

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення

Бакалаврська кваліфікаційна робота

на тему: «Розробка модулю інформаційної системи для керування даними персоналу підприємства»

Виконав: студент 4 курсу
групи 4ПІ-216
спеціальності

121 – Інженерія програмного забезпечення

(шифр і назва напрямку підготовки, спеціальності)

Новицький О.М.

(прізвище та ініціали)

Керівник: к.т.н., доц. каф. ПЗ Ракитянська Г.Б.

(прізвище та ініціали)

Рецензент: д.ф., старш. каф. ОТ Обертюх М.Р.

(прізвище та ініціали)

Допущено до захисту
Зав. кафедри ПЗ Романюк О.Н.
09» 06 2025 р.

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення
Рівень вищої освіти перший (бакалаврський)
Галузь знань 12 «Інформаційні технології»
Спеціальність 121 «Інженерія програмного забезпечення»
Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ
Завідувач кафедри ПЗ
О. Н. Романюк
« 21 » березня 2025 р.

ЗАВДАННЯ НА БАКАЛАВРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Новицькому Олександру Михайловичу

1. Тема роботи – розробка модулю інформаційної системи для керування даними персоналу підприємства. Керівник роботи: Ракитянська Галина Борисівна, к.т.н., доцент кафедри ПЗ, затверджено наказом вищого навчального закладу від 20 березня 2025 р. №97.

2. Строк подання студентом роботи – 2 червня 2025.

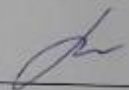
3. Вихідні дані до роботи: методи управління персоналом – оцінка KPI, прогнозування бонусів, моделювання сценаріїв; набори даних – текстові файли з інформацією про співробітників; параметри алгоритму – рейтинг продуктивності, досвід роботи, освіта, відпусткові дні; методи візуалізації – Matplotlib; програмне середовище – Python, tkinter; середовище розробки – Visual Studio Code; вихідні результати – аналітичні звіти в PDF із QR-кодами, інтерактивні графіки розподілу посад і продуктивності; вхідні дані: ім'я, посада, досвід, освіта, рейтинг, фільтри за посадою.

4. Зміст текстової частини: вступ; аналіз стану систем автоматизації управління персоналом; порівняльний аналіз аналогів; аналіз методів розв'язання задачі; постановка задач дослідження; структурний аналіз модуля для управління персоналом; моделювання архітектури системи; розробка підсистеми введення та збереження даних співробітників; розробка підсистеми інтерактивної візуалізації даних; розробка графічної схеми інтерфейсу; аналіз і обґрунтування вибору засобів для реалізації системи; розробка модуля оцінки KPI та прогнозування; розробка модуля генерації звітів у PDF; функціональне

тестування системи; розробка інструкції користувача; висновки; список використаних джерел; додатки.

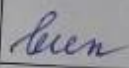
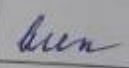
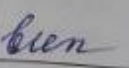
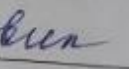
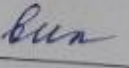
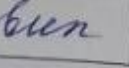
5. Перелік ілюстративної частини: зображення аналогів HR-систем; блок-схеми алгоритмів роботи модуля; схема інтерфейсу; графічний інтерфейс модуля; тестування модуля.

6. Консультанти розділів роботи

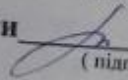
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
1-4	Ракитянська Г.Б. к.т.н., доцент кафедри ПЗ	24.03.25 	01.06.25 

7. Дата видачі завдання 24 березня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз стану систем автоматизації управління персоналом	25.03.25 – 07.04.25	
2	Розробка підсистеми введення та збереження даних співробітників	08.04.25 – 20.04.25	
3	Розробка підсистеми інтерактивної візуалізації даних	21.04.25 – 03.05.25	
4	Розробка модуля оцінки KPI та прогнозування	04.04.25 – 15.05.025	
5	Тестування роботи застосунку	16.05.25 – 24.05.25	
6	Оформлення матеріалів до захисту БКР	25.05.25 – 30.05.25	

Студент  О. М. Новицький
(підпис) (ініціали та прізвище)

Керівник бакалаврської кваліфікаційної роботи  Г. Б. Ракитянська
(підпис) (ініціали та прізвище)

АНОТАЦІЯ

УДК 004.42:658.3

Новицький О.М. Розробка модулю інформаційної системи для керування даними персоналу підприємства: бакалаврська кваліфікаційна робота зі спеціальності 121 – інженерія програмного забезпечення, освітня програма – інженерія програмного забезпечення. Вінниця: ВНТУ, 2025 – 66 с.

