

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління
(повна назва)

Кафедра комп'ютерних інтелектуальних технологій та систем
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)
Розробка інтелектуальної системи
складання резюме з використанням штучного інтелекту
(тема)

Виконав:
здобувач IV року навчання,
групи КІУКІ-21-10
Михайло МАТВЄЄВ
(власне ім'я, прізвище)

Спеціальність 123 – Комп'ютерна інженерія
(код і повна назва спеціальності)
Тип програми освітньо-професійна
Освітня програма Комп'ютерна Інженерія
(повна назва освітньої програми)

Керівник ас. каф. КІТС Андрій ТАТАРНИКОВ
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри _____ Олег РУДЕНКО
(підпис) (власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерної інженерії та управління _____
Кафедра _____ Комп'ютерних інтелектуальних технологій та систем _____
Рівень вищої освіти _____ перший (бакалаврський) _____
Спеціальність _____ 123 Комп'ютерна інженерія _____
(код і повна назва)
Тип програми _____ освітньо-професійна _____
Освітня програма _____ Комп'ютерна інженерія _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«_____» _____ 20 25 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ Матвєєву Михайлу Сергійовичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Розробка інтелектуальної системи автоматизованого
Складання резюме з використанням штучного інтелекту _____

затверджена наказом по університету від “ 21 ” _____ травня _____ 2025 р. № _____ 399 Ст

2. Термін подання студентом роботи до екзаменаційної комісії _____

3. Вхідні дані до роботи _____

1. Документація мови програмування Python та фреймворку FastAPI.

2. Інструкції з роботи з базами даних MySQL та ORM-бібліотекою SQLAlchemy.

3. Документація HTML5, CSS3, JavaScript.

4. Опис інтеграції локальних мовних моделей через платформу Ollama.

5. Технічна документація середовищ розробки Visual Studio Code та PyCharm.

4. Перелік питань, що потрібно опрацювати в роботі _____

1. Аналіз предметної області.

2. Аналіз використовуваних технологій.

3. Програмна реалізація інтерактивного інтерфейсу та роботи з даними.

4. Інструкція користувача.

5. Висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) _____
Слайд – презентація – 16 слайдів

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Аналіз проблеми та огляд існуючих рішень	26.05.2025-28.05.2025	Виконано
2	Вибір технологій розробки, інструментальних засобів	29.05.2025-31.05.2025	Виконано
3	Розробка підходу для реалізації завдання	01.06.2025-02.06.2025	Виконано
4	Проектування інтелектуальної системи	03.06.2025-07.06.2025	Виконано
5	Розробка інтерфейсу та бази даних системи	08.06.2024-10.06.2025	Виконано
6	Тестування застосунку	11.06.2025-12.06.2025	Виконано
7	Оформлення матеріалів атестаційної роботи	13.06.2025-16.06.2025	Виконано
8	Подання атестаційної роботи керівникові та її попередній захист	17.06.2025-18.06.2025	Виконано
9	Подання роботи на рецензування	19.06.2025-20.06.2025	Виконано

Дата видачі завдання 26 травня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____
(підпис)

ас. Андрій ТАТАРНИКОВ
(посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка атестаційної роботи: 68 с., 22 рис., 1 табл., 2 дод., 31 джерело.

ШТУЧНИЙ ІНТЕЛЕКТ, ГЕНЕРАЦІЯ РЕЗЮМЕ, ВЕБ-ЗАСТОСУНОК, МОВНА МОДЕЛЬ, АВТОМАТИЗАЦІЯ, ІНТЕРФЕЙС КОРИСТУВАЧА, МОДАЛЬНІ ВІКНА, ПОПЕРЕДНІЙ ПЕРЕГЛЯД ДОКУМЕНТА.

Метою атестаційної роботи є розробка інтелектуальної системи автоматизованого складання резюме з використанням штучного інтелекту. Такий програмний застосунок покликаний спростити процес створення резюме для користувачів без технічного досвіду та забезпечити високий рівень персоналізації за рахунок інтеграції сучасних мовних моделей.

У ході виконання атестаційної роботи було проаналізовано існуючі сервіси для складання резюме, визначено їх переваги та недоліки, що дало змогу сформулювати перелік ключових функціональних вимог до розроблюваного рішення. Система реалізована у вигляді веб-застосунку з покроковим введенням персональної інформації через модальні вікна. На основі зібраних даних користувач отримує попередній перегляд резюме, а також має можливість згенерувати його фінальну версію за допомогою обраної мовної моделі штучного інтелекту (GPT, Mistral, Claude тощо).

Користувацький інтерфес розроблено з використанням HTML, CSS, JavaScript, а також бібліотеки jQuery. Для збереження введених даних використовується об'єкт `resumeData`, який формує основу для генерації змісту. Система підтримує збереження готового документа у форматі PDF та експорт інформації у форматі JSON для подальшої обробки або повторного використання.

ABSTRACT

Bachelor's thesis: 68 pages, 22 figures, 1 tables, 2 appendices, 31 sources.

ARTIFICIAL INTELLIGENCE, RESUME GENERATION, WEB APPLICATION, LANGUAGE MODEL, AUTOMATION, USER INTERFACE, MODAL WINDOWS, DOCUMENT PREVIEW.

The major goal of this thesis is to develop an intelligent system for automated resume generation using artificial intelligence technologies. This software application is designed to simplify the process of resume creation for users without technical expertise and to provide a high level of personalization through the integration of modern language models.

In order to define the functional requirements of the developed application, a comparative analysis of existing resume builder platforms was conducted. Based on the identified strengths and weaknesses of these platforms, the system was designed as a step-by-step interface for collecting user information via modal windows. The system generates a structured resume preview and allows the user to produce a finalized version of the resume using a selected AI-powered language model.

The user interface was developed using HTML, CSS, JavaScript, and the jQuery library. The collected data is stored in a structured object called `resumeData`, which serves as the source for generating both the preview and the final resume content. The application supports PDF export and JSON data export for further processing or reuse. The interface is intuitive and adaptable for different languages, making the system accessible to a wide range of users.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	9
ВСТУП.....	10
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	11
1.1 Вплив якісного резюме на процес працевлаштування	11
1.2 Відповідальність користувача при створенні резюме	12
1.3 Використання програмних застосунків для автоматизації створення резюме	13
1.4 Огляд існуючих рішень.....	14
1.4.1 Resumaker	15
1.4.2 Canva	17
1.4.3 Jobseeker	18
1.4.4 Zety	19
1.4.5 Kickresume	20
1.5 Порівняльний аналіз програмних рішень	21
1.6 Постановка задачі.....	22
1.6.1 Системні вимоги	22
1.6.2 Опис функцій програмного забезпечення	23
2 АНАЛІЗ ВИКОРИСТОВУВАНИХ ТЕХНОЛОГІЙ.....	24
2.1 Огляд засобів розробки	24
2.2 Технології реалізації застосунку	25
2.2.1 VS Code в якості редактора коду	25
2.2.2 Javascript.....	26
2.2.3 FastAPI.....	27
2.2.4 PyCharm як середовище розробки	27
2.2.5 Система управління базами даних MySQL.....	28
2.2.6 Авторизація користувачів	29

2.2.7 API для інтеграції з моделлю штучного інтелекту	30
3 ПРОГРАМНА РЕАЛІЗАЦІЯ	31
3.1 Загальна структура веб-застосунку	31
3.2 Обробка введення даних користувача.....	32
3.2.1 Функція savePersonalInfoToData()	32
3.2.2 Функція saveLanguagesToData()	33
3.2.3 Функція saveCoursesToData().....	33
3.2.5 Функція saveEducationToData()	35
3.2.6 Функція saveSkillsToData()	36
3.3 Модальна навігація та управління інтерфейсом.....	37
3.3.1 Функція openModal(id)	37
3.3.2 Функція closeModal(id).....	38
3.3.3 Функція goToNextModal(currentId, nextId).....	38
3.3.4 Функція updateProgressBar(currentModalId)	38
3.4 Формування попереднього перегляду резюме.....	39
3.4.1 Функція renderPreview()	40
3.5 Серверна частина: API, база даних та авторизація.....	40
3.5.1 Файл main.py	41
3.5.2 Файл models.py	42
3.5.3 Файл schemas.py	42
3.5.4 Файл auth.py	42
3.5.5 Файл database.py	43
3.5.6 Основна логіка API.....	43
4 ІНСТРУКЦІЯ КОРИСТУВАЧА.... ERROR! BOOKMARK NOT DEFINED.	
4.1 Реєстрація та вхід до системи.....	44
4.2 Основна панель та навігація	45
4.3 Введення персональних даних.....	45
4.4 Введення інформацію про володіння іноземними мовами	46
4.5 Додавання курсів.....	46
4.6 Введення досвіду роботи	47

4.7 Внесення інформації про освіту	48
4.8 Введення навичок	48
4.9 Формування попереднього вигляду резюме	49
4.10 Вибір ШІ-моделі	49
4.11 Відображення прогресу заповнення	50
4.12 Вибір шаблону резюме	50
4.13 Завершення роботи та отримання готового результату	50
ВИСНОВКИ	53
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	54
ДОДАТОК А ГРАФІЧНИЙ МАТЕРІАЛ АТЕСТАЦІЙНОЇ РОБОТИ	ERROR! BOOKMARK NOT DEFINED.
ДОДАТОК Б СЕРТИФІКАТИ ЗА УЧАСТЬ У НАУКОВИХ КОНФЕРЕНЦІЯХ	ERROR! BOOKMARK NOT DEFINED.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ
І ТЕРМІНІВ

CV – Curriculum Vitae

ШІ – Штучний інтелект

ПК – Персональний комп'ютер

ATS – Applicant Tracking System

PDF – Portable Document Format

UI – User Interface

UX – User Experience

API – Application Programming Interface

СУБД – Система управління базами даних

JWT – JSON Web Token

CRUD – Create, Read, Update, Delete

DOCX – Розширення файлів Microsoft Word (версії з 2007 року),
формат редагованих документів.

PNG – Portable Network Graphics

JPG – Joint Photographic Experts Group

ВСТУП

В умовах сьогодення, де запорукою професійного успіху є конкурентоспроможність, все частіше стає питання необхідності виділитись серед інших кандидатів, які претендують на ту саму посаду. Конкуренція на ринку праці має дуже високий рівень, і ефективність працевлаштування в значній мірі залежить від якості складеного резюме. CV є першим документом, який бачить потенційний роботодавець, і на його основі приймається рішення про подальше ймовірне запрошення кандидата на співбесіду. За статистикою, рекрутери витрачають у середньому від 6 до 10 секунд на первинний огляд CV, тому воно повинно бути лаконічним, добре структурованим та містити ключову інформацію про навички та досвід.

