

УДОСКОНАЛЕННЯ ПРОГРАМНИХ СИСТЕМ ПІДГОТОВКИ ДОКУМЕНТІВ ДЛЯ ПРАЦЕВЛАШТУВАННЯ

Хохлов О.К. – гр. МГКІ-24, магістрант, alexeykhoklov2@gmail.com

Стаценко Д.В. – к.т.н., доцент., statsenko.dv@knutd.edu.ua

Київський національний університет технологій та дизайну

Мета: Розробити програмний засіб, що автоматизує підготовку пакета документів для працевлаштування, персоналізує зміст під конкретну вакансію та підвищує ймовірність проходження ATS-фільтрів.

Останні роки ринок працевлаштування остаточно перемістився в цифровий простір: рекрутери використовують ATS-системи, кандидати шукають вакансії онлайн, а документи мають бути готові “тут і зараз” під конкретну позицію.

Для задоволення цієї потреби створено кастомізовану автоматизовану систему підготовки документів для працевлаштування - десктоп-застосунок (Windows) з інтегрованим ШІ-модулем, орієнтований на швидке створення якісних резюме, супровідних листів і пакету документів під вимоги роботодавця.

Під час роботи я послідовно вибудував повний цикл створення настільного застосунку на Python із використанням PyQt, який не просто генерує резюме та супровідні листи, а й формує для користувача відчуття безперервного, інтуїтивного процесу, коли кожен наступний крок логічно впливає з попереднього, а результат одразу виглядає професійно і готовий до подальшої взаємодії з роботодавцем. В основу покладено ідею, що якість вихідних документів визначається не стільки на фінальному етапі експорту в DOCX, скільки на етапі грамотного збору даних, ретельної валідації, продуманого структурування та доречної персоналізації під конкретну вакансію, компанію і очікування адресата. Саме тому інтерфейс організовано як зрозумілий маршрут від первинного заповнення ключових полів профілю до детального наповнення змістових блоків досвіду, освіти, навичок і мов, після чого користувач переходить до мотиваційної частини, де задаються нюанси позиції, цінності компанії та короткі акценти на релевантних досягненнях, що підсилюють переконливість тексту.

Особлива увага приділена двом взаємодоповнювальним режимам формування контенту, адже в реальних умовах кандидат може працювати як онлайн, маючи доступ до інтегрованої мовної моделі для гнучкого доопрацювання формулювань, так і офлайн, коли важливо швидко зібрати

документ за перевіреними шаблонами без залежності від зовнішніх сервісів. Такий підхід дозволив поєднати стабільність і прогнозованість офлайн-шаблонів з варіативністю та природністю мовленнєвих конструкцій, які надає генеративна модель, причому користувач у будь-який момент бачить проміжний результат у вікні попереднього перегляду, може коригувати формулювання, уточнювати дані, переформатовувати абзаци і тільки після цього натискати збереження у Word з обраною назвою та місцем розташування файлу. Кожен технічний крок у застосунку підпорядкований принципам прозорості та передбачуваності: чіткі повідомлення про стан, видимі індикатори процесу, відсутність зайвих переходів і вікон, мінімальна кількість кліків до готового результату, що знімає типові бар'єри та економить час пошукача роботи.

У межах практики було системно вибудовано модульність коду і логіки, щоб у подальшому можна було безболісно додавати нові розділи, змінювати структуру документів, підключати альтернативні шаблони, адаптувати стиль викладу під галузеву специфіку і, за потреби, локалізувати інтерфейс та вихідні тексти для різних мовних аудиторій. Результатом стала не просто робоча утиліта, а цілісне рішення, яке охоплює шлях користувача від ідеї створити конкурентне резюме до моменту, коли на робочому столі лежить охайний .docx з логічно впорядкованими розділами, коректними заголовками, акуратними відступами і зрозумілими акцентами, що добре сприймаються як автоматизованими системами відбору, так і живими рекрутерами. Сам процес розроблення дав змогу відпрацювати важливі компетенції з побудови настільного UI, інтеграції генеративних підходів до створення тексту та експорту у Word, а також навчив мислити продуктивно, з огляду на реальні сценарії використання, навантаження, обмеження доступу до мережі та потребу швидко отримувати якісний, читабельний, структурований результат. У підсумку проєкт підтвердив актуальність обраної теми, продемонстрував практичну користь для користувачів, що готують документи для працевлаштування, і заклав надійну основу для подальшого розширення функціоналу, включно з глибшою персоналізацією під вакансії, розумними підказками щодо змісту та стилю, а також можливістю адаптивно змінювати тон і структуру матеріалів залежно від вимог ринку і специфіки цільових позицій.

Висновок: Результатом роботи став модульний настільний застосунок на Python/PyQt, який забезпечує повний цикл підготовки резюме та супровідного листа від структурованого збору даних до експорту у DOCX із акцентом на зручний UX і можливість працювати як онлайн (через LLM), так і офлайн (через шаблони).

Список використаних джерел:

1. Mark Summerfield. *Rapid GUI Programming with Python and Qt*. Prentice Hall.
2. Martin Fitzpatrick. *Create GUI Applications with Python & Qt6 (PyQt6 Edition)*.
3. Alan D. Moore. *Mastering GUI Programming with Python: Develop resilient GUI applications with PyQt and Tkinter*. Packt.
4. Luciano Ramalho. *Fluent Python* (2nd ed.). O'Reilly.
5. Eric Matthes. *Python Crash Course* (2nd ed.). No Starch Press.
6. Brett Slatkin. *Effective Python* (2nd ed.). Addison-Wesley.
7. Al Sweigart. *Automate the Boring Stuff with Python* (2nd ed.). No Starch Press.