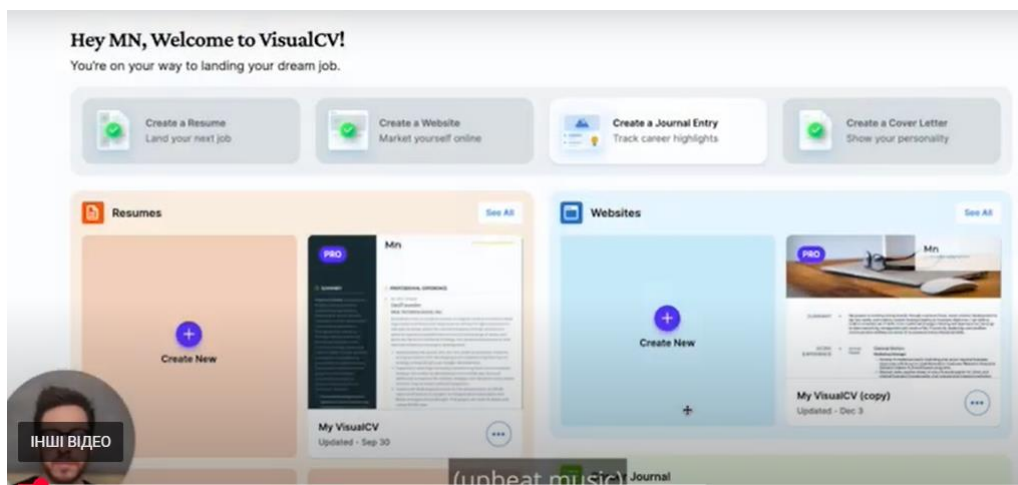


Рисунок 3.1 — VisualCV універсальна онлайн-платформа для пошуку роботи

Платформа має понад 30 настроюваних шаблонів резюме, що дозволяє створення резюме з урахуванням різних галузей і рівнів кар'єри. Конструктор резюме зі штучним інтелектом VisualCV виводить процес створення резюме на новий рівень, дозволяючи користувачам виділятися в заявках на роботу.

Основні характеристики VisualCV. Різноманітні шаблони резюме. Понад 30 настроюваних шаблонів для різних галузей. Створення резюме на основі AI. Революційне створення резюме за допомогою передової технології ШІ. Конструктор супровідних листів. Для виготовлення індивідуальних супровідних листів. Створення онлайн-портфолію. Демонструє роботу та досягнення в персоналізованому онлайн-портфолію. Відновити аналіз. Пропонує статистику продуктивності резюме з відстеженням переглядів і завантажень. Глобальна платформа. Підтримує дев'ять мов, обслуговуючи шукачів роботи в понад 100 країнах.

Teal — виділяється як комплексна платформа для кар'єрного зростання, яка пропонує конструктор резюме на основі штучного інтелекту та інтуїтивно зрозумілий інструмент керування пошуком роботи. Його інноваційний підхід до створення резюме та відстеження роботи робить його цінним союзником для шукачів роботи, які прагнуть оптимізувати свій пошук.

Функція створення резюме використовує технологію штучного інтелекту та попередньо створені «цеглинки» для різних розділів резюме, таких як резюме, досягнення та освіта. Цей інструмент порівнює резюме з описом посади в Job Tracker і забезпечує зворотний зв'язок у режимі реального часу щодо оптимізації ключових слів.

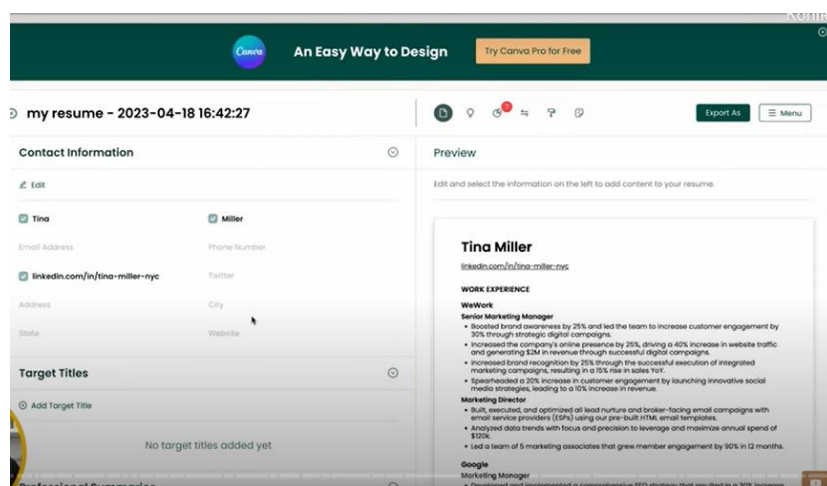


Рисунок 3.2 — Teal, як комплексна платформа для кар'єрного зростання

Job Tracker від Teal покращує пошук роботи, дозволяючи користувачам додавати вакансії в закладки, відстежувати заявки та керувати інформацією, пов'язаною з роботою. Цей інструмент особливо корисний для підтримки організації та зосередженості під час процесу пошуку роботи.

Ключові характеристики Teal. Конструктор резюме на основі AI. Створює резюме із попередньо створеними розділами та відгуками AI. Job Tracker. Керує пошуком роботи за допомогою комплексної системи відстеження. Інтеграція ШІ в поля вільної форми. Створює індивідуальні резюме та супровідні листи. Функція статистики. Надає інформацію про перегляди резюме та розташування глядачів. Шаблони електронних листів. Необмежена кількість шаблонів для обміну ресурсами та вмістом, пов'язаним із роботою. Варіанти ціноутворення. Пропонує безкоштовну версію та платну підписку Teal+.

JobScan — ШІ-редактор резюме від JobScan допомагає при пошуку роботи створювати програми, які виділяються завдяки використанню штучного інтелекту для швидкого створення відповідних ключових фраз, адаптації резюме та визначення важливих навичок. У ньому є вбудовані шаблони, сумісні з Applicant Tracking Systems (ATS) – програмним забезпеченням, яке використовується компаніями для фільтрації, сортування та ранжування резюме на основі ключових слів і критеріїв конкретної роботи.

Це забезпечує відповідність резюме користувачів конкретним вимогам роботи безпосередньо з уніфікованої інформаційної панелі. Інструмент консультує щодо ключових слів і навичок, узгоджених з посадовими інструкціями, оптимізуючи резюме для сканування ATS, що має вирішальне значення для проходження початкових перевірок і привернення уваги менеджерів з найму.

Основні функції включають сумісні з ATS завантаження PDF, оптимізацію розміру шрифту та заголовків, відповідне форматування дати та макети, які легко сканувати. Інтегруючи інформацію від професійних рекрутерів, JobScan посилює вплив резюме, поєднуючи ефективність штучного інтелекту з нюансами людського досвіду, щоб представити навички та досвід у спосіб, який значно покращує видимість для рекрутерів і менеджерів з найму.

Huntr AI Resume Builder дозволяє при пошуку роботи швидко й ефективно створювати професійні, зручні для ATS резюме. Він пропонує сім настроюваних шаблонів, пропозиції вмісту, створені штучним інтелектом, і інтеграцію ключових слів для оптимізації резюме для заявок на роботу. Платформа дозволяє користувачам адаптувати свої резюме до конкретних ролей, отримувати миттєвий зворотний зв'язок і бали відповідності вакансії, а також зберігати кілька версій для різних програм.



Рисунок 3.3 — Huntr AI Resume Builder

Додаткові функції включають створені штучним інтелектом супровідні листи, завантажувані PDF-файли та інтеграцію з описом посад, щоб підкреслити відповідні навички та кваліфікацію. Інструмент спрямований на оптимізацію процесу створення резюме, що робить його швидшим і ефективнішим для забезпечення співбесід.

ResumeUp.AI спрощує створення професійних, зручних для ATS резюме за допомогою інструментів на основі штучного інтелекту. Користувачі можуть швидко створювати або завантажувати резюме, вибирати з 20+ шаблонів і отримувати персоналізовані пропозиції щодо оптимізації для систем ATS.

Додаткові функції включають аналіз резюме, оптимізацію профілю LinkedIn і засіб відстеження заявок на роботу для спрощення процесу пошуку роботи. Гнучкі плани підписки пропонують такі варіанти, як оцінка резюме, генератор супровідних листів зі штучним інтелектом і необмежену кількість завантажень. ResumeUp.AI, якому довіряють понад 20,000 XNUMX користувачів і професіоналів у провідних

компаніях, є найкращим інструментом для створення видатних резюме та підвищення кар'єрних перспектив.

Ключові характеристики — Конструктор резюме AI Швидко створює зручні для ATS резюме за допомогою експертної оптимізації. Професійні шаблони із понад 20 галузевих дизайнів. Комплексні інструменти включає аналіз резюме та оптимізацію LinkedIn.

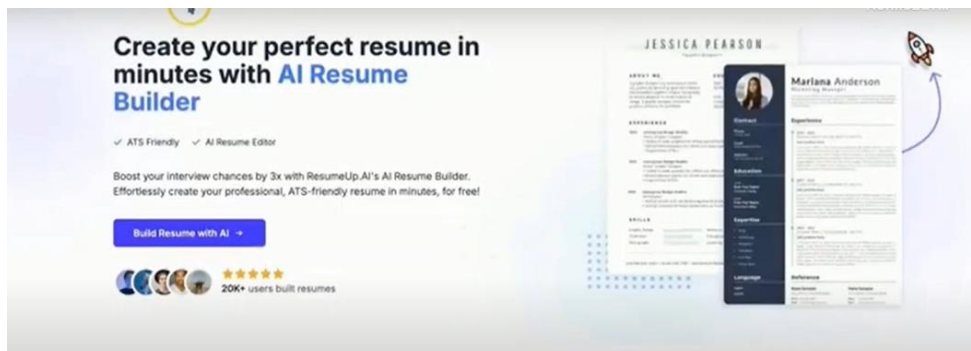


Рисунок 3.4 — ResumeUp.AI

Resume.io — це набагато більше, ніж звичайний конструктор резюме; це комплексна платформа, розроблена для спрощення процесу створення резюме для шукачів роботи в усьому світі. Завдяки надійним функціям і зручному інтерфейсу Resume.io є найкращим вибором для тих, хто хоче створити вражаючі професійні резюме.

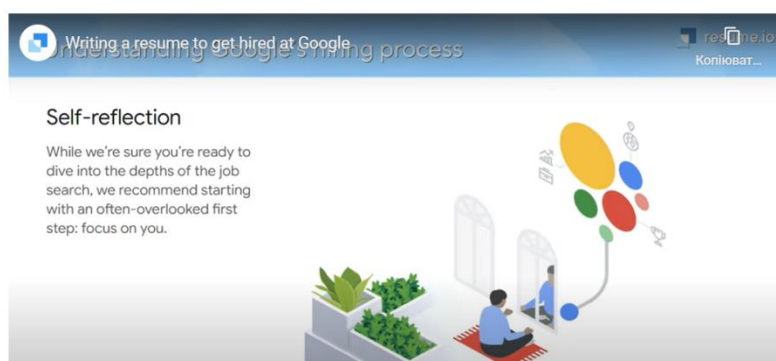


Рисунок 3.5 — Resume.io

Ця платформа пропонує різноманітний набір із понад 25 шаблонів резюме, гарантуючи, що користувачі зможуть знайти ідеальний варіант для своєї галузі та

особистого стилю. Кожен шаблон створено відповідно до професійних стандартів і підвищує шанси користувача привернути увагу роботодавця.

Resume.io також розуміє важливість універсальності форматів документів. Користувачі можуть завантажувати свої резюме у форматах PDF, DOCX і TXT, забезпечуючи гнучкість, необхідну для різноманітних вимог щодо роботи.

Основні характеристики Resume.io. Широкий вибір шаблонів, понад 25 професійних шаблонів для різних галузей і стилів. Кілька форматів завантаження та Гнучкість із форматами PDF, DOCX і TXT для різноманітних потреб додатків. Зручний інтерфейс, тобто Інтуїтивно зрозуміла платформа зі спадними меню для ефективного створення резюме. Функція електронної пошти та Легкий обмін ресурсами, як-от статті та приклади резюме, електронною поштою. Функція статистики – Відстежує перегляди резюме та місцезположення глядачів для цільових покращень резюме. Глобальне розширення: Послуги розширено для більшої кількості країн із доданими функціями, як-от конструктор супровідних листів.

Генератор резюме Jasper виділяється як універсальний помічник з написання штучного інтелекту, який вправно допомагає з різними письмовими завданнями, включаючи створення резюме. Відомий своїми генеративними можливостями штучного інтелекту, Jasper особливо корисний для подолання письменницького блоку та створення високоякісного контенту.



Рисунок 3.6 — Генератор резюме Jasper

Користувачі можуть вибрати з низки шаблонів і ввести свої дані, щоб розпочати роботу. Результати Jasper можна безпосередньо скопіювати в кінцевий

документ, або користувачі можуть вибрати завантаження готового резюме в бажаних форматах.

