

Дворський В. О., бакалавр, Новак Д. С., доц.

Київський національний університет технологій та дизайну

РОЗРОБКА ДЕСКТОПНОГО ЗАСТОСУНКУ ДЛЯ СТВОРЕННЯ РЕЗЮМЕ

Анотація. У роботі представлено концепцію розробки десктопного застосунку для створення резюме, орієнтованого на україномовного користувача, з повною підтримкою офлайн-режиму. Обґрунтовано необхідність такої системи через недоліки наявних онлайн-сервісів (Zety, Canva, Resume.io, Europass), зокрема обмежену локалізацію, платність, відсутність автономності та кастомізації шаблонів. Запропоновано архітектуру застосунку на основі Python, з використанням бібліотеки PyQt6 для побудови інтерфейсу та SQLite для збереження даних. У рамках реалізації розроблено модульну структуру, яка дозволяє створювати, редагувати, зберігати та експортувати резюме у форматі PDF без необхідності підключення до мережі. Система забезпечує повну гнучкість форматування, адаптацію до різних ролей користувачів та можливість розширення функціоналу. Отримані результати можуть бути основою для подальшого вдосконалення засобів автоматизованого створення професійних самопрезентацій.

Ключові слова: резюме, Python, десктопний застосунок, PyQt6, SQLite, PDF, україномовний інтерфейс, генератор резюме.

Dvorsky V. O., bachelor, Novak D. S., Assoc. professor

Kyiv National University of Technologies and Design

DEVELOPMENT OF A DESKTOP APPLICATION FOR RESUME CREATION

Abstract. The paper presents the concept of developing a desktop application for creating a resume, focused on Ukrainian-speaking users, with full offline support. The need for such a system is justified due to the shortcomings of existing online services (Zety, Canva, Resume.io, Europass), in particular limited localization, payment, lack of autonomy and customization of templates. The application architecture based on Python is proposed, using the PyQt6 library to build the interface and SQLite to store data. As part of the implementation, a modular structure has been developed that allows you to create, edit, save and export resumes in PDF format without the need for a network connection. The system provides full formatting flexibility, adaptation to different user roles and the ability to expand functionality. The results obtained can be the basis for further improvement of the means of automated creation of professional self-presentations.

Keywords: resume, Python, desktop application, PyQt6, SQLite, PDF, Ukrainian-language interface, resume generator.

Вступ. У сучасному світі ефективна самопрезентація є ключовим фактором успішного працевлаштування. Резюме – це перший і часто єдиний документ, за яким рекрутер або роботодавець формує враження про кандидата. Від якості, структурованості та візуального оформлення резюме залежить, чи отримає кандидат запрошення на співбесіду, чи його профайл буде відхилено на етапі первинного відбору. За даними дослідження LinkedIn, рекрутери витрачають у середньому 6–7 секунд на перегляд одного резюме, що робить структуру та візуальну подачу інформації критично важливими елементами успіху.

Водночас, більшість доступних інструментів для створення резюме або вимагають підключення до інтернету, або є платними, або не адаптовані до українського користувача – як мовно, так і функціонально. Популярні міжнародні платформи, такі як Zety, Resume.io, Canva, незважаючи на широкий функціонал, мають суттєві обмеження для вітчизняного користувача. По-перше, більшість з них є повністю або частково платними, що обмежує доступ для студентів та початківців. По-друге, вони вимагають

постійного інтернет-з'єднання та передачі персональних даних на сторонні сервери, що створює ризики щодо конфіденційності інформації. По-третє, україномовна локалізація часто є неповною або відсутньою, що ускладнює роботу для користувачів з недостатнім рівнем володіння іноземними мовами.

Європейський сервіс Europass, хоча і безкоштовний, пропонує обмежений набір шаблонів та не дозволяє глибокої кастомізації оформлення. Це особливо критично для творчих професій, де візуальна складова резюме може бути важливішою за стандартну структуру. Крім того, онлайн-інструменти часто застосовують обмеження щодо кількості збережених документів, експортів або доступу до раніше створених резюме без оплати підписки.