Із розвитком технологій в області штучного інтелекту, з'явилися автоматизовані системи створення резюме, які допомагають користувачам швидко та ефективно генерувати професійно оформлені документи. Використання ШІ у таких системах дозволяє не лише структурувати дані, але й надавати рекомендації щодо покращення тексту, адаптації документа під конкретні вакансії та виправляти граматичні помилки. Це значно підвищує шанси кандидата на отримання бажаної посади.

Під час виконання роботи був виконаний порівняльний аналіз сучасних підходів, які застосовуються для спрощення процесу створення резюме та запропоновано власний підхід до розробки інтелектуальної системи генерації CV. Особлива увага приділяється використанню штучного інтелекту для аналізу змісту, оптимізації структури документа та підвищення його ефективності в процесі працевлаштування.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Вплив якісного резюме на процес працевлаштування

Резюме є важливим документом у процесі працевлаштування, оскільки воно створює перше враження про кандидата ще до його безпосередньої взаємодії з роботодавцем. Правильно складене резюме може суттєво вплинути на рішення кадрового агента, підвищуючи імовірність проходження першого етапу відбору потенційних кандидатів [1].

Одним з прикладів проблем сучасного ринку праці є ситуація в ІТ-сфері, де кожна вакансія може збирати декілька сотень відгуків за день. Ця обставина змушує роботодавців використовувати системи відстеження кандидатів для автоматичного фільтрування резюме або ATS. Такі системи проводять аналіз резюме з урахуванням конкретних критеріїв, включаючи використання ключових слів, відповідність формату і структурованість інформації. Профіль, який не відповідає цим вимогам, може бути автоматично відхиленним, навіть якщо кандидат має відповідні кваліфікації.

Поміж автоматизованих систем також вагомий вплив має людський фактор. Дослідження показують, що рекрутери витрачають лише кілька секунд для першого оцінювання анкети. Тому важливо, щоб документ був чітко структурований, містив актуальну інформацію, підкреслював ключові навички та досягнення кандидата [2].

Добре структуроване резюме повинно відповідати наступним критеріям:

- структурованість та поділ на відповідні розділи;
- використання ключових слів для відповідності вимогам вакансії;
- використання сучасних форматів оформлення.

Використання автоматизованих систем для створення резюме сприяє підвищенню якості анкет кандидатів та збільшенню їх шансів на успішний

відбір. Штучний інтелект допомагає не лише спростити процес написання резюме, але й оптимізувати його зміст відповідно до вимог ринку праці [3].

1.2 Відповідальність користувача при створенні резюме

Незважаючи на можливості автоматизованих сервісів, значна частина успіху залежить від самого користувача. Однією з ключових проблем є правильність введення даних про себе. Некоректне або неповне заповнення резюме може призвести до його відхилення ще на етапі автоматичного аналізу в системах ATS, які використовують роботодавці для попереднього відбору кандидатів. Такі системи орієнтовані на пошук резюме, що найбільше відповідають заданим критеріям, тому навіть незначні недоліки можуть суттєво вплинути на підсумкову оцінку кандидата.

Серед ключових помилок можна виділити:

- відсутність чіткості у навичках та використання таких слів, як «комунікабельний», що ускладнює оцінку професійного рівня кандидата;
- помилки в граматиці та орфографії можуть створити враження недбалості та недостатньої грамотності кандидата;
- відсутність чіткої структури та логічної послідовності призводить до збільшення часу необхідного на аналіз та сприйняття інформації, вимагаючи від рекрутера додаткового часу для пошуку ключових даних;
- використання складних або специфічних термінів унеможлиблює автоматичний аналіз та може ускладнити роботу рекрутерів;
- додавання в CV деталей, що не відносяться до вакансії, може відволікти від основних переваг кандидата та перенасичує його профіль.

ІІІ-системи можуть значно зменшити виникнення таких помилок, шляхом аналізу введених даних та наданням користувачу необхідних виправлень та рекомендацій щодо стилю письма [3]. Вони можуть пропонувати відповідні ключові слова, видаляти зайву інформацію або адаптувати текст для конкретної вакансії. Крім того, такі системи можуть

оцінювати ефективність анкети за певними параметрами, такими як відповідність навичок вакансії, логічність структури, правильно підібраних ключових слів та унікальність тексту.

Інтеграція автоматизованих сервісів, що базуються на штучному інтелекті, дозволяє кандидатам підвищити якість своїх документів, зменшити ризик допущення критичних помилок та значно покращити свої шанси на успішне працевлаштування.

1.3 Використання програмних застосунків для автоматизації створення резюме

Стрімкий технологічний розвиток надав можливості для створення різноманітних онлайн-сервісів, які пропонують методи автоматизованого створення CV на основі базової інформації про користувача, що значно спрощує процес написання та оформлення резюме. Сучасні платформи, такі як Resumaker, Jobseeker, Zety та інші, надають можливість користувачам швидко створювати професійно оформлені документи, використовуючи готові шаблони та алгоритми оптимізації тексту.

ResumeMaker пропонує велику кількість різноманітних шаблонів, які можна персоналізувати, що надає користувачам можливості для простого та зручного заповнення структури резюме з урахуванням загальноприйнятих стандартів. Jobseeker надає можливості для використання ШІ для в процесі аналізу введених даних і створення контенту, який адаптується під конкретну вакансію, що дозволяє досягти відповідності резюме вимогам роботодавців. Zety, у свою чергу, запропонував інтерактивний конструктор резюме з вбудованими порадами, які допомагають користувачам поліпшити текст документа, підкреслити ключові навички та усунути типові помилки [4].

Автоматизовані сервіси мають численні переваги, головною з яких є значне прискорення процесу створення портфолію, оскільки користувачам не потрібно самостійно структурувати інформацію або думати над

формулюваннями – система автоматично підказує найкращі варіанти та забезпечує відповідність сучасним стандартам. Більшість платформ також надають професійні шаблони, які адаптовані під різні вакансії, дозволяючи легко налаштувати профіль під конкретні посади. Важливою особливістю таких сервісів є автоматичне виправлення помилок та покращення стилю написання, що допомагає зробити документ більш привабливим для потенційних роботодавців [5].

Проте, незважаючи на переваги, такі сервіси мають багато обмежень при їх використанні, що може ускладнити процес роботи з ними. До основної проблеми можна віднести недостатній рівень персоналізації тексту – більшість систем працюють за шаблонним принципом, що може призвести до зменшення рівня унікальності та знизити рівень цікавості кар'єрного профілю серед великої кількості анкет. Використання стандартних шаблонів часто веде до втрати індивідуальності, оскільки усі створені описи мають схожу структуру та стиль подання інформації. Ще однією суттєвою проблемою є обмежений функціонал безкоштовних версій сервісів, через що користувачі можуть мати певні обмеження на можливість редагування або експорту документа без додаткових фінансових витрат.

Отже, використання програмних застосунків для автоматизації створення резюме є ефективним рішенням для швидкого та професійного оформлення документів. Проте кандидатам слід бути уважними до можливих обмежень таких сервісів та здійснювати власні коригування для підвищення відповідності кваліфікаційного опису конкретним вимогам вакансій в разі нагальної потреби [6].

1.4 Огляд існуючих рішень

Стрімкий розвиток цифрових технологій та штучного інтелекту спричинив появу різноманітних онлайн-сервісів, які дозволяють автоматизувати процес створення резюме прибравши необхідність

самостійного написання інформації, замінивши її простим введенням загальних даних про себе. Такі програмні застосунки надають можливість користувачам створювати за короткий час професійно оформлені документи, що відповідають сучасним вимогам ринку праці [7,8].

Існуючі рішення можна розділити на кілька категорій:

- сервіси для створення резюме надають користувачам можливість заповнювати попередню підготовлені шаблони та завантажувати резюме у форматах PDF або DOCX для подальшого їх використання. Вони пропонують мінімальну персоналізацію, орієнтуючись на швидкість і простоту використання;

- системи на основі ШІ – надають можливості для аналізу введених даних, оптимізації тексту під необхідну вакансію, пропонують виправлення та рекомендації щодо покращення структури вже готових резюме;

- комплексні стратегії для трудового розміщення являють собою багатофункціональні сервіси, що поєднують у собі створення резюме з іншими можливостями, такими як пошук вакансій, автоматичний аналіз відповідності кандидата до вимог роботодавця. При необхідності також можна скористатися наданими кар'єрними порадами.

1.4.1 Resumaker

Resumaker – це простий та зручний у використанні застосунок, який дозволяє користувачам створювати резюме за допомогою заповнення необхідною інформацією шаблону. Платформа надає обмежені можливості для персоналізації, проте забезпечує швидкий і зручний процес створення CV. Головний недолік полягає в тому, що текст може бути шаблонним, та не надати необхідного рівня унікальності тексту. В свою чергу основною перевагою Resumaker є його простота використання та швидкість створення готового до використання CV. Кандидати можуть отримати готове резюме всього за кілька хвилин з моменту початку роботи, що є значною перевагою

у випадках, коли потрібно терміново подати заявку на вакансію. Однак платформа має багато обмежень в своєму функціоналі:

- відсутність можливості редагування текстовий даних. Досупний користувачу текст неможливо самостійно відредагувати;
- відсутність глибокого аналізу контенту та надання рекомендацій щодо покращення тексту.

Всі ці недоліки створюють додаткові перепони для користувачів, та не дозволяють створювати унікальні та відповідно адаптовані під необхідну користувачу вакансію.

The image shows a web form for creating a resume. The 'Personal Info' section has the following fields: Job Title (with a help icon), First Name (filled with 'Михаил'), Last Name (filled with 'Матвеев'), Email (filled with 'mishamatveev2004@gmail.com'), and Phone. There is an 'Upload Photo' button and a 'More Details' link. Below this is the 'Summary' section with a text editor and a 'Pre-Written Examples' link. The 'Work Experience' section is partially visible at the bottom.

Рисунок 1.1 – Користувацький інтерфейс створення резюме у Resumeker

Даний інструмент підходить в першу чергу для тих, хто потребує за коротких час та без зайвих зусиль створити просте за змістом резюме та надати його за необхідністю рекрутеру в критичних моментах. Водночас для тих кандидатів, яким вкрай необхідний високий рівень індивідуалізації та адаптованості CV під вакансію, необхідно бути готовим до самостійного вдосконалення тексту документа або використати додаткові сервіси для оптимізації змісту.

1.4.2 Canva

Canva – це графічний редактор, який крім дизайну для соціальних мереж та маркетингових матеріалів пропонує можливості для створення готових шаблонів резюме. Користувачі можуть редагувати стиль, кольори та розташування блоків, що робить сервіс зручним для створення візуально привабливих CV.