Основні характеристики Jasper. Генерує креативні ідеї вмісту та надає свіжі ідеї та зразки вмісту. Швидке керівництво – пропонує рекомендації щодо важливої інформації щодо резюме. Попередній перегляд у реальному часі та Дозволяє користувачам бачити, як виглядатиме їхнє резюме. Гнучкі параметри завантаження, а саме: легке завантаження у форматах PDF або Word.

Kickresume (див. рисунок 3.7) стає комплексним інструментом для створення резюме, оснащеним понад 50 настроюваними шаблонами, які підходять для різних галузей і рівнів кар'єри. Сильна сторона платформи полягає в її оптимізації контенту за допомогою штучного інтелекту, що допомагає користувачам точно адаптувати свої резюме відповідно до опису конкретної посади та забезпечує сумісність з ATS.

8. Кікресюме



Рисунок 3.6 — Інструмент для створення резюме Kickresume

Зокрема, Kickresume розширює свої пропозиції до конструктора супровідних листів, дозволяючи створювати персоналізовані супровідні листи для супроводу заявок на роботу. Інструмент також має параметри імпорту та експорту в різних форматах, як-от PDF і Word, що додає його універсальності.

Основні характеристики Kickresume. Велика колекція шаблонів: понад 50 галузевих шаблонів резюме, які можна налаштувати. Оптимізація контенту за допомогою AI та резюме для конкретних робіт і сумісності з ATS. Конструктор супровідних листів, що дозволяє створювати персоналізовані супровідні листи.

Гнучкий імпорт і експорт, підтримує різні формати файлів для зручності використання. Керований процес написання — надає пропозиції та приклади для створення резюме. Відстежує перегляди резюме та завантаження для стратегічних покращень.

3.2 Основні параметри засобів штучного інтелекту для генерації резюме

Інструменти ШІ мають переваги для пошуку роботи, які прагнуть виділитися: від подолання блоку автора за допомогою креативного створення вмісту для вашого резюме до забезпечення того, щоб резюме були дружніми до ATS та впливовими. Вони представляють передову технологію створення резюме, що дозволяє користувачам ефективніше представляти свої професійні історії та впевненіше орієнтуватися в конкурентному середовищі заявок на роботу.

Штучний інтелект створив величезну можливість для створення шорт-листа з інформації резюме кандидатів. Особливості залежать як від культури компанії, так і від оголошень про вакансії. Тому вибір ознак повинен бути застосований до моделі навчання. На жаль, алгоритм вилучення ознак, який був неконтрольованим методом, не вдалося використати в дослідженні. Крім того, не було оптимальних налаштувань для моделі навчання. Залежно від даних, модель SVM може давати різні результати для кожного параметра, такі як ядро, допуск або значення ітерацій. Таким чином, можна буде виявити, чи є раса, вік і стать абітурієнтів або випускників шкіл вхідними даними для відбору кандидатів. Це може змінити правила гри для компаній, щоб зберегти свою стійкість і конкурентну перевагу. Це дослідження доводить, що штучний інтелект можна використовувати для зменшення вимог до робочої сили та часових витрат під час створення шорт-листа на основі резюме кандидатів. Також було виявлено, що багато функцій резюме можна вважати шумними значеннями і про них потрібно подбати перед використанням алгоритму навчання. Можна виявити, яка частина резюме кандидатів враховується при складанні шорт-листа ВК раніше. Крім того, порядок важливості можна виявити між характеристиками резюме за допомогою алгоритму відбору даних. Резюме кандидата можна вважати набагато більш ефективним і

валідним для процесу підбору персоналу замість використання особистісних тестів. Для майбутніх проектів база кандидатів має бути більшою. Алгоритми вилучення ознак повинні бути розглянуті на предмет ефективності. Нарешті, NLP AI повинен бути інтегрований в модель, тому що в реальному житті резюме недостатньо достатньо. Лист про наміри та перевірку репутації в соціальних мережах можна поєднати з результатами висновків резюме.

4 РЕАЛІЗАЦІЯ ВЕБ-ДОДАТКУ ТА ШТУЧНОГО ІНТЕЛЕКТУ

4.1 Пояснення основного коду

В цьому розділі детально описується функціональність основного коду .
Генератор та аналізатор резюме — вебзастосунок на Flask.

Це простий вебзастосунок, створений на Flask, який дозволяє користувачам вводити інформацію для резюме, генерувати форматований текст резюме та отримувати базовий аналіз, зокрема підрахунок слів, виявлення email-адрес, номерів телефонів і згадування ключових навичок.

Основні можливості: поля для введення імені, email, короткої інформації, досвіду, освіти та навичок.

Генерація зручного для читання резюме на основі введених даних.

Аналіз резюме, зокрема: загальна кількість слів, кількість email-адрес, кількість номерів телефонів, список виявлених навичок із попередньо заданого набору.

Використані технології: Python, Flask, регулярні вирази, HTML-шаблони для введення даних та показу результатів.

Веб-додаток дозволяє користувачам: згенерувати резюме на основі введених даних, завантажити своє резюме для аналізу, отримати зворотний зв'язок (оцінку, поради щодо покращення, перевірку граматики тощо).

Технології Frontend: HTML, CSS, JavaScript, React (або простий шаблон на Bootstrap), Backend: Python (Flask або FastAPI), AI/НЛП: OpenAI GPT (через API), spaCy або інші NLP-бібліотеки, База даних: SQLite або PostgreSQL, Форматування резюме: PDF (бібліотека pdfkit, WeasyPrint, або reportlab).

Програмний код веб-додатку представлено в лістингах 4.1–4.5.

Лістинг 4.1 — Структура проєкту:

```

resume_app/
├── app.py # Основний файл Flask/FastAPI
├── templates/
│   └── index.html # HTML-форма для введення

```

```

├── static/
│   └── style.css # Стили
├── resume_generator.py # Модуль генерації резюме
├── resume_analyzer.py # Модуль аналізу
├── uploads/ # Завантажені резюме (PDF)
└── requirements.txt #

```

Лістинг 4.2 — Html код проекту

```

HTML –форма (index.html)
<!DOCTYPE html>
<html>
<head>
<title>Генерація та аналіз резюме</title>
</head>
<body>
<h1>Резюме: Генерація та Аналіз</h1>
<form action="/generate" method="post">
<input type="text" name="name" placeholder="Ім'я"><br>
<input type="text" name="position" placeholder="Посада"><br>
<textarea name="experience" placeholder="Досвід роботи">
</textarea><br>
<button type="submit">Згенерувати резюме</button>
</form>
<form action="/analyze" method="post" enctype="multipart/form-data">
<input type="file" name="resume"><br>
<button type="submit">Аналізувати резюме</button>
</form>
</body>
</html>

```

Лістинг 4.3 — Backend (Flask-приклад)

```

from flask import Flask, request, render_template, send_file
from resume_generator import generate_resume
from resume_analyzer import analyze_resume

app = Flask(__name__)

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/generate', methods=['POST'])
def generate():
    data = {

```

```

'name': request.form['name'],
'position': request.form['position'],
'experience': request.form['experience']
}
filepath = generate_resume(data)
return send_file(filepath, as_attachment=True)

@app.route('/analyze', methods=['POST'])
def analyze():
    file = request.files['resume']
    file.save(f'uploads/{file.filename}')
    feedback = analyze_resume(f'uploads/{file.filename}')
    return f'<h2>Аналіз резюме:</h2><pre>{feedback}</pre>'

    from fpdf import FPDF

```

ЛІСТИНГ 4.4 — Генерація резюме

```

def generate_resume(data):
    pdf = FPDF()
    pdf.add_page()
    pdf.set_font("Arial", size=12)
    pdf.cell(200, 10, txt=f'Ім'я: {data['name']}', ln=True)
    pdf.cell(200, 10, txt=f'Посада: {data['position']}', ln=True)
    pdf.multi_cell(200, 10, txt=f'Досвід: {data['experience']}")
    filename = f'resume_{data['name'].replace(' ', '_)}.pdf'
    filepath = f'uploads/{filename}'
    pdf.output(filepath)
    return filepath

```

ЛІСТИНГ 4.5 — Аналіз резюме

```

import fitz # PyMuPDF
import openai

openai.api_key = 'YOUR_API_KEY' # Зберігати безпечно!

def extract_text_from_pdf(filepath):
    doc = fitz.open(filepath)
    text = ""
    for page in doc:
        text += page.get_text()
    return text

def analyze_resume(filepath):
    text = extract_text_from_pdf(filepath)

```

```
prompt = f"Проаналізуй це резюме та дай поради: {text}"
response = openai.ChatCompletion.create(
    model="gpt-4",
    messages=[{"role": "user", "content": prompt}]
)
return response.choices[0].message['content']
```

Запуск додатку . `pip install -r requirements.txt python app.py` (рисунок 4.1)

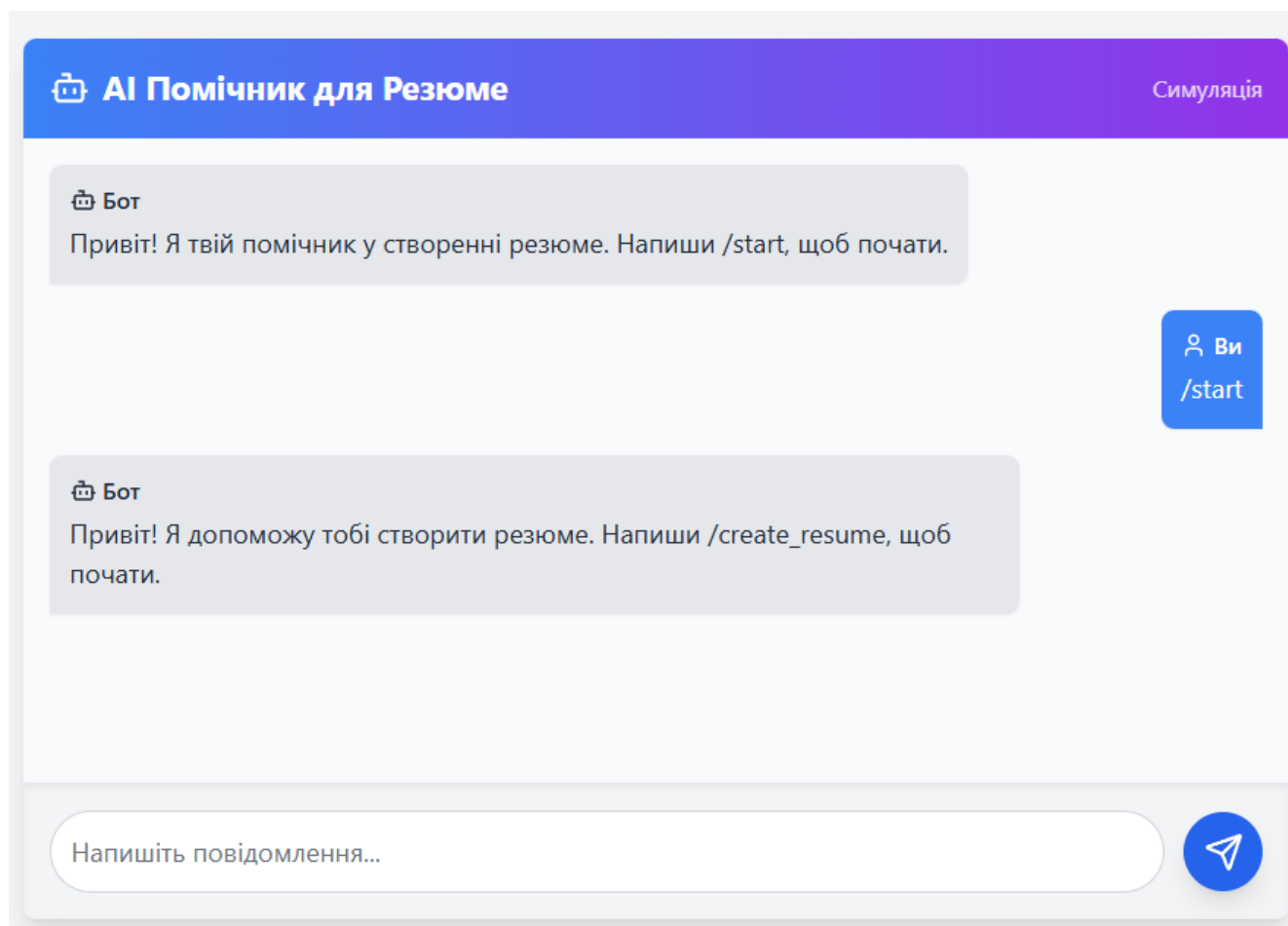


Рисунок 4.1 — Результат запуску коду

Також показано виведення інформації користувачем на рисунку 4.2, та як запускається аналіз резюме у чат боті на рисунку 4.3

Безпека та поліпшення: Додати обмеження на розмір PDF-файлів, валідація полів, можливість експортувати резюме у DOCX, індивідуальні шаблони резюме, статистика оцінювання (графіки, шкала, AI-рейтинг)