Бакалаврська кваліфікаційна робота складається з 64 сторінок формату А4, 25 рисунків, 3 таблиць, 4 додатків. Список використаних джерел містить 18 найменувань.

У бакалаврській кваліфікаційній роботі обґрунтовано актуальність і практичну цінність розробки модуля інформаційної системи для керування даними персоналу підприємства з використанням мови програмування Python та бібліотек tkinter, matplotlib, reportlab. Проведено аналіз існуючих HR-систем, таких як BambooHR, Workday HCM, PeopleForce та Odoo HR, із визначенням їхніх переваг і недоліків.

Розроблено архітектуру модуля, яка включає компоненти для введення та збереження даних співробітників у текстових файлах, інтерактивний графічний інтерфейс, генерацію PDF-звітів із QR-кодами, а також інтерактивні візуалізації, такі як графіки розподілу посад, діаграми продуктивності залежно від досвіду та порівняльні гістограми KPI. Проведене тестування підтвердило стабільність роботи модуля, коректність обчислень і відповідність заявленим функціональним вимогам.

Результати роботи можуть бути використані підприємствами для оптимізації управління персоналом, аналізу продуктивності, прогнозування кадрових рішень і створення аналітичних звітів. Розроблений модуль слугує ефективним інструментом для HR-менеджерів, сприяючи підвищенню ефективності та прозорості кадрових процесів.

Ключові слова: Python, tkinter, matplotlib, reportlab, HR-система, управління персоналом, KPI, прогнозування, візуалізація, PDF-звіти.

ABSTRACT

UDC 004.42:658.3

Novytskyi O.M. Development of an information system module for enterprise personnel data management: bachelor's qualification work in specialty 121 - software engineering, educational program - software engineering. Vinnytsia: VNTU, 2025 - 66 p.

The bachelor's qualification work consists of 64 pages of A4 format, 25 figures, 3 tables, 4 appendices. The list of references includes 18 titles.

The bachelor's thesis substantiates the relevance and practical value of developing an information system module for managing enterprise personnel data using the Python programming language and the tkinter, matplotlib, and reportlab libraries. An analysis of existing HR systems, such as BambooHR, Workday HCM, PeopleForce, and Odoo HR, was carried out to determine their advantages and disadvantages.

The architecture of the module was developed, which includes components for entering and saving employee data in text files, an interactive graphical interface, generating PDF reports with QR codes, as well as interactive visualizations such as position distribution charts, experience-based performance charts, and comparative KPI bar charts. The conducted testing confirmed the stability of the module, the correctness of calculations, and compliance with the stated functional requirements.

The results of the work can be used by enterprises to optimize human resources management, analyze productivity, forecast personnel decisions and create analytical reports. The developed module serves as an effective tool for HR managers, contributing to the efficiency and transparency of HR processes.

Keywords: Python, tkinter, matplotlib, reportlab, HR system, HR management, KPI, forecasting, visualization, PDF reports.

ЗМІСТ

ВСТУП.....	3
1 АНАЛІЗ СУЧАСНОГО СТАНУ ПИТАННЯ ТА ОБҐРУНТУВАННЯ ЗАВДАННЯ НА РОБОТУ	7
1.1 Аналіз стану систем автоматизованого керування персоналом	7
1.2 Порівняльний аналіз аналогів.....	8
1.3 Аналіз методів розв’язання задачі.....	13
1.4 Постановка задач дослідження	16
1.5 Висновки.....	17
2 РОЗРОБКА МОДЕЛІ ТА АЛГОРИТМІВ ПРОГРАМНОГО ПРОДУКТУ	19
2.1 Аналіз вхідних даних системи.....	19
2.2 Моделювання архітектури системи	21
2.3 Алгоритм прогнозування кар’єрного росту	23
2.4 Розробка симуляції управлінських сценаріїв	26
2.5 Розробка графічної схеми інтерфейсу.....	29
2.6 Висновки	35
3 . РОЗРОБКА ПРОГРАМНИХ МОДУЛІВ РЕАЛІЗАЦІЇ СИСТЕМИ	36
3.1 Аналіз і обґрунтування вибору засобів для реалізації системи	36
3.2 Розробка модуля управління персональними файлами співробітників	38
3.3 Розробка модуля візуалізації та генерації звітів	44
3.4 Висновки.....	51
4 ТЕСТУВАННЯ ПРОГРАМИ.....	52
4.1 Тестування системи.....	52
4.2 Розробка інструкції користувача	57
4.3 Висновки.....	58
ВИСНОВКИ	60
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	62
ДОДАТКИ	64
Додаток А – Технічне завдання.....	Error! Bookmark not defined.
Додаток Б – Протокол перевірки на плагіат	69
Додаток В – Лістинг програми	70
Додаток Г – Графічна частина.....	90