З огляду на це, виникає об'єктивна потреба у створенні зручного, безкоштовного, офлайн-доступного десктопного застосунку з повною україномовною підтримкою, який дозволяє швидко створити резюме із можливістю гнучкого налаштування. Такий застосунок повинен бути інтуїтивно зрозумілим, підтримувати кілька шаблонів, зберігати дані локально, а також експортувати резюме у формат PDF або DOCX. Важливою вимогою є також забезпечення конфіденційності даних користувача через повну локальність збереження інформації без будь-яких серверних компонентів.

Постановка завдання. Метою даної роботи є проєктування та реалізація програмного продукту – десктопного застосунку для створення резюме – з урахуванням потреб різних категорій користувачів: студентів, досвідчених спеціалістів, фрилансерів та HR-менеджерів. Проєкт базується на технологіях Python, PyQt6 для побудови інтерфейсу, SQLite для збереження даних та сучасних бібліотеках для експорту документів.

Для досягнення поставленої мети визначено наступні основні завдання дослідження. По-перше, необхідно здійснити аналіз існуючих рішень у сфері створення резюме, виявити їхні переваги та недоліки, а також визначити функціональні вимоги до розроблюваного застосунку. По-друге, потрібно спроектувати архітектуру системи з урахуванням принципів модульності, розширюваності та автономності роботи. По-третє, реалізувати базову функціональність застосунку, включаючи створення, редагування та збереження резюме. По-четверте, розробити механізми експорту документів у поширені формати (PDF, DOCX) з підтримкою шаблонізації та кастомізації оформлення. По-п'яте, забезпечити повну україномовну локалізацію інтерфейсу та підтримку специфічних потреб вітчизняного ринку праці.

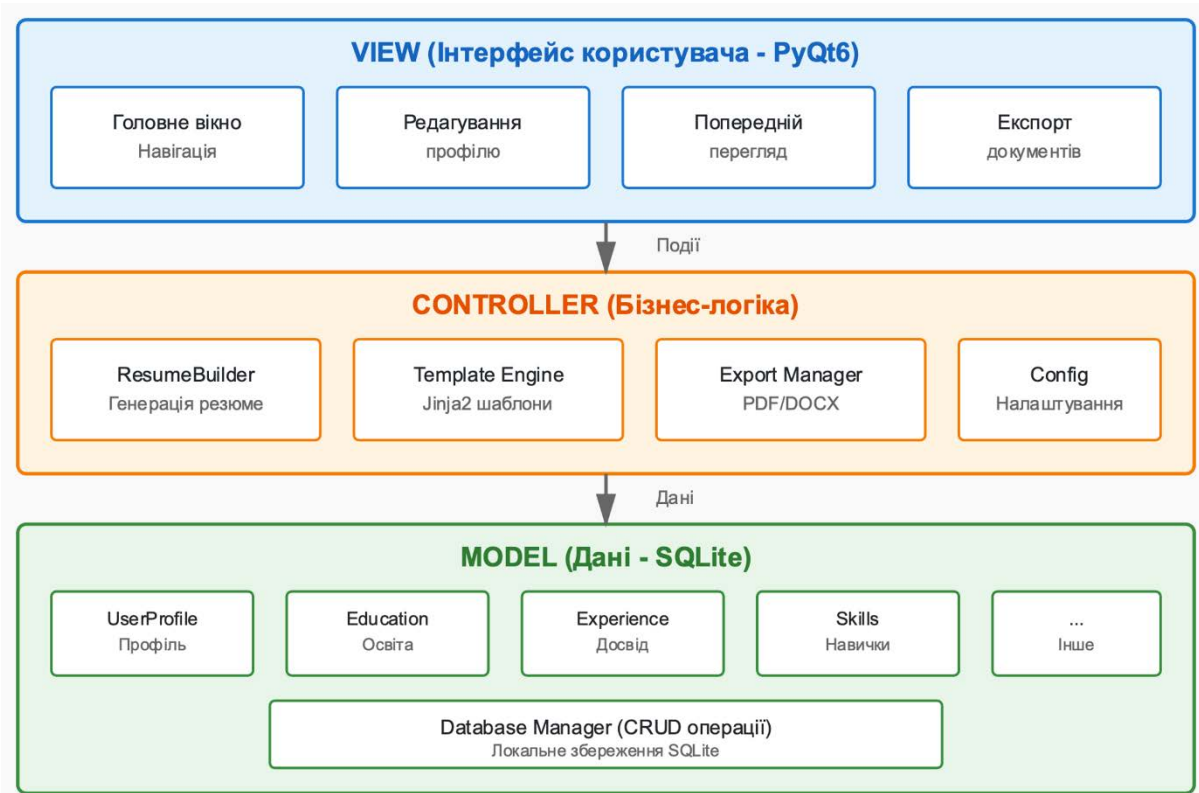
Окрему увагу приділено питанням юзабіліті та доступності інтерфейсу. Застосунок має бути зрозумілим навіть для користувачів з мінімальним рівнем комп'ютерної грамотності, що особливо важливо для студентів та осіб старшого віку, які вперше складають резюме або повертаються на ринок праці після тривалої перерви.