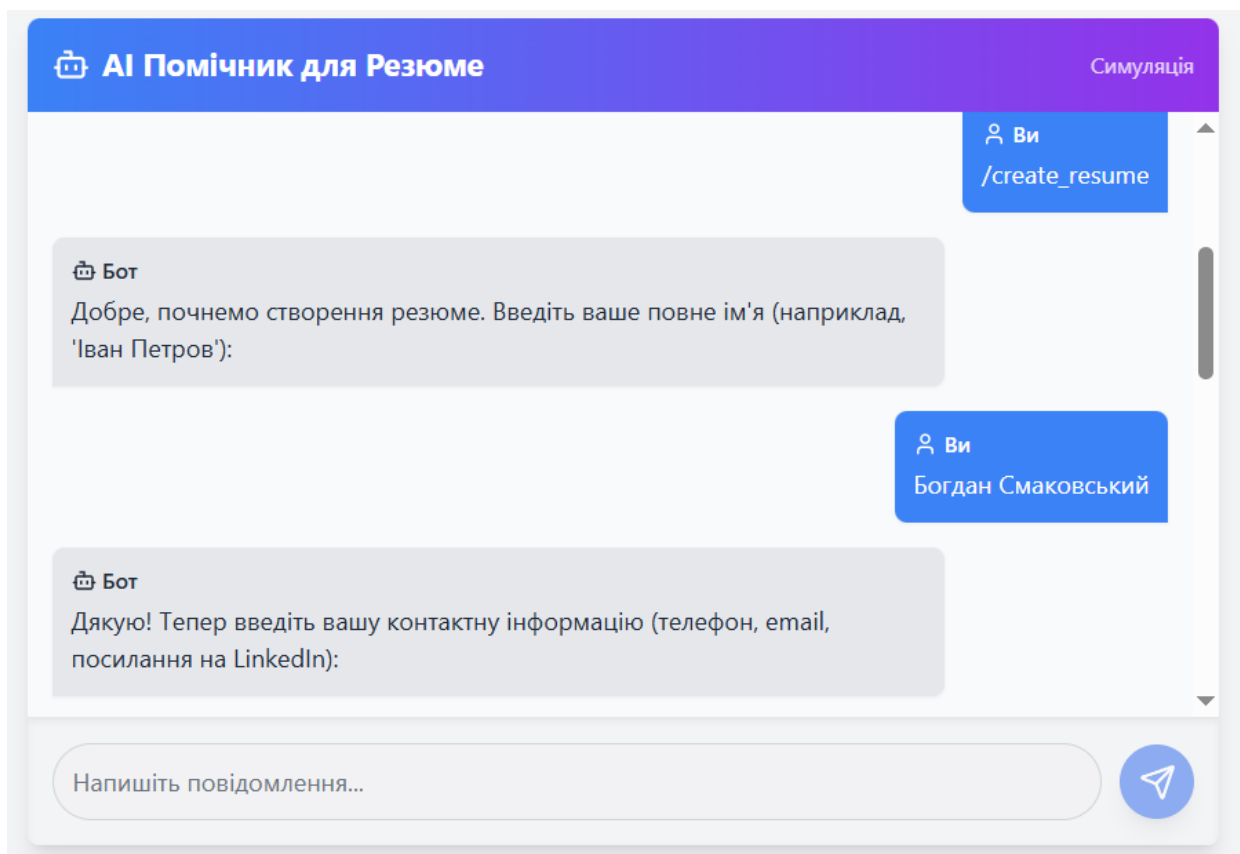


Рисунок 4.2 — Введення інформації користувачем

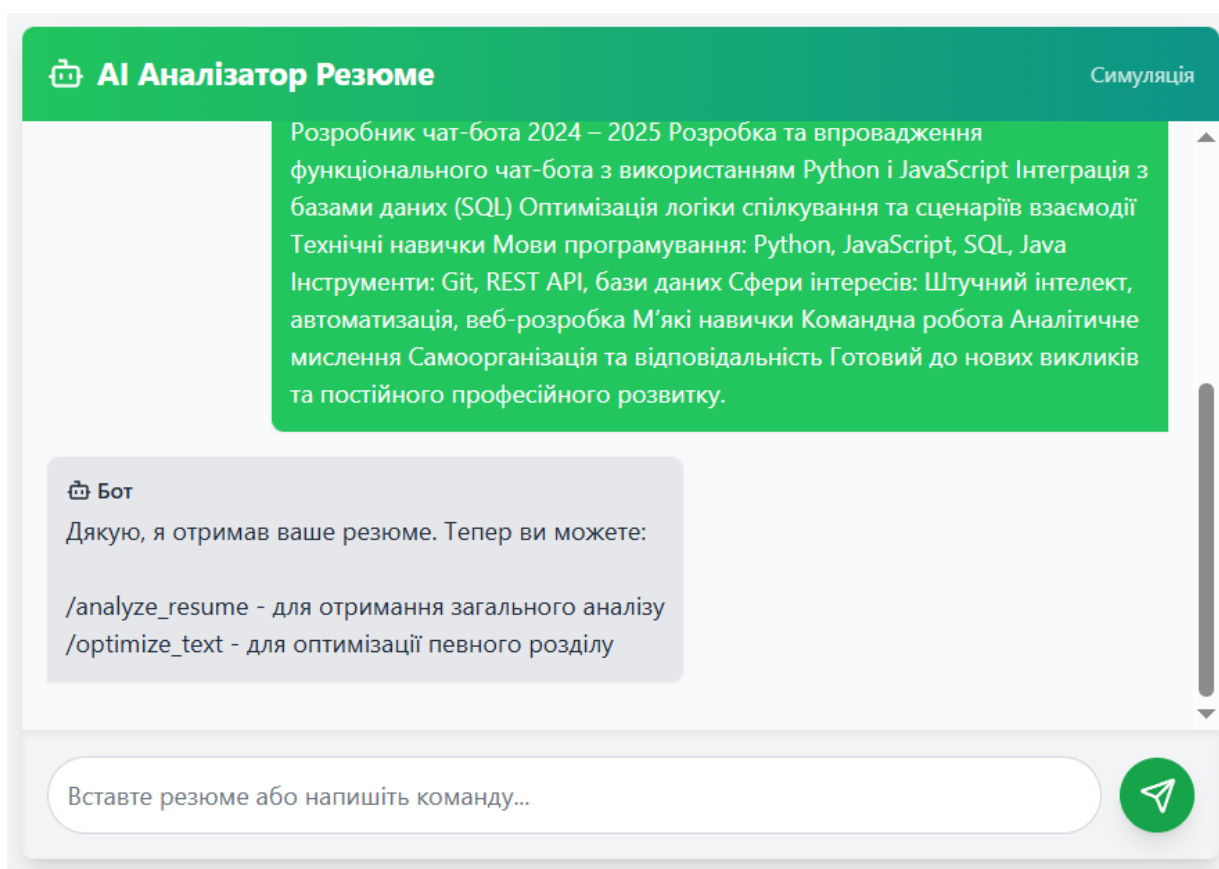


Рисунок 4.3 — Запуск аналізу резюме

Основні компоненти веб-додатку. Розроблений веб-додаток містить:

Main.py:

- основний файл, який запускає Flask сервер.
- містить маршрути для обробки різних HTTP-запитів (GET, POST);
- взаємодіє з Assistant API (assistant_api.py) через функцію
- create_assistant(client);
- використовує шаблони для відображення HTML-сторінок;
- взаємодіє з JavaScript (script.js) для функціонування чат-бота на стороні клієнта;
- підключається до бази даних SQLite (bookstore.db);
- виконує CRUD операції над таблицями бази даних (додавання нових записів, оновлення існуючих, видалення та відображення даних з таблиць Books, Customers, Orders, Publishers. Books).

Assistant_api.py:

- відповідає за створення та налаштування асистента через API OpenAI.
- Script.js:
- взаємодіє з Flask через HTTP-запити до маршрутів /start та /chat.
- відповідає за інтерактивність чат-бота, обробку подій та відправку/отримання повідомлень через API.
- Style.css:
- стилізує HTML сторінки та елементи інтерфейсу, включаючи чат-бота.
- HTML (index.html, orders.html, books.html, customers.html):
- шаблони HTML, які відображають дані, отримані з Flask серверу.
- взаємодіють з CSS для оформлення і JavaScript для функціональності.
- Bookstore.db:
- зберігає інформацію про книги, клієнтів, замовлення та видавців;

— икористовується для виконання CRUD операцій через Flask маршрути (додавання нових записів, оновлення існуючих, видалення та відображення даних з таблиць Books, Customers, Orders, Publishers. Books).

Діаграма класів. Об'єктно-орієнтовна модель компонентів додатку була розроблена за допомогою веб-сайту Plantuml (рис. 4.4).

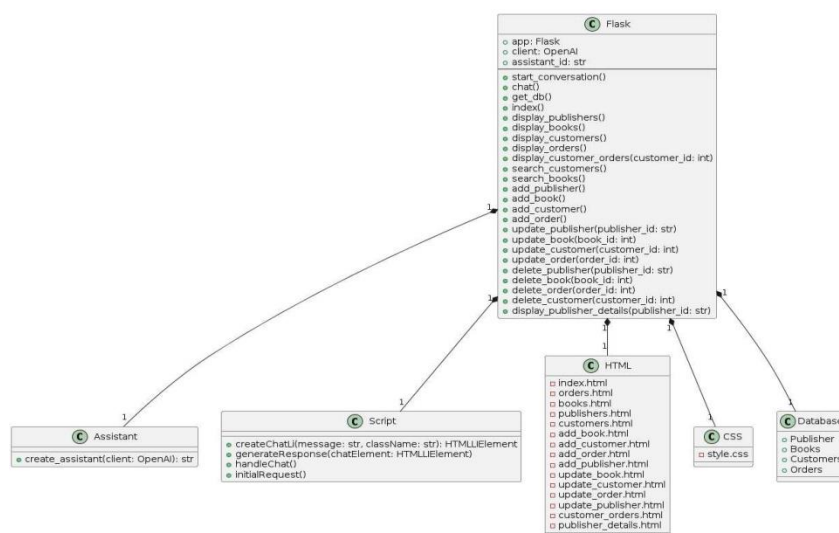


Рисунок 4.4 — Діаграма класів

Діаграма послідовностей (рис. 4.5) представляє собою процес обробки HTTP-запиту через сервер Flask.

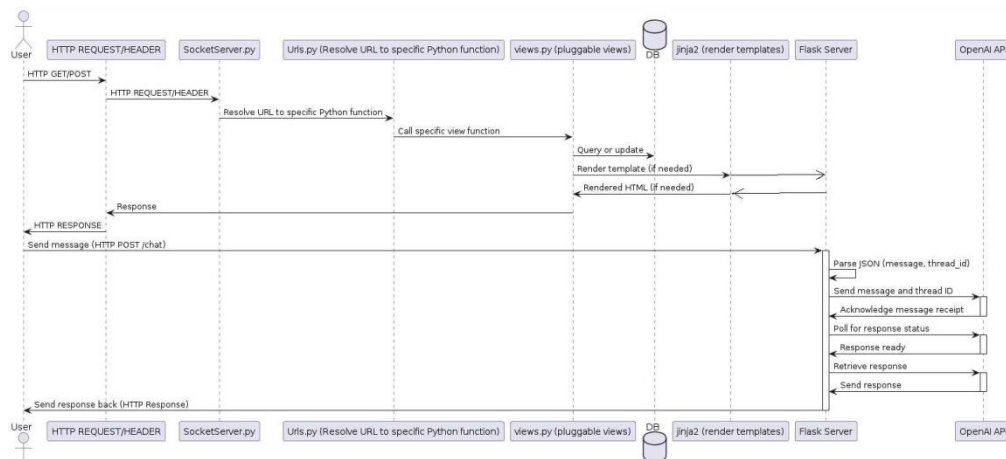


Рисунок 4.5 — Діаграма послідовностей

Діаграма послідовностей показує процес взаємодії компонентів програми.

В сучасному цифровому світі, де кар'єрні можливості постійно еволюціонують, інструменти автоматизації та штучного інтелекту відіграють

ключову роль у процесах пошуку роботи та професійного розвитку. Чат-бот, призначений для аналізу та оптимізації резюме, є яскравим прикладом такої синергії технологій. Його функціональність, що дозволяє користувачам отримувати миттєвий зворотний зв'язок та рекомендації щодо їхніх професійних документів, базується на складному поєднанні передових програмних рішень, інтегрованих на різних рівнях архітектури. Розуміння цих технологій є критично важливим для осмислення принципів роботи та потенціалу подібних інтелектуальних систем.

Центральним елементом архітектури такого бота є його серверна частина, яка традиційно реалізується за допомогою мови програмування Python. Python обирається завдяки своїй універсальності, багатій екосистемі бібліотек, простоті синтаксису та високій продуктивності для завдань, що включають обробку даних та інтеграцію з різноманітними сервісами.