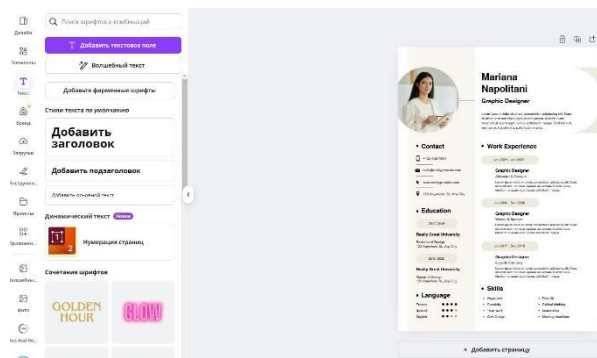


Рисунок 1.2 – Приклад використання Canva для створення резюме

Однією з основних переваг Canva є його простота використання для користувачів без досвіду в графічному дизайні, завдяки чому вони можуть легко створити якісні резюме, використовуючи широкий вибір готових шаблонів. Платформа також надає можливість експорту резюме в різних форматах, таких як PDF, PNG і JPG, що дозволяє адаптувати документ під різні вимоги потенційних роботодавців.

Однак, незважаючи на візуальну привабливість, Canva має свої обмеження. Платформа не надає інструментів для аналізу змісту, автоматичного виправлення помилок або рекомендацій щодо покращення тексту готового резюме. Крім того, відсутність підтримки функціоналу роботи з ключовими словами та оптимізації тексту під ATS може стати суттєвою перешкодою для кандидатів, CV яких може бути відхилене ще до перегляду рекрутером.

1.4.3 Jobseeker

Jobseeker, на відміну від інших аналогів використовує алгоритми ШІ для створення резюме та його можливого удосконалення. Застосунок аналізує введені дані, пропонує корекцію тексту, вносить зміни в граматичну складову та автоматично надає поради щодо кращого формулювання навичок та досвіду. Основна перевага Jobseeker полягає у здатності автоматично обирати ключові слова відповідно до вакансії та використовувати їх для складання опису. Платформа надає рекомендації з покращення структури, доповнення або видалення інформації, а також набір інструментів для швидкої зміни та налаштування дизайну.

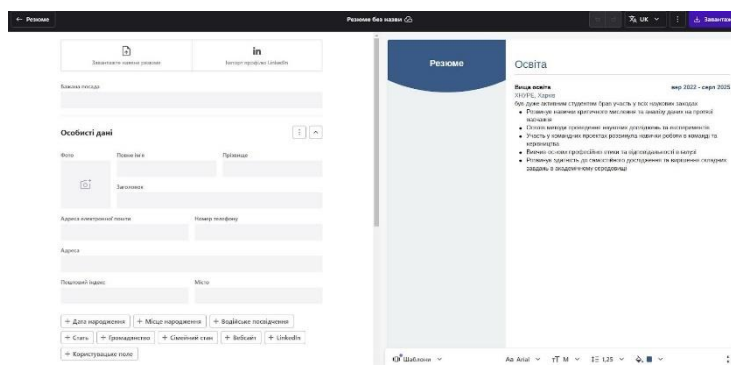


Рисунок 1.3 – Використання Jobseeker для створення CV

Також до переваг можна віднести можливість адаптації тексту до поточних вимог ринку, що особливо корисно для користувачів без досвіду у складанні професійного резюме.

До недоліків можна віднести необхідність придбання платної версії для отримання доступу до всіх наявних функцій, а саме до функцій глибокого аналізу тексту та автоматичного створення унікального змісту. Крім того, автоматичні рекомендації не завжди враховують конкретні вимоги певної компанії чи галузі, через що необхідно додатково редагувати готовий документ під вимоги вакансії.

1.4.4 Zety

Zety – це один з найпопулярніших сервісів для створення резюме, на сьогодні, який використовує алгоритми ШІ для оптимізації тексту та підбору ключових слів. Платформа аналізує введені користувачем дані, надає рекомендації щодо покращення змісту та допомагає адаптувати резюме під конкретну професію. Крім того, Zety пропонує широкий вибір шаблонів, які поєднують професійну структуру та сучасний дизайн, що дозволяє кандидатам створювати правильно стилізовані та стандартизовані оформлені документи, які надаються роботодавцям.

The screenshot displays the Zety resume builder interface. On the left is a dark navigation sidebar with icons for 'Heading', 'Education', 'Work History', 'Skills', 'Summary', and 'Finalize'. The main content area is titled 'What's the best way for employers to contact you?' and includes a form for contact details. The form fields are: FIRST NAME (Mikaeline), SURNAME (Martosca), PROFESSION (e.g. Sr. Accountant), CITY (Харків), POSTAL CODE (e.g. 80331), COUNTRY (Україна), PHONE (+380966441254), and EMAIL (mikaelinev2004@gmail.com). There are 'Preview' and 'Next: Education' buttons at the bottom. A preview of the resume is shown on the right, and a 'Change template' button is visible below it.

Рисунок 1.4 – Загальний вигляд платформи Zety при створенні резюме

Однією з основних переваг сервісу є його можливість автоматично вносити зміни в текст, виправляючи граматичні та стилістичні помилки. Це значно збільшує ймовірність успішного проходження ATS-фільтрів та привертає увагу рекрутерів.

Однак, незважаючи на те, що сервіс надає базові можливості безкоштовно, більшість функцій, серед яких розширене редагування, рекомендації щодо змісту та експорт резюме без порушень авторських прав, доступні лише у платній версії.

1.4.5 Kickresume

Kickresume – це професійний програмний застосунок для створення CV, який комбінує простий інтерфейс з функціями ШІ. Цей сервіс дозволяє швидко створити резюме за допомогою готових шаблонів, а також автоматично створювати текст з використанням GPT-моделі. ШІ-алгоритми аналізують введену інформацію та надають пропозиції щодо її вдосконалення, наприклад редагування граматики, доповнення змісту, оптимізація структури та підбір лексики, що відповідає обраній професії.



Рисунок 1.5 – Загальний вигляд платформи Kickresume при створенні резюме

Платформа має інтуїтивно зрозумілі поля редагування, за допомогою яких користувач може внести зміни в кожен блок резюме, обирати стиль оформлення та завантажити готовий документ у форматі PDF. Важливою перевагою є наявність бази реальних прикладів резюме з різних галузей, що можуть використовуватися як шаблони. Також треба відзначити, що Kickresume пропонує інструменти для створення супровідних листів і мотиваційних заяв, що є важливою частиною сучасного процесу працевлаштування. Завдяки використанню штучного інтелекту платформа допомагає виділити ключові навички кандидата, підвищити привабливість резюме для роботодавця та адаптувати документ відповідно до вимог конкретної вакансії або сфери діяльності.

Kickresume надає зручний набір інструментів навіть для початківців, оскільки всі функції логічно структуровані, а процес створення резюме супроводжується підказками та прикладами. Інтерфейс підтримує кілька мов, що дозволяє створювати міжнародні резюме. Крім того, користувачі можуть відстежувати історію змін, швидко повертатися до попередніх версій документа та створювати кілька варіантів резюме для різних вакансій.

1.5 Порівняльний аналіз програмних рішень

В результаті проведеного аналізу існуючих онлайн-сервісів які призначені для створення резюме було виявлено закономірність в тому, що кожне з існуючих рішень має загальний недолік, який пов'язаний з недостатньою кількістю безкоштовного функціоналу для роботи, через що виникає необхідність придбання платної версії, щоб отримати доступ до всіх доступних функцій. Деякі платформи спрямовані на швидке створення резюме за допомогою шаблонів, інші використовують штучний інтелект для аналізу та оптимізації змісту, в той же час більшість з них пропонує широкий набір дизайнерських інструментів для створення візуально привабливих документів. Результати проведеного аналізу існуючих рішень та їх характеристик наведено у вигляді таблиці.

Таблиця 1.1 – Порівняння характеристик програмних застосунків

Характеристики	Програмні рішення				
	Resumaker	Jobseeker	Zety	Canva	Kickresume
Персоналізація	-	-	-	-	+
Редагування тексту	+	+	+	+	+
Платні функції	+	+	+	+	+
Експорт даних	+	+	+	+	+
Використання ШІ	+	-	-	-	+

1.6 Постановка задачі

Даний проєкт є веб-застосунком для автоматизованого створення резюме з використанням технологій штучного інтелекту. Ця система забезпечує крок за кроком збір даних про користувача через модальні вікна, формує структуру резюме, відображає його у вигляді перегляду, і надає можливість створення фінального документа за допомогою обраної ШІ-моделі. Аналізуючи існуючі рішення та виявлені в них обмеження, були сформульовані вимоги до функціоналу майбутнього застосунку, зокрема передбачене створення простого та зрозумілого інтерфейсу з покроковим введенням інформації, підтримкою кількох мов, гнучкими функціями редагування даних, а також збереження інформації у необхідному форматі.

Особливу увагу було приділено інтеграції лінгвістичних моделей штучного інтелекту, таких як GPT, Mistral та Claude, для автоматичного створення змісту резюме відповідно до потреб користувача. Передбачено реалізацію функції попереднього перегляду створеного документа, можливість його завантаження у форматі PDF, а також перспективу подальшої інтеграції з платформами працевлаштування. Розробка орієнтована на користувачів, які не мають технічної підготовки, але потребують інструменту для швидкого створення професійного та адаптованого до конкретної вакансії резюме.

1.6.1 Системні вимоги

Розроблене рішення для автоматизованого створення резюме з використанням штучного інтелекту призначене для використання в пошукових системах, таких як Google, Mozilla або Microsoft Edge останніх версій. Веб-застосунок реалізований як клієнтська веб-сторінка, що потребує лише доступ до Інтернету та дозвіл на виконання JavaScript-коду. Для повноцінної роботи та забезпечення необхідного рівня безпеки користувач

повинен самостійно надати дозвіл на збереження даних у локальному сховищі браузера (LocalStorage або IndexedDB), що забезпечує збереження введеної інформації між сесіями. Система підтримує роботу на ПК та оптимізована для роздільної здатності екрану від 1280×720 і вище. Платформа також передбачає можливість збереження створених в процесі роботи документів у форматі PDF.

1.6.2 Опис функцій програмного забезпечення

Для виконання поставленого завдання, розроблений застосунок має мати такі функції:

- покрокове введення персональних даних через модальні вікна з можливістю редагування;
- збереження зібраної інформації у форматі JSON;
- формування попереднього перегляду резюме за допомогою зібраної інформації;
- можливість незалежного редагування кожної секції резюме безпосередньо у вікні попереднього перегляду;
- можливість вибору користувачем ШІ-моделі для створення та вдосконалення резюме;
- виправлені та оформлення тексту на основі введених даних за допомогою ШІ;
- завантаження готового документа у форматі PDF.