Результати досліджень. Архітектура розробленого застосунку для створення резюме (рис. 1) – це не просто набір модулів, а злагоджена, розширювана та масштабована система, побудована з орієнтацією на гнуту гнучкість, повторне використання та максимальну автономність користувача. Вона поєднує у собі сучасні практики інженерії програмного забезпечення, інкапсуляцію логіки в окремі компоненти та чіткий поділ відповідальностей між шарами системи [1, 2].

Шар Model відповідає за роботу з даними користувача. Модель даних побудована з максимальною деталізацією та розширюваністю. Користувацькі дані поділяються на кілька сутностей: UserProfile (основна інформація про користувача – ім'я, контакти, фото), Education (освіта – навчальні заклади, спеціальності, дати навчання), Experience (попередні місця роботи – назви компаній, посади, обов'язки, досягнення), Skills (професійні навички з можливістю вказання рівня володіння), Languages (іноземні мови), Certificates (сертифікати та додаткові кваліфікації), Contacts (детальна контактна

інформація) тощо. Кожна з цих сутностей реалізована у вигляді окремої ORM-моделі з чітко визначеними типами полів, опціональністю, а також з можливістю серіалізації в JSON. Завдяки цьому можлива повна персоналізація структури резюме під конкретного користувача та його професійну сферу.

Використання SQLite як системи управління базами даних дозволяє реалізувати повністю автономний режим роботи – усі дані зберігаються локально на комп'ютері користувача без жодної залежності від інтернету [3]. SQLite є легковаговою, файловою СУБД, яка не потребує окремого серверного процесу та може бути вбудована безпосередньо у застосунок. Це робить її ідеальним вибором для десктопних застосунків, де критична швидкість доступу до даних та автономність роботи. Усі операції з базою даних інкапсульовані у спеціалізованому модулі `db_manager`, що підтримує як базові CRUD-операції (Create, Read, Update, Delete), так і більш складні операції: фільтрацію за різними критеріями, сортування, пошук, автоматичне збереження змін. Для запуску бази не потрібно нічого встановлювати – вона створюється автоматично при першому запуску застосунку, що спрощує процес розгортання програми на різних комп'ютерах.



Джерело: власна розробка.

Рис. 1. Архітектура розробленого застосунку

Шар View реалізований на базі фреймворку PyQt6 – сучасної бібліотеки для створення графічних інтерфейсів на Python [4]. PyQt6 забезпечує кросплатформенність (застосунок працює на Windows, Linux, macOS без змін у коді), багаті можливості кастомізації зовнішнього вигляду та сучасний дизайн інтерфейсу, що відповідає актуальним трендам UI/UX. Компоненти інтерфейсу розділені на незалежні вікна та діалоги: головне вікно з навігацією, вікно редагування профілю користувача, окремі вікна для редагування освіти, досвіду роботи, навичок, контактів, вікно попереднього перегляду резюме (прев'ю), діалоги генерації документа з вибором шаблону та налаштувань, модулі імпорту та експорту профілів тощо.

Більше того, реалізовано механізм динамічного завантаження компонентів інтерфейсу, який дозволяє змінювати структуру вікон або додавати нові розділи резюме без перезапуску програми. Це особливо корисно для розширення функціоналу застосунку – наприклад, додавання нових блоків інформації (волонтерський досвід, публікації, проекти) можна здійснити через плагінну систему без модифікації основного коду.