У контексті розробки Telegram-бота, Python дозволяє ефективно взаємодіяти з Telegram Bot API, який служить основним каналом зв'язку між користувачем та програмною логікою. Цей API надає набір методів для відправлення повідомлень, отримання оновлень від користувачів, обробки команд та керування станами діалогу. Механізми взаємодії можуть включати довготривале опитування (long polling), де бот періодично запитує нові повідомлення, або вебхуки (webhooks), коли Telegram самостійно відправляє оновлення на вказаний сервером URL, що є більш масштабованим та ефективним підходом для високозавантажених систем. Асинхронні фреймворки в Python, такі як `asyncio` у поєднанні з бібліотеками для роботи з Telegram API (наприклад, `python-telegram-bot` або `aiogram`), дозволяють обробляти велику кількість одночасних запитів без блокування, забезпечуючи високу чуйність бота. Завдяки цій серверній логіці, бот може розпізнавати команди користувачів, зберігати контекст розмови, керувати послідовністю запитів (як це ілюструє UML діаграма станів) і викликати відповідні модулі для обробки даних.

Ключовим компонентом, що надає боту "інтелекту", є інтеграція зі штучним інтелектом, а саме з Великими Мовними Моделями (LLM), такими як Gemini або моделі OpenAI (наприклад, GPT-3.5 Turbo або GPT-4). Ці моделі є трансформаторними нейронними мережами, навченими на величезних обсягах

текстових даних, що дозволяє їм розуміти, генерувати та маніпулювати людською мовою з дивовижною гнучкістю та точністю. У контексті аналізу резюме, LLM використовуються для виконання кількох критично важливих завдань. По-перше, вони застосовуються для аналізу резюме, де модель отримує текст резюме та промпт із запитом на оцінку його повноти, структури, використання ключових слів, виявлення можливих граматичних чи стилістичних помилок, а також надання рекомендацій щодо покращення. Цей процес включає елементи обробки природної мови (Natural Language Processing, NLP), такі як ідентифікація сутностей (імен компаній, посад, дат), вилучення інформації та семантичний аналіз для розуміння змісту. По-друге, LLM використовуються для оптимізації тексту. Коли користувач просить оптимізувати певну частину резюме (наприклад, опис досвіду роботи або навички), модель отримує відповідний фрагмент тексту та інструкції щодо його переформулювання. Метою є зробити текст більш лаконічним, професійним, орієнтованим на досягнення та насиченим релевантними ключовими словами, що збільшує його привабливість для рекрутерів та систем автоматичного відбору. Ефективність цієї взаємодії з LLM значною мірою залежить від "промпт-інжинірингу" – мистецтва формулювання чітких, детальних та ефективних запитів до моделі, щоб отримати бажаний результат.

Для забезпечення стійкості та збереження даних між сесіями користувачів, критично важливим є використання системи управління базами даних (СУБД). У залежності від масштабу та складності проекту, це може бути реляційна база даних, така як SQLite (для невеликих, локальних застосунків або прототипів), PostgreSQL (для великих, масштабованих систем з комплексними відносинами даних), або NoSQL база даних, така як MongoDB (для гнучкого зберігання неструктурованих або напівструктурованих даних, наприклад, JSON-об'єктів резюме). База даних служить для зберігання повних текстів резюме користувачів, їхніх контактних даних, історії взаємодії з ботом, а також, можливо, проміжних станів розмови, що дозволяє боту "пам'ятати" користувача і продовжувати діалог з того місця, де він був перерваний. Ефективне управління даними забезпечує персоніфікацію сервісу та можливість повертатися до раніше створених чи оптимізованих резюме.

Візуальне моделювання поведінки бота, представлене за допомогою UML діаграми станів, є невід'ємною частиною процесу розробки програмного забезпечення. Діаграми станів у рамках Уніфікованої Мови Моделювання (UML) дозволяють формально описати динамічну поведінку об'єкта або системи, відображаючи всі можливі стани, у яких вона може перебувати, та переходи між цими станами. Кожен стан представляє певний етап у життєвому циклі бота, наприклад, "Очікування повного резюме", "Опції доступні" або "Очікування частини для оптимізації". Переходи між станами ініціюються певними подіями (наприклад, введення користувачем команди `/analyze_resume`) або діями (наприклад, отримання результатів від AI). Ця діаграма служить не лише для документування, але й для проектування, допомагаючи розробникам чітко уявити потік взаємодії, виявити потенційні помилки логіки або відсутні переходи, а також забезпечити повне покриття всіх можливих сценаріїв користувачів. Використання таких інструментів, як Mermaid для генерації діаграм з текстового опису, спрощує процес моделювання та інтеграції діаграм безпосередньо в документацію коду.

Хоча основна логіка бота реалізована на Python, у контексті симуляції або розробки веб-інтерфейсу для управління чи перегляду даних, може бути залучений React. React є бібліотекою JavaScript для побудови динамічних, інтерактивних користувацьких інтерфейсів. Він дозволяє створювати компоненти, які ефективно оновлюються у відповідь на зміни даних, забезпечуючи плавну та чуйну взаємодію. У типовій архітектурі, React-додаток (frontend) взаємодіює з Python-сервером (backend) через RESTful API або WebSockets. Це забезпечує чітке розділення відповідальності між представленням даних та бізнес-логікою, дозволяючи незалежну розробку та масштабування обох частин системи. У випадку симуляції, React імітує поведінку Telegram-бота, візуалізуючи діалог та відгуки, що дозволяє розробникам тестувати та вдосконалювати логіку бота без необхідності розгортання повноцінного Telegram-бота.

Нарешті, ефективне функціонування такого бота неможливе без належного розгортання та хостингу. Серверний додаток на Python повинен бути розміщений на надійній платформі, яка забезпечує безперервну доступність та

масштабованість. Хмарні платформи, такі як Google Cloud Platform (GCP), Amazon Web Services (AWS) або Microsoft Azure, надають інфраструктуру та сервіси (віртуальні машини, контейнерні платформи, керовані бази даних), необхідні для розгортання та ефективного управління додатком. Використання контейнеризації (наприклад, за допомогою Docker) та оркестрації (наприклад, Kubernetes) дозволяє стандартизувати середовище розгортання, спростити масштабування та забезпечити високу доступність бота. Без надійної інфраструктури, навіть найдосконаліший бот не зможе забезпечити безперебійний сервіс для своїх користувачів.

Таким чином, розробка інтелектуального чат-бота для аналізу та оптимізації резюме є багатошаровим процесом, що спирається на глибоке розуміння та вміле поєднання різноманітних технологій. Від міцної серверної основи на Python, що взаємодіє з Telegram API, до передових можливостей штучного інтелекту, що забезпечують розуміння та генерацію природної мови, від систем управління базами даних, що гарантують збереження інформації, до інструментів моделювання, що формалізують поведінку системи, – кожен елемент є критично важливим для створення ефективного, надійного та корисного інструменту для професійного розвитку. Це свідчить про те, як сучасні технології, працюючи в синергії, можуть створювати інноваційні рішення для реальних потреб користувачів.

4.2 Результати дослідження

У сучасному цифровому ландшафті, де динаміка ринку праці та кар'єрні траєкторії зазнають постійних трансформацій, передові інструменти автоматизації та інноваційні рішення у сфері штучного інтелекту відіграють першорядну роль у оптимізації процесів пошуку роботи та особистісного професійного розвитку. Чат-бот, спеціалізований на функціях аналізу та оптимізації резюме, являє собою яскравий приклад синергетичного поєднання різнорідних технологій, що забезпечує користувачам можливість отримувати миттєвий та цільовий зворотний зв'язок, а також рекомендації щодо вдосконалення їхніх професійних документів.

Глибоке розуміння технологічних засад, що лежать в основі такого функціоналу, є невід'ємною передумовою для усвідомлення принципів роботи, архітектурних рішень та загального потенціалу подібних інтелектуальних систем. Цей аналіз розкриває багатогранність програмних компонентів та їхню інтеграцію, що формує цілісну та ефективну платформу для підтримки кар'єрних амбіцій.

Центральним архітектурним компонентом подібного бота є його серверна частина, яка традиційно реалізується за допомогою інтерпретованої високорівневої мови програмування Python. Вибір Python зумовлений його винятковою універсальністю, обширною та постійно зростаючою екосистемою бібліотек, що охоплює широке коло задач, простотою та читабельністю синтаксису, що сприяє прискореній розробці, а також високою продуктивністю для виконання завдань, пов'язаних з обробкою даних, мережевим взаємодією та інтеграцією з різноманітними зовнішніми сервісами.

У контексті розробки Telegram-бота, Python надає інструменти для ефективної та надійної взаємодії з Telegram Bot API. Цей інтерфейс прикладного програмування (API) служить основним каналом зв'язку, що опосередковує комунікацію між кінцевим користувачем та базовою програмною логікою бота. Він пропонує стандартизований набір методів для здійснення операцій, таких як відправлення текстових повідомлень, фотографій, документів, отримання оновлень від користувачів, обробка специфічних команд, що вводяться користувачем (наприклад, /start, /analyze_resume), та комплексне керування станами діалогу в рамках багатоетапних розмов.

Механізми отримання оновлень від Telegram Bot API можуть варіюватися: від довготривалого опитування (long polling), коли бот періодично надсилає запити на сервер Telegram для перевірки наявності нових повідомлень, до використання вебхуків (webhooks), де сервер Telegram самостійно ініціює HTTP POST запит на заздалегідь визначений URL-адресу, надану розробником бота, щойно відбувається подія.

Останній підхід, заснований на вебхуках, вважається більш масштабованим та ефективним для систем з високим очікуваним навантаженням, оскільки він

зменшує накладні витрати на постійне опитування. Для забезпечення високої чутливості та здатності обробляти значну кількість одночасних запитів без виникнення блокування, що є критично важливим для інтерактивних систем, використовуються асинхронні фреймворки в Python. Зокрема, вбудована бібліотека `asyncio` у поєднанні з високопродуктивними асинхронними бібліотеками для роботи з Telegram API (такими як `python-telegram-bot` або `aiogram`) дозволяє реалізувати неблокуючий ввід/вивід. Це означає, що під час очікування відповіді від зовнішніх сервісів (наприклад, Telegram або AI API) додаток не "зависає", а може продовжувати обробляти інші запити або виконувати інші завдання. Завдяки цій міцній серверній логіці, бот має можливість ефективно розпізнавати команди, які вводять користувачі, зберігати контекст поточної розмови та попередніх взаємодій, керувати складною послідовністю запитів та відповідей, що відображено у його UML діаграмі станів, а також динамічно викликати відповідні модулі для виконання спеціалізованої обробки даних та взаємодії з інтелектуальними компонентами системи.

Фундаментальним елементом, що надає боту його "інтелектуальні" можливості та відрізняє його від простих сценарних систем, є глибока інтеграція зі штучним інтелектом, зокрема з Великими Мовними Моделями (LLM). Серед провідних прикладів таких моделей можна виділити Gemini від Google або моделі OpenAI, зокрема GPT-3.5 Turbo чи GPT-4. Ці моделі представляють собою вершину досягнень у галузі глибинного навчання, реалізовані як складні трансформаторні нейронні мережі. Їхня архітектура дозволяє їм паралельно обробляти великі обсяги даних та ідентифікувати складні залежності у послідовностях. Моделі LLM проходять навчання на колосальних, різноманітних та багатомовних корпусах текстових даних, що наділяє їх винятковою здатністю розуміти нюанси людської мови, генерувати зв'язний та контекстуально релевантний текст, а також маніпулювати мовними конструкціями з дивовижною гнучкістю, точністю та креативністю. У контексті функціоналу аналізу та оптимізації резюме, LLM використовуються для виконання кількох ключових та критично важливих завдань, які виходять за межі простого пошуку за шаблоном. По-перше, вони

застосовуються для глибокого аналізу резюме. У цьому процесі модель отримує повний текст резюме від користувача разом з чітко сформульованим промптом, що містить запит на комплексну оцінку документа. Оцінка включає такі аспекти, як повнота представленої інформації, її логічна структура, ефективність використання ключових слів, що є актуальними для бажаної вакансії чи галузі, виявлення потенційних граматичних, орфографічних чи стилістичних помилок, а також надання конкретних, дієвих рекомендацій щодо загального покращення. Цей аналітичний процес не є поверхневим, а включає складні елементи обробки природної мови (Natural Language Processing, NLP), такі як ідентифікація та вилучення сутностей (наприклад, імен компаній, назв посад, релевантних дат, назв навчальних закладів), розпізнавання іменованих сутностей (Named Entity Recognition, NER), аналіз взаємозв'язків між ними, а також глибокий семантичний аналіз для повного розуміння контекстуального змісту та підтексту резюме. По-друге, LLM використовуються для цілеспрямованої оптимізації текстових фрагментів резюме. Коли користувач ініціює запит на оптимізацію певної частини свого резюме – будь то детальний опис досвіду роботи, перелік професійних навичок, формулювання освітніх досягнень або навіть вступний абзац – модель отримує відповідний фрагмент тексту разом з чіткими інструкціями щодо бажаного стилю та мети переформулювання. Основною метою такої оптимізації є перетворення тексту на більш лаконічний, професійний, переконливий та орієнтований на демонстрацію досягнень, а також насичення його релевантними ключовими словами, що є критично важливим для проходження початкових етапів автоматизованого відбору кандидатів (Applicant Tracking Systems, ATS) та для привернення уваги рекрутерів. Ефективність цієї взаємодії з LLM, що є основою для високоякісних результатів, значною мірою залежить від "промпт-інжинірингу" – спеціалізованої дисципліни, що вивчає мистецтво та науку формулювання чітких, детальних, контекстуально точних та ефективних запитів до мовних моделей з метою отримання максимально релевантних та бажаних вихідних результатів. Це передбачає розуміння поведінки моделі та вміння направляти її генерацію.