2 АНАЛІЗ ВИКОРИСТОВУВАНИХ ТЕХНОЛОГІЙ

2.1 Огляд засобів розробки

Розробка сучасних веб-застосунків потребує використання надійних та гнучких інструментів, які можуть забезпечити підвищення швидкості розробки, стабільність та масштабованість. У процесі розробки програмного забезпечення для веб-застосунку AI Resume Builder, який автоматизує створення резюме на основі введених користувачем даних з подальшою генерацією тексту за допомогою ШІ-моделі, були використані інструменти, що відповідають цим вимогам: JavaScript, FastAPI, MySQL, Visual Studio Code, а також інтерфейси REST API для зв'язку з обраною генеративною мовною моделлю.

JavaScript є основною мовою програмування для створення інтерфейсу застосунків, що забезпечує динамічну взаємодію користувача з веб-інтерфейсом. Його можливості дозволяють реалізувати інтерактивні модальні вікна, перевірку введених даних, попередній перегляд створеного резюме та інші функції UI/UX, що є необхідними для зручної та швидкої роботи користувача [9].

Для серверної частини використовується FastAPI – це один із найтехнологічніших фреймворків для Python, та надає можливість створення високопродуктивних REST API. Завдяки асинхронній обробці запитів, FastAPI дозволяє ефективно обробляти запити незалежно від їх типу та даних які були введені користувачем включаючи авторизацію та вибір необхідного шаблону резюме [10].

MySQL використовується для керування та управління доступом до бази даних, що дозволяє зберігати дані користувачів, структуру резюме та вибрані параметри. Його основними перевагами є стабільність, підтримка транзакцій і сумісність з FastAPI через ORM бібліотеки, такі як SQLAlchemy

або Tortoise ORM [11].

Інтегрований редактор коду VS Code має важливе значення для щоденної роботи над проектом. Завдяки розширенням для мов програмування Python, JavaScript, інтеграції з Git та підтримці форматів OpenAPI/Swagger, VS Code сприяє ефективному керуванню проектом та використанню різноманітних API [12].

Особливу увагу в рамках реалізації проекту приділено підключенню до моделі штучного інтелекту, яка виконує операцію створення резюме на основі введеної інформації. Для цього реалізується взаємодія із зовнішнім ШІ-сервером через HTTP-запити. У майбутньому також заплановано реалізацію системи вибору шаблонів оформлення, індикатора інформації про прогрес заповнення даними та завантаження резюме у формат PDF – усе це вимагає стабільної взаємодії між клієнтською та серверною частиною через API та передачу структурованих даних [13].

2.2 Технології реалізації застосунку

2.2.1 VS Code в якості редактора коду

Обґрунтований вибір середовища розробки має ключову роль при розробці застосунків різного ступеня складності, особливо в тих випадках, коли проект об'єднує клієнтський інтерфейс, серверну логіку та зовнішні API-інтеграції. Для створення інтелектуальної системи автоматизованого складання резюме дуже важливо обрати правильне середовище розробки – легке, але потужне середовище з відкритим кодом, яке має бути індустріальним стандартом для розробників.

Однією з ключових переваг VS Code є його широка підтримка зовнішніх модулів, що дозволяє адаптувати середовище до потреб конкретного проекту. У цьому застосунку були використані розширення, такі як Prettier для автоматичної стандартизації коду, JavaScript ES6 Snippets для

оптимізації написання типового коду, Live Server для швидкого локального перегляду результатів без потреби в окремому сервері. Ці засоби сприяють підвищенню швидкості розробки, зниженню кількості синтаксичних помилок і загальному підвищенню якості коду [14].

VS Code має можливість роботи з Git, що забезпечує неперервну інтеграцію змін у проєкті, ведення історії версій і ефективну спільну роботу. Вбудовані засоби для роботи з кодом дозволяють виконувати налагодження програмного коду, перевіряти його на помилки, відстежувати логіку взаємодії з сервером та аналізувати виконання програм, що особливо важливо в контексті складних асинхронних запитів до мовної моделі.

VS Code також підтримує взаємодію з віртуальними середовищами, контейнерами та командною строкою, що сприяє централізованому керуванню залежностями проєкту, правильній роботі локального середовища, взаємодії з сервером (наприклад, у FastAPI), а також тестуванню API за допомогою REST-клієнтів, таких як Thunder Client чи REST Client.

2.2.2 Javascript

Мова програмування JavaScript займає важливе місце в сучасній веб-розробці завдяки своїй універсальності та здатності до створення інтерактивних інтерфейсів. При реалізації інтелектуальної системи складання резюме ця мова була обрана як основний інструмент для розробки клієнтської частини веб-інтерфейсу. JavaScript дозволяє ефективно взаємодіяти з користувачем, зокрема, створювати динамічні модальні вікна, обробляти введені дані, забезпечувати попередній перегляд тексту резюме та відправляти інформацію на сервер за допомогою асинхронних HTTP-запитів (за допомогою методів `fetch`, використовуючи бібліотеку `Axios`) [15].

Javascript має декілька переваг, серед яких важливо відзначити підтримку подієво-орієнтованої моделі програмування, яка дозволяє гнучко керувати реакцією інтерфейсу на дії користувача, та широку підтримку з

боку браузерів, що унеможлиблює залежність від платформених обмежень. Важливим фактором на користь вибору цієї мови також є розвинена екосистема бібліотек і фреймворків, яка дозволяє швидко масштабувати функціонал, інтегрувати інтерфейсні модулі вибору шаблонів, реалізувати індикатор очікування або налаштувати завантаження створеного PDF-файлу.

2.2.3 FastAPI

Для реалізації серверної логіки інтелектуальної системи автоматизованого складання резюме обрано FastAPI – високопродуктивний фреймворк на мові Python.

Ключовим критерієм вибору цієї технології є її швидкодія, яка досягається завдяки використанню асинхронних обчислень, а також вбудованій підтримці автоматичного створення документації у форматі OpenAPI, що значно полегшує масштабування системи [16].

Необхідно також виділити те, що FastAPI має функціонал для використання стандартизації, що дозволяє забезпечити високий рівень контролю за вхідними і вихідними даними, зменшуючи кількість помилок при розробці. У контексті даного проєкту це особливо важливо, оскільки система приймає структуровані дані від користувача (персональні дані, досвід, навички, побажання до вакансій), які мають бути передані до зовнішньої мовної моделі у чіткому форматі. Крім того, FastAPI має зручну інтеграцію з такими бібліотеками, як SQLAlchemy (для ORM), OAuth2 (для авторизації) та rудantic (для перевірки коректності схем), що значно спрощує побудову архітектури застосунку.

2.2.4 PyCharm як середовище розробки

У процесі створення інтелектуальних веб-систем важливу роль відіграє вибір відповідного середовища розробки, яке забезпечить ефективне

написання, налагодження та підтримку серверного коду. У контексті реалізації серверної частини системи автоматизованого складання резюме з використанням штучного інтелекту, було обрано інтегроване середовище розробки PyCharm, яке спеціалізується на підтримці проєктів мовою програмування Python.

PyCharm забезпечує повну інтеграцію з інструментами, використовуваними в сучасній розробці серверних застосунків, включаючи веб-фреймворки, такі як FastAPI, бібліотеки для роботи з базами даних (ORM SQLAlchemy), системи керування версіями (Git) та інструменти тестування. Також це середовище надає можливості для глибокого аналізу коду, автоматичного створення документації, перевірки типів та керування залежностями проєкту, що є важливим в умовах необхідності швидкого створення прототипів програм та масштабування їх функціоналу [17].

2.2.5 Система управління базами даних MySQL

У складі веб-застосунків, які обробляють користувацькі дані, система управління базами даних (СУБД) відіграє критично важливу роль, забезпечуючи збереження, доступність та цілісність інформації. Для реалізації СУБД у межах проєкту інтелектуального складання резюме було обрано MySQL – реляційну систему з відкритим кодом, яка зарекомендувала себе як стабільне, масштабоване та високопродуктивне рішення [18].

Головною перевагою MySQL у цьому проєкті є його здатність обробляти структуровані дані з чіткими зв'язками між таблицями. Це особливо важливо у випадках, коли система працює з великою кількістю логічно пов'язаних об'єктів, таких як облікові записи користувачів, збережені шаблони, історії генерацій, сесії або результати. MySQL підтримує транзакції, що забезпечує цілісність операцій в процесі збереження взаємозалежних сутностей. MySQL дозволяє здійснювати ефективні вибірки навіть при зростанні їх обсягу, що є важливою перевагою.

У поєднанні з Python-орієнтованими ORM-рішеннями, зокрема бібліотекою SQLAlchemy, MySQL не лише спрощує взаємодію з наборами даних різного типу та розміру, а також забезпечує абстрагування логіки зберігання, завдяки чому розробник працює з об'єктами, а не з таблицями напряму. Всі ці функції відкривають можливості для налаштування та редагування структури інформації необхідності самостійного контролю. Такий підхід сприяє підтримці принципів чистої архітектури та забезпечує розділення відповідальностей між рівнями логіки застосунку.

2.2.6 Авторизація користувачів

Веб-застосунки, що збирають особисті або конфіденційні дані, повинні забезпечувати надійний механізм аутентифікації та авторизації. У контексті інтелектуальної системи для автоматизованого складання резюме дана особливість є необхідною при вводі інформації, що відображає професійний шлях, освіту та кар'єрні наміри. Забезпечення приватності та контролю доступу є обов'язковими умовами функціонування системи, спрямованої на персоналізований сервіс.

Для здійснення процесу авторизації було вирішено використовувати стандартну архітектуру на основі токенної аутентифікації, де ключову роль відіграють протокол OAuth2 та бібліотеки для створення токена безпеки. Використання такого підходу у створенні програмних застосунків надає необхідний рівень безпеки, тому що токен доступу додається до кожного запиту. Така модель є ефективною для масштабування системи та реалізації багаторівневого доступу [19]. Система токенів дозволяє ізолювати публічні та приватні маршрути на сервері, забезпечити часові обмеження доступу (expiration), контроль сесій та уникнення стороннього доступу до критичних операцій, таких як перегляд історії резюме або завантаження PDF-файлів. З технічного погляду реалізація авторизації в FastAPI тісно пов'язана з middleware-функціями, які перевіряють заголовки запиту (headers),

виконують процес декодування токенів та передають отриману інформацію про авторизованого користувача до бізнес-логіки. Це дозволяє зберігати цілісність архітектури, розмежовувати основну логіку та безпеку, і водночас уникнути проблем дублювання коду [20].