Шаблонізація є однією з найсильніших сторін архітектури застосунку. Уся логіка побудови фінального резюме винесена у окремий шар Template Engine, що працює з HTML-шаблонами, в які динамічно підставляються дані користувача через спеціальні змінні-плейсхолдери (наприклад, `{{ name }}`, `{{ education[0].institution }}`, `{{ experience[0].description }}` тощо) [5]. Шаблони використовують мову розмітки Jinja2, яка дозволяє не лише вставляти значення змінних, але й застосовувати умовну логіку, цикли, фільтри форматування. Це дає змогу легко додавати нові шаблони різних стилів оформлення – від строгих академічних до креативних дизайнерських, а також створювати тематичні резюме під конкретні вакансії, індустрії чи стилі компаній (наприклад, для IT-компаній, корпоративного сектору, стартапів, творчих агенцій).

На основі HTML-рендерингу реалізовано експорт у формат PDF через бібліотеки WeasyPrint та pdfkit, що дає змогу формувати якісні друковані документи з повною підтримкою CSS-стилів, користувацьких шрифтів, кольорових схем, графічних елементів [6]. WeasyPrint забезпечує високу точність перетворення HTML у PDF з коректним відображенням складних макетів, таблиць, списків та інших елементів форматування. Також підтримується експорт у формат DOCX завдяки бібліотеці python-docx – важлива опція для тих користувачів, хто бажає доопрацьовувати резюме вручну у Microsoft Word або потребує формат, який легко редагується [7].

Уся логіка генерації, збереження, оновлення та експортів організована у вигляді окремого класу ResumeBuilder, який є центральним компонентом бізнес-логіки застосунку. ResumeBuilder отримує дані з моделей, обробляє їх (форматує дати, текстові поля, перевіряє на коректність), накладає на обраний шаблон і повертає згенерований документ у вигляді об'єкту, готового до експорту. Таким чином, програма легко адаптується під нові сценарії використання – від простого редагування одного резюме до повноцінної автоматизованої генерації десятків резюме з єдиної бази даних, що може бути корисним для HR-відділів або кадрових агентств.

Не менш важливою частиною системи є підсистема збереження та відновлення даних. Вона базується на серіалізації JSON-дамів усіх сутностей користувача у вигляді профілів, які можна експортувати та імпортувати окремо, наприклад, на зовнішній носій (флешку, хмарне сховище) або в іншу копію застосунку на іншому комп'ютері. Це зручно для HR-менеджерів, які хочуть створювати типові шаблони резюме для різних посад у компанії, або для користувачів, які підтримують кілька версій резюме під різні типи вакансій.

Архітектура передбачає майбутнє масштабування та розширення функціоналу. Завдяки гнучкому структуруванню модулів можна буде додати такі функції, як багатокористувацький режим (для навчальних закладів або кадрових агентств), синхронізація з хмарними сервісами (Google Drive, Dropbox), інтеграція з job-порталами через API для автоматичного заповнення профілів на сайтах Work.ua, Robota.ua, DOU, LinkedIn, система рекомендацій щодо змісту резюме на основі аналізу вакансій [8], вбудований перевірювач граматики та орфографії для україномовних текстів. Усе це можна реалізувати без необхідності перебудови всієї системи завдяки модульній архітектурі.

Усі константи, шляхи до файлів, налаштування за замовчуванням зібрані в централізованому конфігураційному модулі config.py, що дає змогу швидко адаптувати

застосунок під різні середовища, локалізації або потреби окремих організацій. Наприклад, навчальний заклад може налаштувати власний набір шаблонів та обов'язкових полів резюме для своїх випускників, а кадрова агенція – інтегрувати власні брендингові елементи у шаблони.

Висновки. У результаті реалізації архітектурної концепції застосунку для створення резюме було досягнуто низку ключових функціональних, структурних та експлуатаційних переваг, що закладають основу для його ефективного використання у різних контекстах: як для індивідуального пошуку роботи, так і для масової HR-аналітики та автоматизації рекрутингових процесів.