Для забезпечення безперервної стійкості функціонування бота та можливості

збереження даних між різними сесіями взаємодії з користувачами, застосування надійної системи управління базами даних (СУБД) є абсолютно критичним. Вибір конкретної СУБД визначається масштабом проекту, очікуваним навантаженням та складністю структури даних, які необхідно зберігати. Для невеликих проектів або етапів прототипування може бути достатньо використання реляційної бази даних, такої як SQLite, яка вбудовується безпосередньо в додаток і не потребує окремого серверного процесу. Для середніх та великих, високомасштабованих систем, що оперують комплексними відносинами між даними, перевага віддається потужним реляційним СУБД, як-от PostgreSQL. PostgreSQL відома своєю надійністю, розширеним функціоналом, підтримкою складних запитів та можливістю ефективно обробляти великі обсяги структурованих даних. В альтернативному сценарії, коли необхідно зберігати неструктуровані або напівструктуровані дані (наприклад, резюме у форматі JSON-об'єктів), або коли вимагається висока гнучкість схеми даних, оптимальним вибором може бути NoSQL база даних, наприклад, MongoDB. MongoDB пропонує документо-орієнтовану модель, що дозволяє зберігати дані у форматі, подібному до JSON (BSON), забезпечуючи високу продуктивність та масштабованість для певних типів додатків. База даних служить для персистентного зберігання повних текстів резюме користувачів, їхніх контактних даних, історії взаємодії з ботом, включаючи попередні запити та отримані відповіді, а також, що не менш важливо, проміжних станів розмови. Остання функція є критично важливою для забезпечення безшовного користувацького досвіду, оскільки вона дозволяє боту "пам'ятати" кожного користувача, відновлювати діалог з того моменту, де він був перерваний, та надавати контекстуально релевантну підтримку. Ефективне управління даними у СУБД гарантує не лише збереження інформації, але й забезпечує персоніфікацію сервісу, дозволяючи користувачам повертатися до раніше створених, аналізованих чи оптимізованих версій своїх резюме, а також відстежувати прогрес у вдосконаленні їхніх професійних документів.

Візуальне моделювання поведінки бота, що формалізовано представлене за допомогою UML діаграми станів, становить невід'ємну та фундаментальну частину

циклу розробки програмного забезпечення. Діаграми станів у рамках Уніфікованої Мови Моделювання (UML) надають стандартизований та потужний механізм для формального опису динамічної поведінки об'єкта або цілої системи. Вони дозволяють наочно візуалізувати всі можливі дискретні стани, у яких система може перебувати протягом свого життєвого циклу, а також чітко відобразити переходи між цими станами. Кожен стан на діаграмі представляє собою певний, чітко визначений етап у послідовності операцій або станів свідомості бота, як-от "Очікування повного резюме", "Опції доступні" для користувача, або "Очікування частини для оптимізації" після вибору відповідної команди. Переходи між цими станами не є випадковими; вони ініціюються конкретними подіями, які можуть бути як зовнішніми (наприклад, введення користувачем певної команди, такої як `/analyze_resume`, або надсилання тексту), так і внутрішніми діями (наприклад, отримання результатів від API штучного інтелекту). Ця діаграма виконує кілька ключових функцій: вона служить не лише для комплексного документування поточної логіки поведінки бота, але й є потужним інструментом для його проектування на ранніх етапах розробки. Завдяки візуальному представленню, розробники можуть чітко уявити та прослідкувати логічний потік взаємодії користувача з ботом, виявити потенційні архітектурні недоліки, логічні помилки або пропущені переходи, а також гарантувати повне покриття всіх можливих сценаріїв взаємодії з користувачами, забезпечуючи надійність та передбачуваність системи. Використання сучасних інструментів, таких як Mermaid для генерації діаграм безпосередньо з текстового опису в Markdown-форматі, значно спрощує процес моделювання, забезпечує актуальність діаграм та полегшує їх інтеграцію безпосередньо в документацію коду, що є критично важливим для підтримки великих проектів.

Незважаючи на те, що основна бізнес-логіка та взаємодія з Telegram Bot API реалізуються на серверній стороні за допомогою Python, у контексті створення симуляції бота або розробки комплексного веб-інтерфейсу для моніторингу, управління чи детального перегляду даних, може бути залучений React. React є потужною та декларативною бібліотекою JavaScript, розробленою для ефективної

побудови динамічних, високоінтерактивних та чуйних користувацьких інтерфейсів. Він дозволяє розробникам створювати складні інтерфейси шляхом композиції незалежних, повторно використовуваних компонентів, кожен з яких відповідає за рендеринг певної частини інтерфейсу. Основна перевага React полягає у його ефективному механізмі оновлення інтерфейсу: React використовує віртуальний DOM, який дозволяє мінімізувати прямі маніпуляції з реальним DOM, що є "дорогим" з точки зору продуктивності. Коли відбуваються зміни у стані даних, React спочатку порівнює віртуальний DOM з його попередньою версією, обчислює мінімальні зміни, а потім ефективно оновлює лише необхідні частини реального DOM, забезпечуючи плавну та високопродуктивну взаємодію з користувачем. У типовій багатоплановій архітектурі, React-додаток (який є frontend-частиною) взаємодіє з Python-сервером (backend) за допомогою стандартизованих протоколів, таких як RESTful API або WebSockets. RESTful API дозволяє клієнту (React-додатку) робити HTTP-запити (GET, POST, PUT, DELETE) до визначених ендпоінтів на сервері для отримання, створення, оновлення або видалення даних. WebSockets, своєю чергою, забезпечують двонаправлене, постійне з'єднання між клієнтом і сервером, що є ідеальним для реалізації функцій реального часу, таких як оновлення стану діалогу або отримання миттєвих повідомлень. Таке архітектурне розділення відповідальності між представленням даних (frontend) та бізнес-логікою (backend) забезпечує ряд переваг: воно дозволяє незалежну розробку та розгортання обох частин системи, спрощує масштабування кожного шару окремо, підвищує ремонтпридатність та покращує загальну організацію коду. У випадку симуляції, як це було представлено, React імітує інтерфейс та поведінку Telegram-бота, візуалізуючи діалог та миттєві відгуки. Ця симуляція служить цінним інструментом для розробників, дозволяючи їм тестувати, відлагоджувати та вдосконалювати логіку бота та його користувацький досвід без необхідності розгортання повноцінного Telegram-бота в реальному часі.

Нарешті, для забезпечення стабільного, безперебійного та масштабованого функціонування будь-якого інтелектуального бота, належне розгортання та хостинг є абсолютно незамінними. Серверний додаток, реалізований на Python, повинен

бути розміщений на надійній інфраструктурній платформі, яка гарантує безперервну доступність, високу продуктивність та здатність масштабуватися відповідно до зростаючого навантаження. Сучасні хмарні платформи, такі як Google Cloud Platform (GCP), Amazon Web Services (AWS) або Microsoft Azure, пропонують широкий спектр сервісів, що задовольняють ці вимоги. Ці платформи надають не лише базову інфраструктуру (віртуальні машини, обчислювальні інстанси), але й високооптимізовані та керовані сервіси, такі як контейнерні платформи (наприклад, Kubernetes Engine на GCP або Amazon EKS на AWS), безсерверні обчислення (Cloud Functions або Lambda), а також керовані бази даних (Cloud SQL, Cloud Spanner, DocumentDB), необхідні для ефективного розгортання та управління додатком протягом усього його життєвого циклу. Використання технологій контейнеризації, зокрема за допомогою Docker, дозволяє "упакувати" додаток та всі його залежності в ізольовані, портативні контейнери. Це стандартизує середовище розгортання, забезпечуючи, що додаток працюватиме однаково в будь-якому середовищі – від локальної розробки до виробничих серверів. Технології оркестрації контейнерів, такі як Kubernetes, дозволяють автоматизувати розгортання, масштабування, управління та самовідновлення контейнеризованих додатків, що є критично важливим для великих систем. Без надійної, гнучкої та масштабованої інфраструктури, навіть найдосконаліший та найрозумніший бот не зможе забезпечити безперебійний сервіс для своєї аудиторії, що може призвести до негативного користувацького досвіду та втрати функціональності. Таким чином, вибір та налаштування хостингу є завершальним, але не менш важливим етапом у створенні повноцінної та життєздатної інтелектуальної системи.

Отже, розробка інтелектуального чат-бота для аналізу та оптимізації резюме є багатошаровим, міждисциплінарним процесом, що спирається на глибоке розуміння та вміле поєднання різноманітних технологій та методологій. Від міцної та гнучкої серверної основи на Python, яка ефективно взаємодіє з Telegram API для обробки запитів та управління діалогом, до передових можливостей штучного інтелекту, реалізованих через Великі Мовні Моделі, що забезпечують складне

розуміння, генерацію та маніпуляцію природною мовою, від надійних систем управління базами даних, що гарантують персистентне збереження та доступність інформації користувачів, до інструментів візуального моделювання, таких як UML діаграми станів, що формалізують та візуалізують поведінку системи, а також до технологій розгортання та хостингу, що забезпечують безперебійну роботу – кожен з цих елементів є критично важливим для створення ефективного, надійного та корисного інструменту для професійного розвитку. Цей комплексний підхід свідчить про те, як сучасні технології, працюючи в синергії та доповнюючи одна одну, можуть створювати інноваційні, масштабовані та потужні рішення для задоволення реальних потреб користувачів у сучасному цифровому світі.

Це дослідження все ж має деякі обмеження. По-перше, це дослідження фокусується на сценаріях скринінгу резюме і не вивчає інші сценарії управління людськими ресурсами для узагальнення результатів цього дослідження. Тому майбутні дослідження можуть вивчити більше сценаріїв у сфері управління людськими ресурсами для підтвердження висновків цього дослідження. Зокрема, хоча управління людськими ресурсами охоплює багато різних сценаріїв, між ними є деякі спільні риси. Наприклад, будь то підбір персоналу, навчання або оцінка продуктивності, необхідно збирати, аналізувати та використовувати інформацію про персонал. Тому важливо уточнити модель і метод цього дослідження, виявити характеристики та потреби різних сценаріїв управління людськими ресурсами, провести дослідження крос-сценарного застосування і, нарешті, поширити результати досліджень на більш широкий спектр сценаріїв. Беручи за приклад сценарій оцінювання продуктивності, модель у цьому дослідженні включає двох оцінювачів, людину та штучний інтелект. Різні результати оцінювання та досвід штучного інтелекту, що використовується в процесі оцінювання, можуть взаємодіяти з оцінювачем, впливаючи на сприйняття працівниками справедливості. Виходячи з вищевикладеного, для верифікації висновків, зроблених у цьому дослідженні, розроблені відповідні сценарні експерименти, які зможуть популяризувати модель.

По-друге, у цьому дослідженні окремо розглядалися модеруючі ефекти сприятливості результатів та експертність штучного інтелекту, але ефекти інтеграції двох змінних не досліджувалися. Таким чином, для підвищення цілісності дослідження та узагальнення результатів, майбутні дослідження також можуть збільшити експеримент, включивши всі змінні в модель. По-третє, майбутні рішення можуть бути спрямовані на співпрацю людини та штучного інтелекту для управління людськими ресурсами, про що це дослідження не заглиблюється. Люди, які співпрацюють зі штучним інтелектом для прийняття рішень щодо управління персоналом, не тільки скористаються можливостями штучного інтелекту з обробки інформації, але й зможуть подбати про деталі, які раніше могли бути пропущені під наглядом людини, що потенційно може призвести до більш збалансованих, справедливих і контекстуально врахованих результатів прийняття рішень. Таким чином, у майбутніх дослідженнях можна порівняти прийняття рішень людиною, прийняття рішень за допомогою штучного інтелекту та спільне прийняття рішень людиною та штучним інтелектом, щоб розширити модель дослідження в цьому дослідженні.