2.2.7 API для інтеграції з моделлю штучного інтелекту

Зараз інтелектуальні системи, які використовують моделі природної мови, все частіше звертаються до зовнішніх сервісів, що надають обчислювальні можливості для генерації текстів на основі вхідних даних [21]. При розробці системи автоматизованого складання резюме важливою складовою є інтеграція з генеративною мовною моделлю, яка виконує функцію створення змістового та логічно-структурованого тексту документа за запитом користувача [22]. Ця інтеграція здійснюється через REST API для забезпечення стандартизованої взаємодії між серверною частиною застосунку та хмарною мовною моделлю GPT [23].

З технічної точки зору, архітектура такого рішення ґрунтується на використанні JSON-формату для передачі даних, що забезпечує універсальність і гнучкість у побудові запитів [24]. Сервер формує запит до API моделі, вказуючи вхідні параметри, такі як бажана структура, стиль, інформація про освіту, досвід, навички користувача та інші необхідні блоки, після чого отримує у відповідь готовий текст [25].

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Загальна структура веб-застосунку

Програмна реалізація представлена у вигляді односторінкового HTML-документа з використанням мови JavaScript. Основна мета – зібрати в зручному вигляді необхідну інформацію від користувача для створення резюме, зробити перетворення отриманих даних в формат JSON для того, щоб передати підготовлену інформацію в модель штучного інтелекту для зручного відображення готових результатів користувачу застосунку для їх подальшого використання [26].

Застосунок містить такі основні компоненти:

- HTML-структура – блоки з кнопками, модальними вікнами, полями вводу та динамічними секціями;
- CSS-оформлення – стилізація для інтерфейсу;
- JavaScript-логіка – містить усі функції обробки даних, функції обміну інформацією між користувачем та ШІ.

Для покращення користувацького досвіду, всі дані, які вводить користувач в процесі своєї роботи із застосунком зберігаються в глобальній змінній `resumeData` та можуть при необхідності використовуватися повторно.

Лістинг 3.1 – Структура об'єкта з даними резюме `resumeData`

```
let resumeData = {
  personalInfo: {},
  languages: [],
  courses: [],
  experience: [],
  education: [], skills: [],
  aiModel: ''}
```

Завдяки використанню окремої змінної для збереження результатів, досягається спрощення формування JSON-документу, із усією введеною

інформацією, необхідною для роботи моделі ШІ.

Інтерфейс програми побудований як серія модальних вікон, які відкриваються послідовно де кожне з них відповідає розділу резюме [27].

3.2 Обробка введення даних користувача

На кожному етапі введення даних (у відповідних модальних вікнах) інформація зчитується з полів форми і зберігається у відповідну секцію об'єкта `resumeData`. Кожна категорія має власну функцію збереження, яка викликається автоматично при переході до наступного кроку (через функцію `goToNextModal()`).

3.2.1 Функція `savePersonalInfoToData()`

Ця функція відповідає за збереження персональної інформації користувача, введеної у модальному вікні `#personalModal`. Дані зчитуються з полів ПІБ, Посада, Email, Телефон, Місто та Дата народження, і зберігаються в об'єкт `resumeData.personalInfo`.

Лістинг 3.2 – Функція збереження персональних даних користувача

```
function savePersonalInfoToData () {
  resumeData.personalInfo = {
    fullName: document.getElementById('fullName').value,
    position: document.getElementById('position').value,
    email: document.getElementById('email').value,
    phone: document.getElementById('phone').value,
    city: document.getElementById('city').value,
    dob: document.getElementById('dob').value};}
```

Функція не повертає значення, а лише оновлює відповідне поле в глобальній змінній `resumeData`.

У разі, якщо поля залишено порожніми, у подальшому відображення таких даних буде замінене на дефіс (-).

3.2.2 Функція saveLanguagesToData()

Ця функція виконує збирання даних з модального вікна #languagesModal, у якому користувач може вказати одну або декілька мов та рівень володіння кожною з них. Усі введені пари «мова-рівень» додаються у список resumeData.languages.

Лістинг 3.3 – Функція збереження даних про мови

```
function saveLanguagesToData () {
  resumeData.languages = [];

  document.querySelectorAll('.language-entry').forEach(entry => {
    const name = entry.querySelector('input[id$="_name"]').value;
    const level = entry.querySelector('input[id$="_level"]').value;
    if (name)
    {
      resumeData.languages.push({ name, level });
    }

  });
}
```

Функція виконує обробку всіх елементів із класом language-entry, зчитує значення з полів name та level після чого додає їх до списку, якщо поле «мова» не є порожнім. Такий підхід унеможливорює запис неповних або випадкових записів. У кожному об'єкті зберігається назва мови (name) та рівень володіння нею (level).

Надалі ці дані використовуються для відображення інформації у розділі «Мови» при формуванні попереднього вигляду резюме.

3.2.3 Функція saveCoursesToData()

Ця функція обробляє дані, введені користувачем у модальному вікні #coursesModal, де вводиться інформація про пройдені користувачем професійні курси підвищення кваліфікацій [28].

Користувач може додати довільну кількість курсів, але кожен курс має містити такі дані:

- Назва курсу;
- Платформа або установа де був пройдений курс;
- Посилання на сертифікат з пройденого матеріалу;
- Дата завершення проходження;
- Короткий опис того, про що був курс.

Функція `saveCoursesToData()` зчитує дані з усіх елементів з класом `course-entry` та формує список об'єктів, який автоматично зберігається у `resumeData.courses`.

Лістинг 3.4 – Функція збереження даних про курси

```
function saveCoursesToData() {resumeData.courses = [];
  document.querySelectorAll('.course-entry').forEach(entry => {
    resumeData.courses.push({
      title: entry.querySelector('input[id$="_title"]').value,
      platform: entry.querySelector('input[id$="_platform"]').value,
      link: entry.querySelector('input[id$="_link"]').value,
      date: entry.querySelector('input[id$="_date"]').value,
      desc: entry.querySelector('textarea[id$="_desc"]').value
    });});}
```

Ця функція не виконує перевірку даних, тому бажано, щоб користувач самостійно стежив за їх заповнення та коректністю.

3.2.4 Функція `saveExperienceToData()`

Ця функція обробляє дані з вікна `#experienceModal`, в якому користувач може додавати записи про свій досвід роботи:

- назву компанії де працював користувач;
- посада на минулому місці роботи;
- період в продовж якого користувач працював;
- місце (місто/країна) де розташовувалася компанія;
- обов'язки та особливі досягнення користувача.

Функція `saveExperienceToData()` виконує обробку всіх елементів з класом `experience-entry`, зчитує введені дані та формує список об'єктів, який зберігається у `resumeData.experience`.

Лістинг 3.5 – Функція збереження досвіду роботи

```
function saveExperienceToData() {
  resumeData.experience = [];
  document.querySelectorAll('.experience-entry').forEach(entry => {
    resumeData.experience.push({
      company: entry.querySelector('input[id$="_company"]').value,
      position: entry.querySelector('input[id$="_position"]').value,
      period: entry.querySelector('input[id$="_period"]').value,
      location: entry.querySelector('input[id$="_location"]').value,
      tasks: entry.querySelector('textarea[id$="_tasks"]').value
    });
  });
}
```

Записи зберігаються у вигляді об'єктів з п'ятьма ключами: `company`, `position`, `period`, `location`, `tasks`. Усі вони використовуються під час формування фінального тексту резюме та попереднього перегляду.

3.2.5 Функція `saveEducationToData()`

Ця функція відповідає за обробку та збереження введення інформації про рівень освіти користувача, яка вводиться у відповідному модальному вікні `#educationModal`.

У цьому вікні користувач може додати декілька навчальних закладів, кожен із яких містить:

- назву навчального закладу;
- факультет або спеціальність;
- рівень освіти (бакалавр, магістр тощо);
- роки впродовж яких користувач проходив навчання;
- місто або країну;
- додаткову інформацію (диплом, проєкти, стипендії тощо).

Функція `saveEducationToData()` виконує обробку всіх елементів з класом `education-entry`, зчитує значення з відповідних полів і формує список об'єктів для `resumeData.education`.

Лістинг 3.6 – Функція збереження даних про освіту

```
function saveEducationToData() {
  resumeData.education = [];
  document.querySelectorAll('.education-entry').forEach(entry => {
    resumeData.education.push({
      institution:
entry.querySelector('input[id$="_institution"]').value,
      specialty: entry.querySelector('input[id$="_specialty"]').value,
      level: entry.querySelector('input[id$="_level"]').value,
      years: entry.querySelector('input[id$="_years"]').value,
      location: entry.querySelector('input[id$="_location"]').value,
      note: entry.querySelector('textarea[id$="_note"]').value });
  });
}
```

Ця інформація відображається у секції «Освіта» під час формування попереднього перегляду резюме, і також використовується ШІ-моделлю для створення логічного та змістовного опису користувача [29].

3.2.6 Функція `saveSkillsToData()`

Функція `saveSkillsToData()` зберігає перелік навичок, які користувач додає вручну у модальному вікні `#skillsModal`.

Лістинг 3.7 – Функція збереження навичок користувача

```
function saveSkillsToData() {
  resumeData.skills = [];
  document.querySelectorAll('#skillsList.skill-tag').forEach(tag => {
    const skill = tag.childNodes[0]?.nodeValue?.trim();
    if (skill) resumeData.skills.push(skill);});}
```

Інтерфейс реалізовано у вигляді поля вводу з кнопкою «Додати навичку», після чого кожна навичка виводиться у вигляді візуального тегу (елемент з класом `skill-tag`).

Користувач може видаляти окремі теги при необхідності. Функція зчитує всі елементи з класом `skill-tag` і автоматично зберігає їх текстовий вміст у список `resumeData.skills`.

Навички зберігаються як прості текстові рядки. У фінальному тексті резюме вони відображаються у вигляді тегів або маркованого списку.

3.3 Модальна навігація та управління інтерфейсом

Однією з ключових особливостей застосунку є послідовна навігація між модальними вікнами, що дозволяє користувачу зручно заповнювати резюме крок за кроком з можливістю повернутися до вже заповнених даних та внести необхідні зміни при необхідності. Перехід між секціями реалізовано за допомогою функцій `openModal()`, `closeModal()` та `goToNextModal()`. Також реалізовано динамічне відображення прогресу етапів заповнення.

3.3.1 Функція `openModal(id)`

Функція `openModal()` відповідає за відкриття модального вікна з відповідним `id`. За допомогою унікального ідентифікатора виконується зміна активного вікна. Завдяки чому користувач може змінити вже внесені дані.

Лістинг 3.8 – Функція відкриття модального вікна

```
function openModal(id) {
  document.getElementById(id).style.display = "block";
  updateProgressBar(id);

  if (id === 'previewModal') renderPreview();
}
```

Ця функція викликає оновлення відображення прогресу, передаючи `id` поточного модального вікна у функцію `updateProgressBar()`.