По-перше, запропонована багаторівнева архітектура з чітким розділенням відповідальностей між модулями (дані, логіка, інтерфейс, експорт) забезпечує не лише високу гнучкість і масштабованість системи, а й дозволяє легко вносити зміни у будь-яку частину застосунку без порушення загальної логіки роботи. Такий підхід узгоджується з сучасними принципами інженерії програмного забезпечення, зокрема концепціями Clean Architecture, SOLID-принципами та парадигмою Separation of Concerns. Це забезпечує довгострокову підтримуваність коду та можливість його розвитку різними командами розробників.

По-друге, розроблена модель даних є деталізованою, розширюваною та повністю серіалізованою, що створює умови для її інтеграції у ширші системи управління кадрами (HRM/ATS-системи), автоматизовані трекери вакансій або хмарні сервіси створення професійних профайлів. Структурованість даних у вигляді окремих сутностей дозволяє легко експортувати та імпортувати інформацію, що може бути корисним при міграції між різними платформами або при створенні централізованих баз талантів у великих організаціях.

По-третє, застосування SQLite як вбудованої бази даних дозволяє реалізувати повністю автономний, офлайн-орієнтований підхід до роботи, що особливо важливо для українських реалій – нестабільного доступу до інтернету у деяких регіонах, обмежених ресурсів у малих громадах або в умовах мобільної роботи. Це також вирішує питання конфіденційності персональних даних, оскільки вся інформація зберігається виключно на локальному комп'ютері користувача без передачі на сторонні сервери.

Крім того, інтерфейс на базі PyQt6 дозволив створити кастомізований, інтуїтивно зрозумілий та швидкий у роботі застосунок, який можна розгортати на будь-яких операційних системах (Windows, Linux, macOS) без потреби у сторонніх веб-ресурсах чи додаткових залежностях. Це особливо важливо у контексті кібербезпеки та збереження персональних даних, а також для користувачів, які працюють у закритих корпоративних мережах без доступу до інтернету.

Особливо варто відзначити реалізацію шаблонного механізму та підтримку експорту в PDF/DOCX – ці компоненти надають користувачу повний контроль над візуальним виглядом документа, дозволяють легко адаптувати оформлення під конкретну вакансію, компанію, галузевий стандарт або культурні особливості цільового ринку праці. Можливість швидкого перемикання між різними стилями резюме без повторного введення даних значно підвищує продуктивність роботи користувача.

Реалізація механізмів збереження, відновлення, а також експорту та імпорту профілів у форматі JSON відкриває перспективи для мультикористувацьких сценаріїв, створення типових заготовок та швидкої роботи з шаблонами в рекрутингових агентствах, університетських кар'єрних центрах або в освітньому середовищі (наприклад, у студентських кар'єрних сервісах, центрах зайнятості, програмах перекваліфікації).

В цілому, створена система демонструє високий рівень автономності, надійності та кастомізації, що робить її релевантною альтернативою відомим закордонним веб-

сервісам типу Zety, Canva, Resume.io, Europass тощо, з урахуванням локальних потреб українського ринку праці, повної україномовної локалізації інтерфейсу та принципів відкритого програмного забезпечення.

Список використаної літератури

1. Gamma E., Helm R., Johnson R. & Vlissidess J. Design Patterns: Elements of Reusable Object Oriented Software. Pearson Education India, 2015. 400 p.
2. Secretariat E. Agile software development handbook. In *ECSS-E-HB-40-01A*, 2020. 106 p.
3. Shields W. SQL QuickStart Guide: The Simplified Beginner's Guide to Managing, Analyzing, and Manipulating Data With SQL. E-book, 2019. 242 p.
4. Willman J. M. Creating GUIs with QT designer, in *Beginning PyQt: A Hands-On Approach to GUI Programming With PyQt6*. Cham, Switzerland: Springer, 2022, pp. 217–258.
5. Ronacher A. Jinja2 Documentation. *Pallets Projects*, 2017. URL: <https://palletsprojects.com/p/jinja/>
6. Kozea. WeasyPrint Documentation: HTML/CSS to PDF Converter. CourtBouillon, 2018. URL: <https://weasyprint.org/>
7. Python Software Foundation. Python-docx: Create and Update Microsoft Word. *docx Files*. 2016. URL: <https://python-docx.readthedocs.io/>
8. Grinberg M. Flask Web Development: Developing Web Applications with Python. 2nd ed. O'Reilly Media, 2018. 316 p.