ВИСНОВКИ

Штучний інтелект (ШІ) продовжує швидко розвиватися та значно підвищує ефективність обробки інформації, багато компаній почали використовувати штучний інтелект у своїх УЛР-процесах. Оскільки залишається незрозумілим, як співробітники реагують на рішення, прийняті штучним інтелектом, це дослідження досліджує вплив різних перевіряючих резюме (людей та штучного інтелекту) на сприйняття заявниками процедурної та розподільчої справедливості на основі сценарію перевірки резюме. Ця робота також включає дослідження сприятливості результатів і експертності штучного інтелекту за допомогою двох сценарних експериментів. Дослідження зосереджене на ситуаціях, коли ШІ використовується в завданнях з управління людськими ресурсами. Випадок відбору резюме показує, як рішення, прийняті штучним інтелектом, впливають на процедурну та розподільчу справедливість претендентів. По-друге, в той час як попередні дослідження в основному були зосереджені на вивченні справедливості в загальному сенсі, це дослідження розділяє справедливість на два виміри (процедурну справедливість і розподільчу справедливість) і виявляє, що заміна людей на ШІ при відборі резюме мала негативні наслідки для обох суджень про справедливість. Крім того, це дослідження демонструє відносно повну модель, поєднуючи особистість особи, яка приймає рішення, результати прийняття рішень та особливості осіб, які приймають рішення. Підсумовуючи, це дослідження забезпечує глибше розуміння значення справедливості, а також уточнює попередні спірні дослідження щодо штучного інтелекту та справедливості та наголошує, що результати рішень не слід нехтувати, зосереджуючись на справедливості процесу. Крім того, це дослідження надає ідеї щодо вибору змінних та експериментального дизайну для подальших досліджень, пов'язаних із рішеннями, що приймаються штучним інтелектом.

В рамках бакалаврської роботи був розроблений та впроваджений веб-додаток для генерації тематичних повідомлень з використанням генеративних моделей штучного інтелекту. Ідея проекту полягала у створенні чат-бота-консультанта для веб-додатку-бібліотеки, який здатен тренуватись на вхідний

інформації та дотримуватися впроваджених інструкцій, що значно допоможе підвищити ефективність комунікацій між клієнтами та компаніями.

Як результат, функціонал бібліотеки дозволяє продивлятися, створювати, редагувати та видаляти книги, клієнтів, видавців та замовлення, тоді як чат-бот-консультант здатен відповідати на будь-які запити користувачів, стосовні наявних книг або надавати загальну інформацію о компанії.

На початковому етапі було проаналізовано предметну область, а саме: вивчено поняття інтелектуальних мовних моделей та проведено огляд популярних веб-сайтів з інтегрованими чат-ботами на основі штучного інтелекту. Наступним кроком стало обрання засобів реалізації проекту: огляд існуючих засобів створення веб-додатку та інтеграції чат-бота-консультанта.

Розроблений веб-додаток успішно пройшов тестування і повністю задовольняє поставлені вимоги. Він готовий до впровадження та використання в реальних умовах.

Завдяки виконанню цієї роботи було отримано цінні знання та практичний досвід у сферах розробки веб-додатків, інтеграції інтелектуальних моделей, а також проектування зручних користувацьких інтерфейсів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Jainh, M. A Comparison of Nationalized and Private Banks' Strategic Human Resource Management Practices. *J. Contemp. Issues Bus. Gov.* 2021, 27, 709–713.
2. Stone, D.L.; Deadrick, D.L.; Lukaszewski, K.M.; Johnson, R. The Influence of Technology on the Future of Human Resource Management. *Hum. Resour. Manag. Rev.* 2015, 25, 216–231.
3. Marler, J.H.; Parry, E. Human resource management, strategic involvement and e-HRM technology. *Int. J. Hum. Resour. Manag.* 2016, 27, 2233–2253.
4. García-Arroyo, J.; Osca Segovia, A. Big data contributions to human resource management: A systematic review. *Int. J. Hum. Resour. Manag.* 2021, 32, 4337–4362.
5. Hong, Z. Research on human resource recommendation algorithm based on machine learning. *Sci. Program.* 2021, 2021, 8387277.
6. Michael, H.; Kaplan, A. A Brief History of Artificial Intelligence: On the Past, Present, and Future of Artificial Intelligence. *Calif. Manag. Rev.* 2019, 61, 5–14.
7. Tambe, P.; Cappelli, P.; Yakubovich, V. Artificial intelligence in human resources management: Challenges and a path forward. *Calif. Manag. Rev.* 2019, 61, 15–42.
8. Tekkesin, A.I. Artificial Intelligence in Healthcare: Past, Present and Future. *Anatol. J. Cardiol.* 2019, 22, 8–9.
9. Buiten, M.C. Towards intelligent regulation of artificial intelligence. *Eur. J. Risk Regul.* 2019, 10, 41–59.
10. Kuzior, A. Technological Unemployment in The Perspective of Industry 4.0 Development. *Virtual Econ.* 2022, 5, 7–23.
11. Kuzior, A.; Kwilinski, A. Cognitive Technologies and Artificial Intelligence in Social Perception. *Manag. Syst. Prod. Eng.* 2022, 30, 109–115.
12. Berhil, S.; Benlahmar, H.; Labani, N. A review paper on artificial intelligence at the service of human resources management. *Indones. J. Electr. Eng. Comput. Sci.* 2020, 18, 32–40. [CrossRef]

13. Bhardwaj, G.; Singh, S.V.; Kumar, V. An Empirical Study of Artificial Intelligence and its Impact on Human Resource Functions. In Proceedings of the 2020 International Conference on Computation, Automation and Knowledge Management (ICCAKM), Dubai, United Arab Emirates, 9–10 January 2020; pp. 47–51.
14. Ore, O.; Sposato, M. Opportunities and risks of artificial intelligence in recruitment and selection. *Int. J. Organ. Anal.* 2021, 30, 1771–1782.
15. Li, F.; Ruijs, N.; Lu, Y. Ethics & AI: A Systematic Review on Ethical Concerns and Related Strategies for Designing with AI in Healthcare. *AI* 2023, 4, 28–53.
16. Jepsen, D.M.; Grob, S. Sustainability in recruitment and selection: Building a framework of practices. *J. Educ. Sustain. Dev.* 2015, 9, 160–178.
17. Bauer, T.N.; Erdogan, B.; Taylor, S. Creating and Maintaining Environmentally Sustainable Organizations: Recruitment and Onboarding; Jossey-Bass/Wiley: New York, NY, USA, 2012.
18. Saad, M.F.M.; Nugro, A.W.L.; Thinakaran, R.; Baijed, M. A Review of Artificial Intelligence Based Platform in Human Resource Recruitment Process. In Proceedings of the 6th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE), Kedah, Malaysia, 1–3 December 2021. *Sustainability* 2023, 15, 2737 14 of 15
19. Furtmuellera, E.; Wilderom, C.; Tate, M. Managing recruitment and selection in the digital age: E-HRM and resumes. *Hum. Syst. Manag.* 2021, 30, 243–259.
20. Wilfred, D. AI in recruitment. *NHRD Netw. J.* 2018, 11, 15–18.
21. Da Silva, L.B.P.; Soltovski, R.; Pontes, J.; Treinta, F.T.; Leitão, P.; Mosconi, E.; Yoshino, R.T. Human resources management 4.0: Literature review and trends. *Comput. Ind. Eng.* 2022, 168, 108111. [CrossRef]
22. Merlin, R.P.; Jayam, R. Artificial Intelligence in Human Resource Management. *Int. J. Pure Appl. Math.* 2018, 119, 1891–1896.
23. Alavuo, N.H. Modern Recruitment Process as a Competitive Advantage in Talent Acquisition: A Recruiter's Playbook. Master's Thesis, Haaga-Helia University of Applied Sciences, Helsinki, Finland, 2020.

24. Hoang, L. From Customer Journey to Candidate Journey: Applying Marketing Principles to Build a Winning Hiring Culture. Master's Thesis, Metropolia University of Applied Sciences, Helsinki, Finland, 2018.
25. Al-Alawi, A.I.; Naureen, M.; AlAlawi, E.I.; Al-Hadad, A.A.N. The Role of Artificial Intelligence in Recruitment Process DecisionMaking. In Proceedings of the 2021 International Conference on Decision Aid Sciences and Application (DASA), Sakheer, Bahrain, 7–8 December 2021; pp. 197–203.
26. Utami, E.; Luthfi, E.T. Profiling Analysis Based on Social Media for Prospective Employees Recruitment Using SVM and Chi-Square. J. Phys. Conf. Ser. 2018, 1140, 012043.
27. Wang, Q.L.; Li, B.; Hu, J. Feature selection for human resource selection based on affinity propagation and SVM sensitivity analysis. In Proceedings of the 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), Coimbatore, India, 9–11 December 2009.
28. Majumder, S.; Mondal, A. Are chatbots really useful for human resource management? Int. J. Speech Technol. 2021, 24, 969–977. [CrossRef]
29. Nawaz, N.; Gomes, A.M. Artificial intelligence chatbots are new recruiters. (IJACSA) Int. J. Adv. Comput. Sci. Appl. 2019, 10. [CrossRef]
30. Anitha, K.; Shanthi, V. A Study on Intervention of Chatbots in Recruitment. In Innovations in Information and Communication Technologies (IICT-2020); Springer: Cham, Switzerland, 2021; pp. 67–74.
31. Koivunen, S.; Ala-Luopa, S.; Olsson, T.; Haapakorpi, A. The March of Chatbots into Recruitment: Recruiters' Experiences, Expectations, and Design Opportunities. Comput. Support. Coop. Work. (CSCW) 2022, 31, 1–30. [CrossRef]
32. Shehu, M.A.; Saeed, F. An adaptive personnel selection model for recruitment using domain-driven data mining. J. Theor. Appl. Inf. Technol. 2016, 91, 117–130.
33. Singh, S.; Kumar, V. Performance Analysis of Engineering Students for Recruitment Using Classification Data Mining Techniques. Int. J. Sci. Eng. Comput. Technol. 2013, 3, 31–37.

34. Kumar, V.; Minz, S. Feature Selection: A literature Review. SmartCR 2014, 4, 211–229.
35. Chandrashekar, G.; Sahin, F. A survey on feature selection methods. Comput. Electr. Eng. 2014, 40, 16–28.
36. Django introduction - Learn web development | MDN. MDN Web Docs. URL: <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction> (date of access: 30.05.2024).
37. Joshi B. Beginning JQuery 2 for ASP. NET Developers: Using JQuery 2 with ASP. NET Web Forms and ASP. NET MVC. Springer, 2014.
38. Паздрій Ігор. Порівняльний аналіз систем розробки програмного забезпечення на основі фреймворків. Вісник Хмельницького національного університету. 2023. 155 с. URL: <http://journals.khnu.km.ua/vestnik/wp-content/uploads/2023/03/vknu-ts-2023-n1317-155-161.pdf>.
39. What is Flask Python - Python Tutorial. Learn Python Programming - Python Tutorial. URL: <https://pythonbasics.org/what-is-flask-python/> (date of access: 30.05.2024).

ДОДАТОК А

Технічне завдання

Міністерство освіти і науки України

Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії

Кафедра обчислювальної техніки

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ

проф., д.т.н.. Азаров О.Д.

«__» 2025 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання бакалаврської кваліфікаційної роботи

«Система динамічної емуляції x64 коду в середовищі Windows»

08-54.БКР.016.00.000 ТЗ

Науковий керівник: доцент к.т.н.

Кожем'яко А. В. _____

Студент групи 2КІ-216

Смаковський Б.А. _____

1 Підстави для виконання бакалаврської кваліфікаційної роботи (БКР)

1.1 Актуальність дослідження обумовлена зростаючою потребою в ефективних засобах аналізу виконання програмного коду в середовищі Windows, зокрема у контексті забезпечення інформаційної безпеки

1.2 Тема бакалаврської кваліфікаційної роботи затверджена наказом ректора університету №97 від 20.03.2025 року.

2 МетаБКР і призначення розробки

2.1 Метою цієї роботи є підвищення ефективності комунікацій між компаніями та клієнтами шляхом створення веб-додатків для генерації та аналізу резюме завдяки впровадженню чат-бота-консультанта на основі штучного інтелекту

2.2 Призначення розробки полягає у підвищенні ефективності комунікацій між компаніями та клієнтами завдяки впровадженню чат-бота-консультанта на основі штучного інтелекту.

3 Вихідні дані для виконанняБКР

3.1 Проведення аналізу сучасних методів динамічної генерації коду резюме, а також огляд інструментів і бібліотек, що використовуються у сфері створення засобів із штучним інтелектом.

3.2 Розробка загальної архітектури системи внутрішньопроектної динамічної емуляції, визначення основних модулів та принципів їх взаємодії.

3.3 Створення спеціалізованих тестових бібліотек та утиліт для перевірки правильності емуляції, трасування виконання та відлагодження основних сценаріїв роботи системи.

4 Вимоги до виконанняБКР

Головна вимога – проаналізувати предметну область, оглянути поняття

штучного інтелекту; провести огляд мови програмування та інших технологій для реалізації візуальної частини веб-додатку; оглянути наявні технології створення чат-бота-консультанта; спроектувати та реалізувати веб-додаток з використанням чат-бота-консультанта; розробити веб-додаток бібліотеку з інтегрованим чат-ботом консультантом.

5 Етапи БКР та очікувані результати

Оформлення пояснювальної записки, графічного матеріалу, формулювання висновків та узагальнення отриманих результатів. Очікувані результати: завершений звіт з описом реалізації, підтвердженою працездатністю та аналізом досягнутих цілей.