3.3.2 Функція closeModal(id)

Коли користувач завершує роботу з модальним вікном і закриває його, автоматично виконується функція closeModal() яка закриває вікно з відповідним ідентифікатором

Лістинг 3.9 – Функція закриття модального вікна

```
function closeModal(id) {
  document.getElementById(id).style.display = "none";
}
```

3.3.3 Функція goToNextModal(currentId, nextId)

Функція goToNextModal() реалізує перехід від одного модального вікна до іншого. В процесі роботи функції виконується збереження інформації в залежності від поточного id, після чого активне вікно закривається і відбувається перемикання на наступне.

Лістинг 3.10 – Функція переходу між модальними вікнами

```
function goToNextModal(currentId, nextId) {
  if (currentId === 'personalModal') savePersonalInfoToData();
  if (currentId === 'languagesModal') saveLanguagesToData();
  if (currentId === 'coursesModal') saveCoursesToData();
  if (currentId === 'experienceModal') saveExperienceToData();
  if (currentId === 'educationModal') saveEducationToData();
  if (currentId === 'skillsModal') saveSkillsToData();
  if (currentId === 'modelSelectModal') saveAiModelToData();
  if (currentId === 'skillsModal' && nextId === 'previewModal') {
    renderPreview();
  }
  closeModal(currentId);
  openModal(nextId);
}
```

3.3.4 Функція updateProgressBar(currentModalId)

Функція відповідає за візуальне оновлення даних про заповнення користувачем інформації про себе. В процесі роботи користувача

автоматично визначається індекс поточного кроку на основі змінної `steps`, обчислюється відсоток заповнення та відображається відповідний текст.

Лістинг 3.11 – Функція оновлення прогрес-бара

```
function updateProgressBar(currentModalId) {
  const index = steps.indexOf(currentModalId);
  const progressBar = document.getElementById('progressBar');
  const progressText = document.getElementById('progressText');

  if (index !== -1)
  {
    const percent = ((index + 1) / steps.length) * 100;
    progressBar.style.width = `${percent}%`;
    progressText.textContent = `Крок ${index + 1} з ${steps.length} –
    ${stepNames[currentModalId]}`;
  }
}
```

Функція забезпечує інтуїтивну взаємодію з користувачем, дозволяючи відстежувати поточний статус заповнення.

Це особливо важливо при великій кількості кроків, оскільки створює у користувача відчуття прогресу. Його було реалізовано за допомогою простих HTML-елементів `div`, які стилізовано через CSS, а відсоток заповнення оновлюється плавно завдяки властивості `transition`.

3.4 Формування попереднього перегляду резюме

Після заповнення всіх модальних вікон користувач переходить до етапу формування попереднього перегляду резюме, який реалізовано у вигляді модального вікна `#previewModal`. На цьому етапі усі введені дані з об'єкта `resumeData` об'єднуються в структуровану HTML-розмітку перед передачею інформації в обрану ШІ-модель.

Формування попереднього вигляду автоматично здійснюється функцією `renderPreview()`, в процесі виклику вікна `previewModal` або самостійно користувачем при переході з секції «Навички».

3.4.1 Функція renderPreview()

Функція виконує обробку даних в структурі resumeData та по черзі додає кожен блок до HTML-рядка, який згодом додається до елемента з ідентифікатором previewContent.

В процесі збереження виконується автоматична перевірка даних і у разі їх відсутності виконується зміна на дефіс (-).

Лістинг 3.12 – Функція формування HTML-прев'ю з даних резюме

```
function renderPreview() {
  let html = '';
  const { personalInfo, languages, courses, experience, education,
  skills } = resumeData;

  html += `

### 👤 ${personalInfo.fullName || '-'}</h3>`; html += ` <strong>Посада:</strong> ${personalInfo.position || '-'}</p>`; if (languages.length) html += formatList('Мови', languages, 'name', 'level'); if (courses.length) html += formatList('Курси', courses, 'title', 'platform'); if (experience.length) html += formatList('Досвід', experience, 'position', 'company'); if (education.length) html += formatList('Освіта', education, 'institution', 'specialty'); if (skills.length) { html += `📌 Навички</h4>`; html += skills.map(s => `


```

Функція формує просту та логічну структуру резюме, яку користувач бачить перед остаточним затвердженням. Впровадження такого методу дає змогу опрацьовувати всі дані та за потреби повернутись до редагування.

3.5 Серверна частина: API, база даних та авторизація

Серверна частина веб-застосунку реалізована за допомогою фреймворку FastAPI, який забезпечує швидку обробку HTTP-запитів,

підтримує асинхронність і автоматично створює документацію (OpenAPI). Код організовано окремим проєктом. Структура наведена на рисунку 3.5.1.

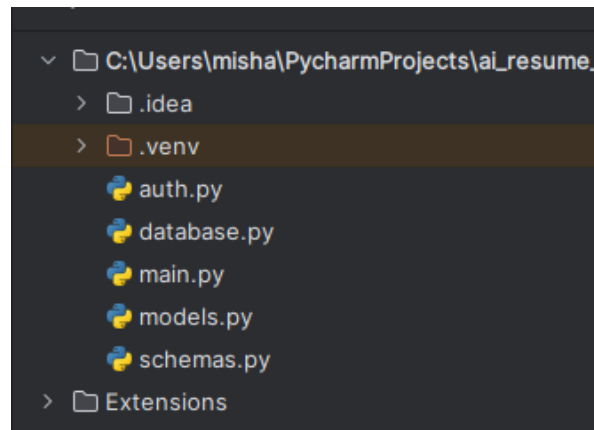


Рисунок 3.1 – Побудова файлової структури у PyCharm

3.5.1 Файл main.py

Файл main.py є точкою входу до серверної частини застосунку. У ньому оголошуються маршрути для реєстрації, авторизації користувачів та CRUD-операцій над резюме і його компонентами.

Лістинг 3.13 – Реєстрація користувача

```
@app.post("/register")
def
    register(user: UserCreate, db: Session = Depends(get_db)):existing_user
= db.query(User).filter(User.email == user.email).first()

    if existing_user:raise HTTPException(status_code=400, detail="Користувач
існує")
    hashed_password      =      bcrypt.hash(user.password)new_user      =
User(name=user.name, email=user.email, password_hash=hashed_password)
        db.add(new_user)db.commit()
    return {"message": "Користувач створений"}
```

Даний файл використовується для виконання операцій реєстрації та аутентифікації користувачів, створення резюме та отримання CRUD для персональних даних, мов, курсів, досвіду, освіти, навичок.

3.5.2 Файл models.py

Файл містить SQLAlchemy-моделі, які відображають структуру побудованої інформаційної системи:

- User (користувачі);
- Resume (резюме);
- PersonalInfo, Language, Course, Experience, Education, Skill (компоненти резюме).

Лістинг 3.14 – Модель користувача

```
class User(Base):
    __tablename__ = "users"
    id = Column(Integer, primary_key=True)
    name = Column(String(100))
    email = Column(String(100), unique=True)
    password_hash = Column(String(255))
```

3.5.3 Файл schemas.py

Файл містить Pydantic-схеми для перевірки вхідних та вихідних даних на можливі помилки та спроби несанкційного доступу до інформації.

Лістинг 3.15 – Схема для створення резюме

```
class ResumeCreate(BaseModel):
    title: str
    content: str
```

3.5.4 Файл auth.py

Лістинг 3.16 – Функція створення токена

```
def create_access_token(data: dict, expires_delta: timedelta = None):
    to_encode = data.copy()
    expire = datetime.utcnow() + timedelta(days=7)
    to_encode.update({"exp": expire})
    return jwt.encode(to_encode, SECRET_KEY, algorithm=ALGORITHM)
```

Файл реалізує функції створення JWT-токенів для авторизації.

3.5.5 Файл database.py

Файл визначає параметри підключення методів доступу до інформації в БД і створює сесію для взаємодії.

Лістинг 3.17 – Підключення до MySQL

```
engine = create_engine(SQLALCHEMY_DATABASE_URL)
SessionLocal = sessionmaker(bind=engine)
Base = declarative_base()
```

3.5.6 Основна логіка API

У main.py для кожного компонента резюме (PersonInfo, Language, Course, Experience, Education, Skill) реалізовано маршрути для створення та отримання даних. Усі запити перевіряють існування відповідного резюме та забезпечують ізольовану роботу даних кожного користувача. Для представлення структури даних, що використовуються у маршрутах API, побудовано діаграму зв'язків сутностей (ER-діаграму).

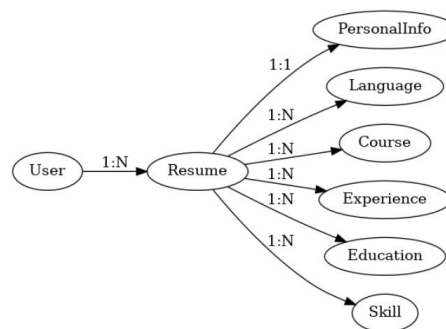


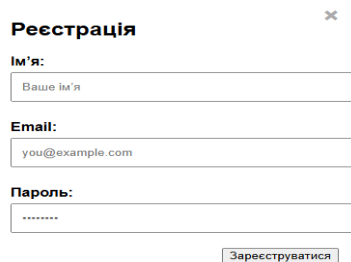
Рисунок 3.2 – Діаграма зв'язків моделей бази даних (ER-діаграма)

Таким чином, серверна частина забезпечує ефективну взаємодію між користувачами, резюме та пов'язаними з ним даними. Завдяки продуманій структурі бази даних і реалізації REST API, система гарантує збереження цілісності даних та можливість подальшого розширення функціоналу.

4 ІНСТРУКЦІЯ КОРИСТУВАЧА

4.1 Реєстрація та вхід до системи

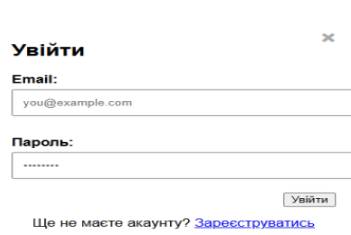
Після запуску застосунку AI Resume Builder користувачу пропонується увійти або зареєструвати новий обліковий запис, який дозволить зберігати введені дані та працювати із персональним кабінетом. Для реєстрації користувач має обрати відповідну кнопку, яка відкриє вікно для введення інформації про ім'я, поштову адресу та код доступу.



The image shows a modal window titled "Реєстрація" (Registration) with a close button (X) in the top right corner. It contains three input fields: "Ім'я:" (Name) with the placeholder text "Ваше ім'я", "Email:" with the placeholder text "you@example.com", and "Пароль:" (Password) with a masked input field showing "*****". Below the fields is a button labeled "Зареєструватися" (Register).

Рисунок 4.1 – Модальне вікно реєстрації

В разі успішної реєстрації, користувач має змогу ввести електронну пошту та пароль на сторінці входу.