6 Матеріали, що подаються до захисту БКР

До захисту подаються: пояснювальна записка БКР, графічні та ілюстративні матеріали, протокол попереднього захисту на кафедрі, відгуки наукового керівника та опонента, протоколи складання державних екзаменів, анотації до БКР українською та іноземною мовами, довідка про відповідність оформлення БКР діючим вимогам.

7 Порядок контролю виконання та захисту БКР

Виконання етапів документації БКР контролюється науковим керівником згідно зі встановленими термінами. Захист БКР відбувається на засіданні Екзаменаційної комісії, затвердженої наказом ректора.

8 Вимоги до оформлювання та порядок виконання БКР

8.1 Вимоги до оформлювання

При оформлюванні БКР використовуються:

– ДСТУ 3008: 2015 «Звіти в сфері науки і техніки. Структура та правила

оформлювання»;

– ДСТУ 8302: 2015 «Бібліографічні посилання. Загальні положення та правила складання»;

– міждержавний ГОСТ 2.104-2006 «Єдина система конструкторської документації. Основні написи»;

– методичні вказівки до виконання бакалаврських кваліфікаційних робіт зі спеціальності 123 – «Комп’ютерна інженерія» (освітня програма «Системне програмування»). Кафедра обчислювальної техніки ВНТУ 2023.

8.2 Порядок виконання БКР викладено в «Положення про кваліфікаційні роботи на першому (бакалаврському) рівні вищої освіти СУЯ ВНТУ-03.02.02-П.001.01:21».

ДОДАТОК Б

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: Веб-додаток для генерації та аналізу резюме із застосуванням штучного інтелекту

Тип роботи: бакалаврська кваліфікаційна робота
(бакалаврська кваліфікаційна робота / магістерська кваліфікаційна робота)

Підрозділ кафедра обчислювальної техніки
(кафедра, факультет, навчальна група)

Коефіцієнт подібності текстових запозичень, виявлених у роботі системою StrikePlagiarism 17 %

Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне)

- ☒ Запозичення, виявлені у роботі, є законними і не містять ознак плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту
- ☐ У роботі не виявлено ознак плагіату, фабрикації, фальсифікації, але надмірна кількість текстових запозичень та/або наявність типових розрахунків не дозволяють прийняти рішення про оригінальність та самостійність її виконання. Роботу направити на доопрацювання.
- ☐ У роботі виявлено ознаки плагіату та/або текстових маніпуляцій як спроб укриття плагіату, фабрикації, фальсифікації, що суперечить вимогам законодавства та нормам академічної доброчесності. Робота до захисту не приймається.

Експертна комісія:

<u>Азаров О.Д., зав. каф. ОТ</u>	_____
(прізвище, ініціали, посада)	(підпис)
<u>Крупельницький Л.В., доц. каф ОТ</u>	_____
(прізвище, ініціали, посада)	(підпис)

Особа, відповідальна за перевірку <u>Захарченко С.М.</u>	_____
(прізвище, ініціали)	(підпис)

З висновком експертної комісії ознайомлений(-на)

Керівник _____	<u>Кожем'яко А.В. доц. каф ОТ</u>
(підпис)	(прізвище, ініціали, посада)
Здобувач _____	<u>Смаковський Б.А.</u>
(підпис)	(прізвище, ініціали)

ДОДАТОК В
Графічна частина

ДІАГРАМА КЛАСІВ FLASK

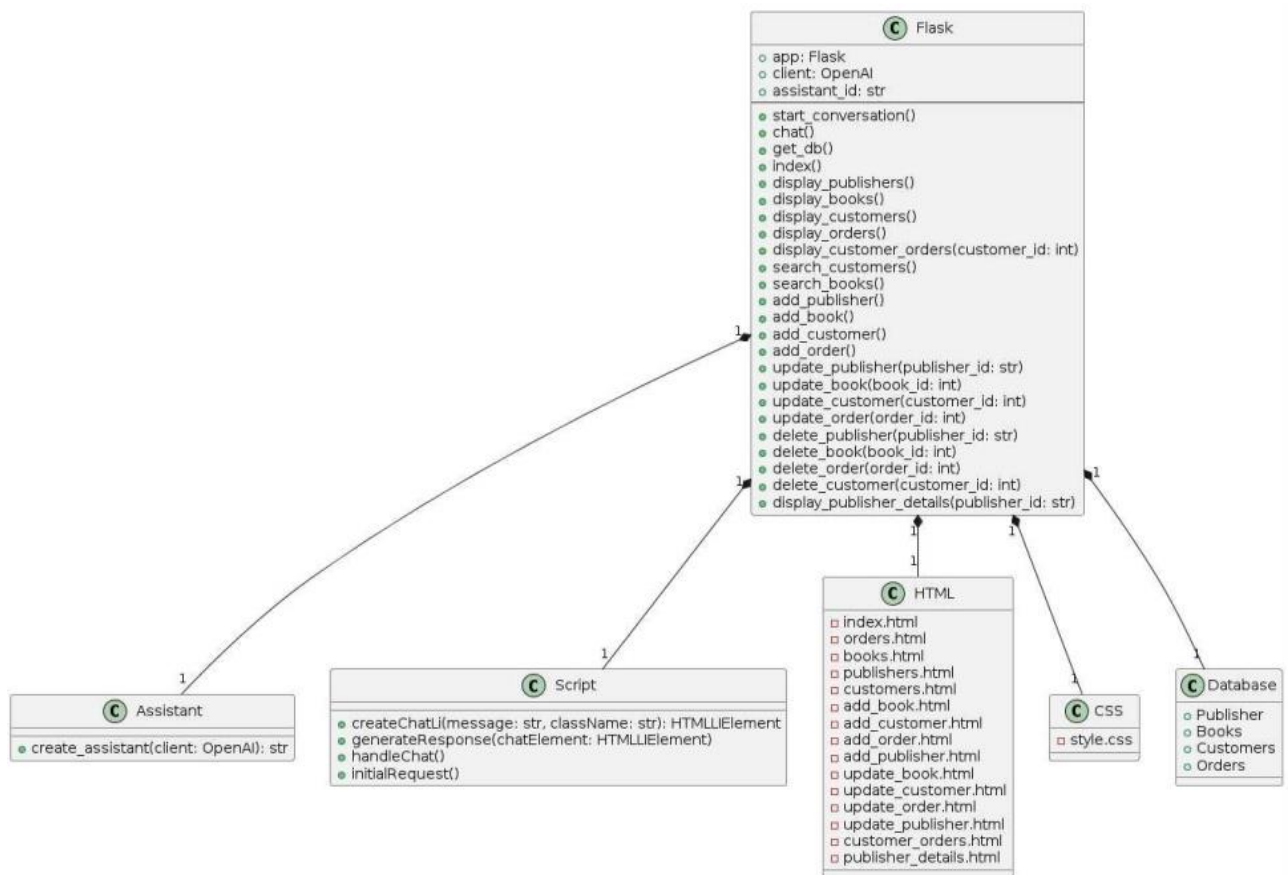


Рисунок В.1 — Діаграма класів

ДОДАТОК Г

Ілюстративна частина

**ВЕБ-ДОДАТОК ДЛЯ ГЕНЕРАЦІЇ ТА АНАЛІЗУ РЕЗЮМЕ ІЗ
ЗАСТОСУВАННЯМ ШТУЧНОГО ІНТЕЛЕКТУ**

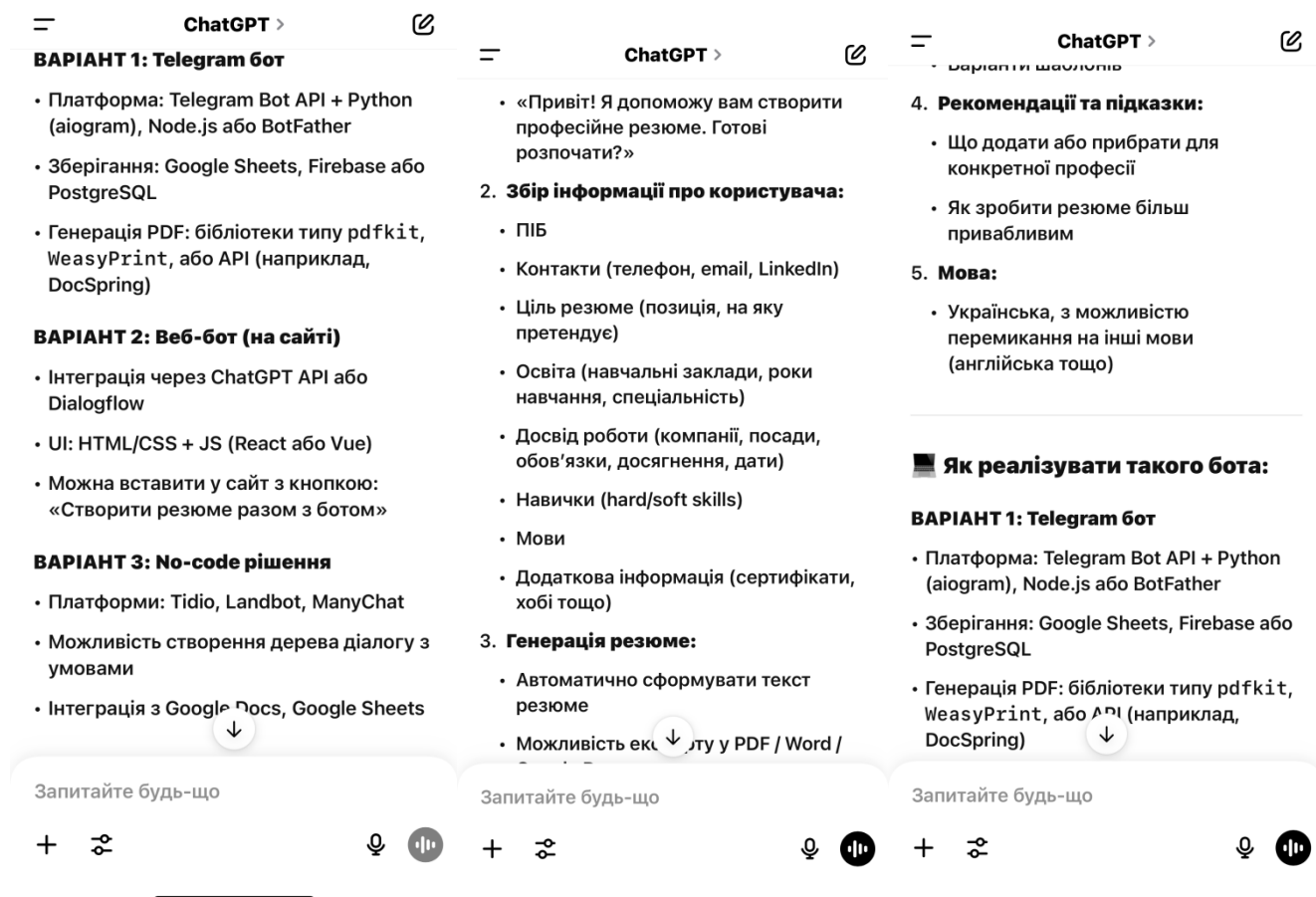


Рисунок Г.1 — Чат-бот консультант

ДОДАТОК Д

Лістинг програмного забезпечення

```
import React, { useState, useEffect, useRef } from 'react';
import { Send, User, Bot, Sparkles, FileText, CheckCircle, XCircle } from 'lucide-react'; // Icons

// Main App Component
const App = () => {
  const [messages, setMessages] = useState([]);
  const [input, setInput] = useState("");
  const [currentStep, setCurrentStep] = useState('start'); // State for bot's
conversation flow
  const [resumeData, setResumeData] = useState({}); // Stores collected resume
data
  const messagesEndRef = useRef(null);
  const [isLoading, setIsLoading] = useState(false);

  // Scroll to the latest message
  const scrollToBottom = () => {
    messagesEndRef.current?.scrollIntoView({ behavior: "smooth" });
  };

  useEffect(() => {
    scrollToBottom();
  }, [messages]);

  // Initial bot message on load
  useEffect(() => {
    addBotMessage("Привіт! Я твій помічник у створенні резюме. Напиши
/start, щоб почати.");
  }, []);

  // Function to add a user message
  const addUserMessage = (text) => {
    setMessages((prevMessages) => [...prevMessages, { type: 'user', text }]);
  };

  // Function to add a bot message
  const addBotMessage = (text) => {
    setMessages((prevMessages) => [...prevMessages, { type: 'bot', text }]);
  };
}
```

```

    };

    // Simulate AI API call
    const callGeminiAPI = async (prompt, schema = null) => {
      setIsLoading(true);
      // Simulate network delay
      await new Promise(resolve => setTimeout(Math.random() * 1500 + 500,
        resolve));

      let simulatedResponse = "";
      if (prompt.includes("Створи професійний драфт резюме")) {
        simulatedResponse = `
        **Резюме: ${resumeData.fullName} || 'Ваше Ім'я'**

        **Контакти:**
        ${resumeData.contactInfo} || 'Не вказано'}

        **Освіта:**
        ${resumeData.education} || 'Не вказано'}

        **Досвід роботи:**
        ${resumeData.workExperience} || 'Не вказано'}

        **Навички:**
        ${resumeData.skills} || 'Не вказано'}

        *Це драфт. Ви можете його оптимізувати або проаналізувати.*
        `;
      } else if (prompt.includes("Оптимізуй наступний текст")) {
        simulatedResponse = `
        *Оптимізований текст:*
        "${prompt.split("Текст: ")[1].split(" ")[0].trim()}"

        Цей текст тепер звучить більш професійно та лаконічно, підкреслюючи ваші
        досягнення.
        `;
      } else if (prompt.includes("Проаналізуй наступне резюме")) {
        simulatedResponse = `
        *Аналіз резюме:*

        Ваше резюме виглядає досить повним. Рекомендую додати конкретні цифри
        та метрики до опису досвіду роботи, щоб продемонструвати ваші досягнення.
        Також переконайтеся, що навички відповідають вимогам вакансій, на які ви
        подаєтесь.