The image shows a login form titled "Увійти" (Login) with a close button (X) in the top right corner. It contains two input fields: "Email:" with the placeholder text "you@example.com" and "Пароль:" (Password) with a masked input field showing "*****". Below the fields is a button labeled "Увійти" (Login). At the bottom, there is a link: "Ще не маєте акаунту? [Зареєструватись](#)".

Рисунок 4.2 – Форма входу в систему

Після введення правильних даних та натискання кнопки «Увійти», відкривається основна сторінка програми.

4.2 Основна панель та навігація

На головній сторінці застосунку розміщено підказки для заповнення кожного розділу резюме: «Персональна інформація», «Мови», «Курси», «Досвід», «Освіта» та «Навички». Також доступні кнопки для виходу з облікового запису та індикатор прогресу, який вказує на етап заповнення.

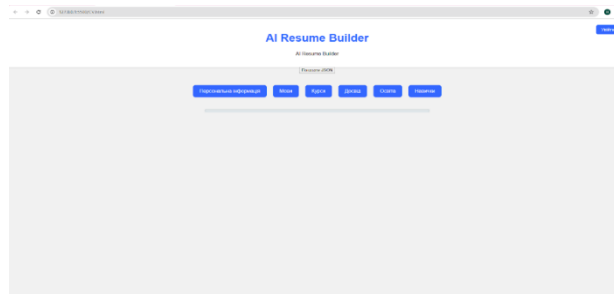


Рисунок 4.3 – Головна сторінка застосунку з панеллю навігації

4.3 Введення персональних даних

Після натискання кнопки «Персональна інформація» відкривається модальне вікно для введення необхідної інформації.

Рисунок 4.4 – Введення персональної інформації

Для введення особистих даних використовуються відповідні поля, такі як: ПІБ, посада, електронна пошта, телефон, місто та дата народження.

Заповнені дані автоматично зберігаються та використовуватимуться для створення резюме.

4.4 Введення інформацію про володіння іноземними мовами

На наступному етапі користувач може вказати одну чи кілька мов та визначити свій рівень володіння ними.

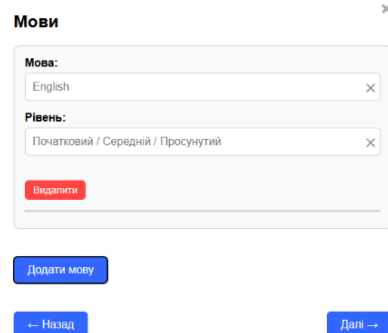


Рисунок 4.5 – Введення інформації про мови

Програма також надає можливість додавати нові мови або видаляти вже внесені. Даний функціонал призначений для використання незалежно від мови, якої володіє користувач, та надає широкі можливості з розповсюдження застосунку за межі України.

4.5 Додавання курсів

У секції «Курси» користувачу необхідно заповнити інформацію про його професійні навички, пройдені курси, зазначаючи назву курсу, платформу чи організацію, посилання на сертифікат, дату закінчення та короткий опис. Доступно необмежене число курсів для додавання. Обмеження необхідне для запобігання перевантаженню CV великою кількістю інформації. Тому користувач заповнює лише найважливішу

інформацію про себе, що в свою чергу спрощує роботу як автоматичними системами оцінки резюме так і рекрутером.

The screenshot shows a form titled "Курси" (Courses). It contains several input fields with placeholder text and a red "Видалити курс" (Delete course) button at the bottom. Below the form are three navigation buttons: "Додати курс" (Add course), "Назад" (Back), and "Далі" (Next).

Курси

Назва курсу:
Наприклад, Deep Learning Specialization

Платформа / Установа:
Coursera, Prometheus...

Посилання на сертифікат:
https://...

Дата завершення:
03.2024 або просто рік

Опис курсу:
Короткий зміст або теми курсу

Видалити курс

Додати курс

Назад

Далі

Рисунок 4.6 – Введення інформації про курси

Завдяки використанню відповідних кнопок «Вперед» і «Назад», користувач може повернутися до заповнення інформації.

4.6 Введення досвіду роботи

Секція досвіду використовується для перелічення інформації про минулі місця роботи: назва організації, посада, термін зайнятості, місце роботи (місто/країна) та опис обов'язків чи досягнень.

The screenshot shows a form titled "Досвід роботи" (Work Experience). It contains several input fields with placeholder text and a red "Видалити досвід" (Delete experience) button at the bottom. Below the form are three navigation buttons: "Додати досвід" (Add experience), "Назад" (Back), and "Далі" (Next).

Досвід роботи

Компанія:
Наприклад, SoftServe

Посада:
Frontend Developer

Період роботи:
01.2022 — 03.2023

Місто / країна:
Київ, Україна

Обов'язки / досягнення:
Опис обов'язків, інструментів, результатів

Видалити досвід

Додати досвід

Назад

Далі

Рисунок 4.7 – Введення інформації про досвід роботи

4.7 Внесення інформації про освіту

Користувач вказує назву навчального закладу, факультет чи спеціальність, рівень освіти, період навчання, місто та будь-яку додаткову інформацію (наприклад, диплом із відзнакою).

Рисунок 4.8 – Введення інформації про освіту

Користувач може заповнити інформацію для всіх навчальних закладів без обмежень по кількості введених даних.

4.8 Введення навичок

У розділі «Навички» користувач має можливість заповнити інформацію про свої професійні вміння, які згодом з'являться у фінальному резюме.

Рисунок 4.9 – Введення навичок

4.9 Формування попереднього вигляду резюме

Після заповнення всіх секцій форми з'являється можливість попереднього перегляду заповнення резюме, в якому демонструється документ перед збереженням. У користувача є можливість додати зміни в будь-який розділ інформації до моменту створення готового документа.

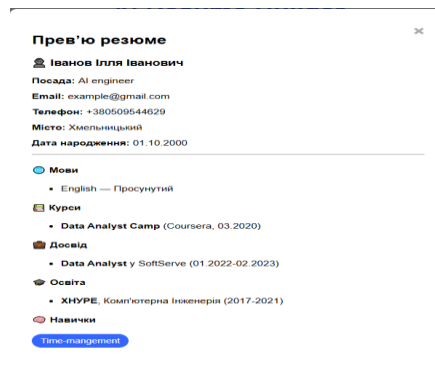


Рисунок 4.10 – Попередній вигляд резюме

Попередній перегляд дозволяє виявити помилки, які були допущені при заповненні інформації, та додати необхідні дані перед збереженням.

4.10 Вибір ШІ-моделі

На останньому етапі користувач обирає модель штучного інтелекту для створення тексту резюме: GPT, Mistral, LLaMA, Bloom, T5 або Claude

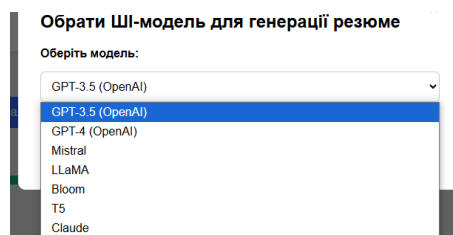


Рисунок 4.11 – Вибір мовної моделі для генерації резюме

4.11 Відображення прогресу заповнення

В процесі заповнення модальних вікон можна побачити індикатор прогресу, який демонструє номер активного етапу і загальну кількість кроків яку вже було пройдено.

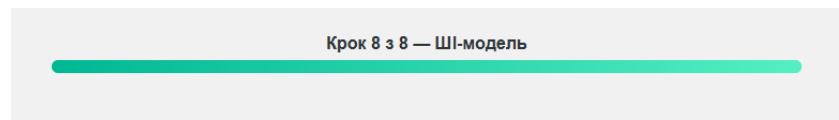


Рисунок 4.12 – Індикатор прогресу заповнення резюме

4.12 Вибір шаблону резюме

Після обрання ШІ-моделі, користувачу автоматично пропонується обрати шаблон оформлення резюме відповідно до типу вакансії. Наприклад, шаблон для ІТ-вакансій має стриманий стиль з сірими та синіми відтінками, мінімалістичним дизайном та акцентом на структурованість, що підходить для розробників, DevOps-фахівців, аналітиків тощо.

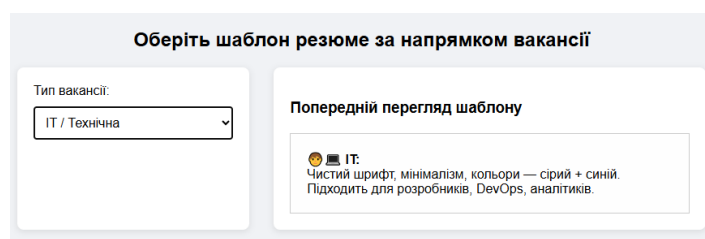


Рисунок 4.13 – Вибір шаблону резюме за напрямком вакансії

4.13 Завершення роботи та отримання готового результату

По завершенню всіх необхідних кроків користувач обирає ШІ-модель та шаблон резюме. Система на основі введеної інформації, обраної моделі та

типового шаблону, формує підсумковий документ, який стає доступним для користувача [30]. Користувач також може переглянути, оцінити та завантажити готовий документ у форматі PDF. Нижче наведено приклад резюме кандидата на посаду AI\ML Engineer, створеного за результатами роботи застосунку [31].

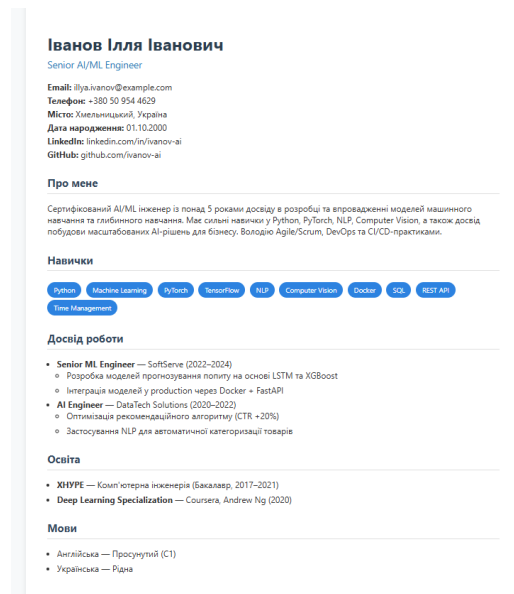


Рисунок 4.14 – Згенероване резюме, готове до завантаження

Після введення даних та вибору шаблону користувач може переглянути стилізоване резюме у форматі, що нагадує професійний. У програмі реалізовано кілька шаблонів для різних типів вакансій, включаючи IT, фінанси та креативні спеціальності. Кожен шаблон має унікальний дизайн, сучасну графіку, чітку структуру та логічне розбиття інформації. Після вибору шаблону користувач може створити резюме, після чого система автоматично додасть покращений текст, який сформує обрана модель ШІ, у стандартній стилізованій HTML-документ. Завдяки функції збереження, користувач може повертатися до раніше створених резюме, переглядати або знову використовувати їх. Для зручності передбачено візуальну аналітику, яка відображає кількість мов, навичок, досвіду, курсів тощо. Під час тестового використання застосунку серед обмеженої групи користувачів було

помічено, що ймовірність отримання відповіді від рекрутерів значно зростає, якщо резюме було створене за допомогою розробленої інтелектуальної системи.

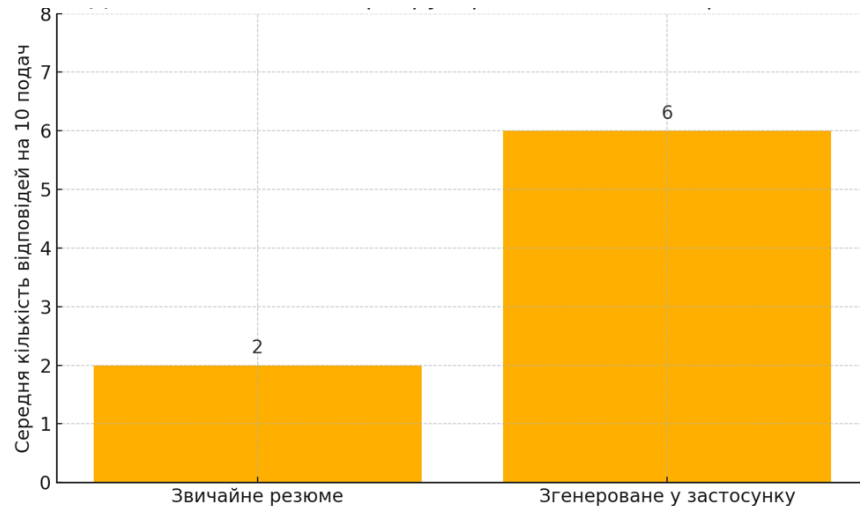


Рисунок 4.15 – Динаміка відповідей рекрутерів до і після використання застосунку

Зокрема, користувачі, що використовували застосунок, у середньому отримували втричі більше відповідей на вакансії, ніж ті, хто надсилав резюме, створене самостійно. Це підтверджує ефективність використання розробленого застосунку та його відповідність сучасним вимогам ринку праці. Це підтверджується наведеним нижче графіком, що демонструє динаміку відповідей рекрутерів до та після використання застосунку.

ВИСНОВКИ

Дипломна робота присвячена створенню інтелектуальної системи для автоматизованого написання резюме в основі якої лежить використання технологій ШІ.

В процесі виконання аналізу аналогів було виявлено, що переважна більшість наявних платформ не забезпечують достатнього рівня персоналізації, не зберігають структуровані вхідні дані користувачів для подальшого редагування, мають обмежену варіативність стилістичних шаблонів та часто не підтримують генерацію змісту без підключення до зовнішніх серверів.

Для вирішення цих проблем був розроблений веб-застосунок, який дозволяє користувачу поетапно вводити персональні дані, обирати тип вакансії, стиль оформлення резюме, а також створювати текст за допомогою вбудованих локальних мовних моделей штучного інтелекту. Отриманий результат представлений у графічному вигляді з можливістю подальшого редагування та завантаження у форматі PDF-документа.

Клієнтську частину було реалізовано за допомогою HTML, CSS та JavaScript, а серверну – з використанням фреймворку FastAPI зі застосуванням SQLAlchemy для взаємодії зі збереженою в процесі роботи інформацією. Унікальністю цієї системи є здійснення інтеграції локальних модулів ШІ, що в свою чергу дозволяє користувачам виконувати операції без необхідності підключення до мережі Інтернет та підвищує рівень конфіденційності даних.

Усі вказані завдання були виконані в повному обсязі. У майбутньому планується розширення асортименту шаблонів бібліотеки, удосконалення облікового запису користувача, зокрема впровадження історії збережених резюме, а також розширення мовних моделей для забезпечення підтримки декількох мов.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Вимоги щодо взаємодії із споживачами при врегулюванні простроченої заборгованості. URL: <https://bankalliance.ua/pages/cards/credit/overdue/Інформація%20щодо%20врегулювання%20простроченої%20заборгованості.pdf>. (дата звернення: 12.04.2025).
2. Гетьман О. О., Безбатько А. О. Підготовка молодих спеціалістів (фахівців) у навчальних закладах відповідно до вимог ринку праці. Економіка та держава. 2016. Т. 1. С. 75–80. – URL: <http://www.economy.in.ua/>. (дата звернення: 20.03.2025).
3. Homeworkify alternatives. URL: <https://aimojo.io/uk/homeworkify-alternatives/>. (дата звернення: 20.03.2025).
4. 8 зручних онлайн-сервісів для створення резюме. URL: <https://cambridge.ua/uk/blog/8-udobnyh-onlajn-servisov-dlya-sozdaniya-rezyume/>. (дата звернення: 20.03.2025).
5. Форум: обговорення сервісів створення резюме. dou.ua. URL: <https://dou.ua/forums/topic/49064/>. (дата звернення: 20.03.2025).
6. Як виконати конкурентний аналіз. URL: <https://topdigital.com.ua/blog/як-виконати-конкурентний-аналіз-клас/>. (дата звернення: 20.03.2025).
7. Moroz S. Рівень відповідність університетських знань вимогам ринку праці: аналіз думки працедавців та напрями вдосконалення механізмів державного управління у сфері забезпечення якості вищої освіти. Актуальні проблеми державного управління. 2020. № 1(57). С. 309–317. DOI: <https://doi.org/10.34213/ap.20.01.36>. (дата звернення: 20.03.2025).
8. ResumeQuill: створіть ідеальне резюме без реєстрації. infobase.com.ua. URL: <https://infobase.com.ua/resumequill-stvorit-idealne-rezyume-bez-reyestraciyi/>. (дата звернення: 20.03.2025).
9. Flanagan D. JavaScript: The Definitive Guide: Master the World's Most-

Used Programming Language. 7th Edition O'Reilly Media, 2020, 706 p.

10. Tiangolo S. FastAPI documentation. fastapi.tiangolo.com.
URL: <https://fastapi.tiangolo.com>. (дата звернення: 20.03.2025).

11. Microsoft. visual studio code documentation. code.visualstudio.com.
URL: <https://code.visualstudio.com/docs>. (дата звернення: 20.03.2025).

12. Goodfellow I., Bengio Y., Courville A. Deep learning. Cambridge: MIT Press, 2016. 800 p.

13. Wieruch R. The road to learn react: your journey to master plain yet pragmatic react.js. CreateSpace Independent Publishing Platform, 2018, 198 p.

14. JetBrains. pycharm documentation. www.jetbrains.com.
URL: <https://www.jetbrains.com/pycharm/>. (дата звернення: 20.03.2025).

15. DuBois P. MySQL cookbook: solutions for database developers and administrators. O'Reilly Media, Incorporated, 2014, 866 p.

16. Python web penetration testing cookbook / B. May et al. Packt Publishing, Limited, 2015, 224 p.

17. Tom B. Brown, Benjamin Mann, Nick Ryder. Language models are few-shot learners. Advances in neural information processing systems. 2020, 25 p.

18. Матвєєв М. С. Аналіз сучасних інструментів автоматизації створення резюме. Модернізація та сучасні українські і світові наукові дослідження : МАТЕРІАЛИ VII МІЖНАР. СТУДЕНТ. НАУК. КОНФ., м. Чернігів,, 6 груд. 2024 р. / координатор А. О. Татарников.

19. Матвєєв М. С. Автоматичне оцінювання та покращення резюме за допомогою штучного інтелекту. Наука сьогодні: від досліджень до стратегічних рішень : МАТЕРІАЛИ VIII МІЖНАР. СТУДЕНТ. НАУК. КОНФ., м. Чернігів, 11 квіт. 2025 р. / ред. А. О. Татарников.

20. Machine learning-based multi-documents sentiment-oriented summarization using linguistic treatment / A. Abdi et al. Expert systems with applications. 2018. Vol. 109. P. 66–85. DOI: <https://doi.org/10.1016/j.eswa.2018.05.010>. (дата звернення: 05.06.2025).

21. Stefano Federici, Simone Borsci. Usability evaluation: models URL:

https://www.researchgate.net/publication/228079044_Usability_evaluation_model_s_methods_and_applications. (дата звернення: 05.06.2025).

22. Ransbotham S., Khodabandeh S., LaFountain B. The state of AI in 2022 – and a half decade in review. *www.mckinsey.com*. URL: <https://www.mckinsey.com/~media/mckinsey/business%20functions/quantumblack/our%20insights/the%20state%20of%20ai%20in%202022%20and%20a%20half%20decade%20in%20review/the-state-of-ai-in-2022-and-a-half-decade-in-review.pdf>. (date of access: 02.06.2025).

23. Liang Z. AI hiring with llms: a context-aware and explainable multi-agent framework for resume screening. 2024, 10 p.

24. Chuan Qin, Le Zhang. A comprehensive survey of artificial intelligence techniques for talent analytics. 2023, 53p.

25. Gibby R. E., Ducey A. J. Connecting people and opportunities with artificial intelligence. *Talent assessment*. 2023. P. 349–C22P46. URL: <https://doi.org/10.1093/oso/9780197611050.003.0022>. (дата звернення: 05.06.2025).

26. Krug S. Don't make me think, revisited: a common sense approach to web usability. Pearson Education, 2015, 218 p.

27. Tidwell J. Designing interfaces: patterns for effective interaction design. O'Reilly Media, Incorporated, 2019.

28. Nguyen T. N. T., Lai N. V., Nguyen Q. T. Artificial intelligence (AI) in education: a case study on chatgpt's influence on student learning behaviors. *Educational process international journal*. 2024. Vol. 13, no. 2. DOI: <https://doi.org/10.22521/edupij.2024.132.7> (дата звернення: 05.06.2025).

29. Anand S., Raja R. G., Sheela T. An overview of AI platforms, frameworks, libraries, and processes. *Explainable Artificial Intelligence (XAI): concepts, enabling tools, technologies and applications*. 2023. P. 93–113. DOI: https://doi.org/10.1049/pbpc062e_ch6 (дата звернення: 05.06.2025).

30. Reuben Binns. Fairness in machine learning: lessons from political philosophy. ArXiv:1712.03586v1 [cs.cy] 10 dec 2017journal of machine learning

research. 2017. URL: <https://arxiv.org/abs/1712.03586> (дата звернення: 02.06.2025).

31. Russell S. J., Norvig P. Artificial intelligence: a modern approach. Pearson Education, Limited, 2021, 1152