```

```

        `;
    } else {
        simulatedResponse = "Я не розумію цього запиту. Спробуйте іншу
команду.";
    }

    setIsLoading(false);
    return {
        candidates: [{
            content: {
                parts: [{ text: simulatedResponse }]
            }
        }]
    };
};

// Handle user input
const handleSendMessage = async () => {
    if (input.trim() === "") return;

    addUserMessage(input);
    const lowerCaseInput = input.toLowerCase().trim();
    setInput(""); // Clear input field

    switch (currentStep) {
        case 'start':
            if (lowerCaseInput === '/start') {
                addBotMessage("Привіт! Я допоможу тобі створити резюме.
Напиши /create_resume, щоб почати.");
                setCurrentStep('waiting_for_create_resume');
            } else {
                addBotMessage("Будь ласка, напишіть /start, щоб почати.");
            }
            break;

        case 'waiting_for_create_resume':
            if (lowerCaseInput === '/create_resume') {
                setResumeData({}); // Reset data for new resume
                addBotMessage("Добре, почнемо створення резюме. Введіть ваше
повне ім'я (наприклад, 'Іван Петров'):");
                setCurrentStep('ask_full_name');
            } else {

```

```

        addBotMessage("Будь ласка, напишіть /create_resume, щоб почати створення резюме.");
    }
    break;

    case 'ask_full_name':
        setResumeData(prev => ({ ...prev, fullName: input }));
        addBotMessage("Дякую! Тепер введіть вашу контактну інформацію (телефон, email, посилання на LinkedIn):");
        setCurrentStep('ask_contact_info');
        break;

    case 'ask_contact_info':
        setResumeData(prev => ({ ...prev, contactInfo: input }));
        addBotMessage("Будь ласка, введіть інформацію про вашу освіту (навчальний заклад, спеціальність, роки навчання):");
        setCurrentStep('ask_education');
        break;

    case 'ask_education':
        setResumeData(prev => ({ ...prev, education: input }));
        addBotMessage("Опишіть ваш досвід роботи (компанія, посада, роки, основні обов'язки та досягнення):");
        setCurrentStep('ask_work_experience');
        break;

    case 'ask_work_experience':
        setResumeData(prev => ({ ...prev, workExperience: input }));
        addBotMessage("Перелічіть ваші ключові навички (через кому, наприклад: Python, JavaScript, Project Management):");
        setCurrentStep('ask_skills');
        break;

    case 'ask_skills':
        setResumeData(prev => ({ ...prev, skills: input }));
        addBotMessage("Дякую! Я отримав всю інформацію. Генерую чернетку вашого резюме за допомогою ШІ...");
        setCurrentStep('generating_resume');

    const promptGenerate = `Створи професійний чернетку резюме на основі наступних даних:
        Ім'я та прізвище: ${resumeData.fullName}
        Контактна інформація: ${resumeData.contactInfo}

```

Освіта: \${resumeData.education}
 Досвід роботи: \${resumeData.workExperience}
 Навички: \${input}
 Зроби його лаконічним, але інформативним, підкресливши ключові моменти.`;

```
const generatedResult = await callGeminiAPI(promptGenerate);
if (generatedResult.candidates && generatedResult.candidates.length > 0) {
  const generatedText =
    generatedResult.candidates[0].content.parts[0].text;
    addBotMessage(generatedText);
    addBotMessage("Ви можете оптимізувати текст, написавши /optimize_text, або проаналізувати резюме, написавши /analyze_resume.");
    setCurrentStep('resume_generated');
  } else {
    addBotMessage("Виникла помилка при генерації резюме. Спробуйте ще раз.");
    setCurrentStep('waiting_for_create_resume'); // Allow retry
  }
  break;
```

```
case 'resume_generated':
  if (lowerCaseInput === '/optimize_text') {
    addBotMessage("Яку частину резюме ви хочете оптимізувати? (Наприклад: 'досвід роботи', 'навички', 'освіта')");
    setCurrentStep('ask_optimize_part');
  } else if (lowerCaseInput === '/analyze_resume') {
    addBotMessage("Аналізую ваше резюме...");
    const promptAnalyze = `Проаналізуй наступне резюме та надай рекомендації щодо його покращення.
    Зокрема, зверни увагу на:
    - Повноту інформації
    - Використання ключових слів
    - Опис досягнень замість обов'язків
    - Граматичні та стилістичні помилки (якщо помітні)
    - Загальну структуру та читабельність.
```

Резюме:

Ім'я: \${resumeData.fullName}
 Контакти: \${resumeData.contactInfo}
 Освіта: \${resumeData.education}
 Досвід: \${resumeData.workExperience}


```

        Навички: ${resumeData.skills}`;

        const analysisResult = await callGeminiAPI(promptAnalyze);
        if (analysisResult.candidates && analysisResult.candidates.length >
0) {
            addBotMessage(analysisResult.candidates[0].content.parts[0].text);
            addBotMessage("Ви можете оптимізувати текст, написавши
/optimize_text, або завершити, написавши /done.");
        } else {
            addBotMessage("Виникла помилка при аналізі резюме.
Спробуйте ще раз.");
        }
        } else if (lowerCaseInput === '/done') {
            addBotMessage("Дякую за використання бота! Ваше резюме готове.
Ви можете почати знову, написавши /start.");
            setCurrentStep('start');
        } else {
            addBotMessage("Будь ласка, виберіть опцію: /optimize_text,
/analyze_resume або /done.");
        }
        break;

    case 'ask_optimize_part':
        let textToOptimize = "";
        let fieldToUpdate = "";
        if (lowerCaseInput.includes('досвід роботи')) {
            textToOptimize = resumeData.workExperience;
            fieldToUpdate = 'workExperience';
        } else if (lowerCaseInput.includes('навички')) {
            textToOptimize = resumeData.skills;
            fieldToUpdate = 'skills';
        } else if (lowerCaseInput.includes('освіта')) {
            textToOptimize = resumeData.education;
            fieldToUpdate = 'education';
        } else {
            addBotMessage("Невідома частина резюме. Будь ласка, вкажіть
'досвід роботи', 'навички' або 'освіта'.");
            break;
        }

        if (textToOptimize) {
            addBotMessage(`Оптимізую "${lowerCaseInput}"...`);

```

```

        const promptOptimize = `Оптимізуй наступний текст для резюме.
Зроби його більш професійним, лаконічним та impactful.
Текст: "${textToOptimize}"
        Якщо це опис досвіду роботи, сфокусуйся на досягненнях,
використовуючи дієслова дії.`;

        const optimizedResult = await callGeminiAPI(promptOptimize);
        if (optimizedResult.candidates && optimizedResult.candidates.length
> 0) {
                                const    optimizedText    =
optimizedResult.candidates[0].content.parts[0].text;
                addBotMessage(optimizedText);
                // Update resumeData with optimized text
                setResumeData(prev => ({ ...prev, [fieldToUpdate]:
optimizedText.replace('*Оптимізований текст:*\\n', "").replace("\\n\\nЦей текст тепер
звучить більш професійно та лаконічно, підкреслюючи ваші досягнення.", "").trim()
})));
                addBotMessage("Текст оптимізовано. Ви можете оптимізувати
іншу частину, проаналізувати резюме (/analyze_resume) або завершити (/done).");
                } else {
                        addBotMessage("Виникла помилка при оптимізації тексту.
Спробуйте ще раз.");
                }
                } else {
                        addBotMessage("У вас немає даних для цієї частини резюме.");
                }
                setCurrentStep('resume_generated'); // Return to options after
optimization
                break;

        default:
                addBotMessage("Невідома команда. Будь ласка, напишіть /start.");
                setCurrentStep('start');
                break;
        }
};

const handleKeyDown = (e) => {
        if (e.key === 'Enter') {
                handleSendMessage();
        }
};

```

```

return (
  <div className="flex flex-col h-screen bg-gray-100 font-inter p-4 sm:p-6
md:p-8">
    <div className="flex-1 flex flex-col bg-white rounded-lg shadow-lg
overflow-hidden max-w-3xl mx-auto w-full">
      {/* Chat Header */}
      <div className="bg-gradient-to-r from-blue-500 to-purple-600 text-
white p-4 rounded-t-lg flex items-center justify-between shadow-md">
        <h1 className="text-xl font-bold flex items-center">
          <Bot className="mr-2" size={24} /> AI Помічник для Резюме
        </h1>
        <div className="text-sm opacity-80">Симуляція</div>
      </div>

      {/* Messages Area */}
      <div className="flex-1 p-4 overflow-y-auto space-y-4 bg-gray-50">
        {messages.map((msg, index) => (
          <div
            key={index}
            className={`flex ${msg.type === 'user' ? 'justify-end' : 'justify-
start'} `}
            >
              <div
                className={`relative max-w-[80%] p-3 rounded-lg shadow-
sm ${
                  msg.type === 'user'
                    ? 'bg-blue-500 text-white rounded-br-none'
                    : 'bg-gray-200 text-gray-800 rounded-bl-none'
                } `}
                >
                  <div className="flex items-center text-sm font-semibold mb-
1">
                    {msg.type === 'user' ? <User size={16} className="mr-1 "
/> : <Bot size={16} className="mr-1 " />}
                    {msg.type === 'user' ? 'Ви' : 'Бот'}
                  </div>
                  <div className="prose prose-sm max-w-none"
dangerouslySetInnerHTML={{ __html: msg.text.replace(/\n/g, '<br>') }} />
                </div>
              </div>
            >>>
            {isLoading && (
              <div className="flex justify-start">

```

```

<div className="relative max-w-[80%] p-3 rounded-lg shadow-
sm bg-gray-200 text-gray-800 rounded-bl-none">
  <div className="flex items-center text-sm font-semibold mb-
1">
    <Bot size={16} className="mr-1" /> Бот
  </div>
  <div className="flex items-center">
    <span className="dot-pulse"></span>
    <span className="ml-2">Друкую...</span>
  </div>
</div>
</div>
)}
<div ref={messagesEndRef} />
</div>

{/* Input Area */}
<div className="p-4 bg-gray-100 border-t border-gray-200 flex items-
center rounded-b-lg shadow-inner">
  <input
    type="text"
    className="flex-1 p-3 border border-gray-300 rounded-full
focus:outline-none focus:ring-2 focus:ring-blue-500 transition duration-200 ease-in-out
text-gray-800 placeholder-gray-500"
    placeholder="Напишіть повідомлення..."
    value={input}
    onChange={(e) => setInput(e.target.value)}
    onKeyDown={handleKeyDown}
    disabled={isLoading}
  />
  <button
    onClick={handleSendMessage}
    className="ml-3 p-3 bg-blue-600 text-white rounded-full shadow-
lg hover:bg-blue-700 focus:outline-none focus:ring-2 focus:ring-blue-500 focus:ring-
offset-2 transition duration-200 ease-in-out transform hover:scale-105 active:scale-95
disabled:opacity-50 disabled:cursor-not-allowed"
    disabled={isLoading}
  >
    <Send size={24} />
  </button>
</div>
</div>

```

```

{/* Dot pulse animation CSS */}
<style jsx>{`
  .dot-pulse {
    position: relative;
    width: 10px;
    height: 10px;
    border-radius: 50%;
    background-color: #333;
    color: #333;
    animation: dotPulse 1.5s infinite ease-in-out;
  }

  .dot-pulse::before, .dot-pulse::after {
    content: "";
    display: inline-block;
    position: absolute;
    top: 0;
    width: 10px;
    height: 10px;
    border-radius: 50%;
    background-color: #333;
    color: #333;
    animation: dotPulse 1.5s infinite ease-in-out;
  }

  .dot-pulse::before {
    left: -15px;
    animation-delay: 0.2s;
  }

  .dot-pulse::after {
    left: 15px;
    animation-delay: 0.4s;
  }

  @keyframes dotPulse {
    0% {
      transform: scale(0.8);
      opacity: 0.7;
    }
    50% {
      transform: scale(1.2);
      opacity: 1;
    }
  }
`}

```

```
    }  
    100% {  
      transform: scale(0.8);  
      opacity: 0.7;  
    }  
  }  
`}</style>  
</div>  
);  
};  
  
export default App;
```