ВСТУП

Обґрунтування вибору теми дослідження. Сучасний етап розвитку інформаційних систем [1] характеризується стрімким зростанням обсягів даних про персонал підприємств, що потребує створення ефективних інструментів для їх управління, аналізу та обробки [2]. Розробка модулів інформаційних систем для керування даними персоналу є актуальною задачею, оскільки підприємства прагнуть оптимізувати процеси управління людськими ресурсами, підвищувати ефективність оцінки продуктивності працівників та прогнозувати кадрові зміни. У цьому контексті особливе значення має створення інтерактивних платформ, які забезпечують зручне введення, зберігання, аналіз і візуалізацію даних про співробітників, а також дозволяють проводити оцінку ключових показників ефективності (KPI), прогнозувати бонуси та моделювати сценарії розвитку персоналу.

Сучасні інформаційні [3] системи для управління персоналом часто обмежені у можливостях інтерактивної взаємодії, аналізу даних і прогнозування. Наприклад, стандартні системи можуть не надавати інструментів для динамічної оцінки KPI чи моделювання сценаріїв, таких як скорочення бонусів чи втрата працівників. Недоліком багатьох існуючих систем є складність інтеграції різних компонентів, таких як управління даними, аналіз продуктивності та прогнозування.

Таким чином, актуальність теми дослідження зумовлена потребою в розробці інтерактивних, наочних і комплексних інформаційних систем для управління персоналом, які враховують сучасні вимоги до аналізу даних і прогнозування.

Зв'язок роботи з науковими програмами, планами, темами. Робота виконувалася згідно плану виконання наукових досліджень на кафедрі програмного забезпечення.

Мета та завдання дослідження. Метою роботи є підвищення інтерактивності та наочності управління даними персоналу шляхом розробки

модуля інформаційної системи з використанням Python і бібліотек tkinter, matplotlib, reportlab. Це дозволить полегшити введення, аналіз і візуалізацію даних про співробітників, а також оптимізувати оцінку KPI, прогнозування бонусів і моделювання сценаріїв.

Основними задачами дослідження є:

- Провести аналіз існуючих методів і засобів управління даними персоналу для визначення напрямків підвищення їх інтерактивності та наочності;
- запропонувати нові методи підвищення інтерактивності введення, фільтрації та аналізу даних співробітників;
- створення наочних візуалізацій, таких як графіки розподілу посад, досвіду проти продуктивності та порівняння KPI;
- генерації звітів із QR-кодами для швидкого доступу до даних;
- розробити програмні компоненти та модуль інформаційної системи на основі запропонованих методів;
- провести експериментальні дослідження розроблених засобів візуалізації та аналізу ЕМ-алгоритму.

Об'єкт дослідження – процес управління даними персоналу в інформаційних системах.

Предмет дослідження – методи та засоби інтерактивного управління даними співробітників, оцінки KPI, прогнозування бонусів і візуалізації результатів.

Методи дослідження. В процесі досліджень використовувались: методи програмування на Python для реалізації модуля, бібліотеки tkinter для створення графічного інтерфейсу, matplotlib для візуалізації даних, reportlab для генерації PDF-звітів, методи аналізу даних для оцінки KPI та прогнозування бонусів, комп'ютерне моделювання для перевірки функціоналу симуляції сценаріїв (наприклад, скорочення бонусів чи втрати працівників).

Новизна отриманих результатів.

1. Алгоритм прогнозування кар'єрного росту працівників на основі

продуктивності та досвіду. Функція `calculate_bonus_forecast` реалізує модель, що враховує рейтинг продуктивності, досвід, освіту та відпустки для розрахунку бонусу та прогнозу на майбутнє. Такий підхід дозволяє приймати обґрунтовані HR-рішення з використанням простого евристичного аналізу.

2. Симуляція управлінських сценаріїв (звільнення працівників, скорочення бонусів). Функція `simulate_scenarios` дозволяє HR-аналітикам змодельовати наслідки управлінських рішень та прогнозувати як зміниться середній рейтинг після звільнення працівників або який буде ефект від скорочення бонусів працівникам.

Практична цінність отриманих результатів. Практична цінність полягає в розробці модуля інформаційної системи, який може використовуватися підприємствами для управління даними персоналу, оцінки продуктивності, прогнозування бонусів і моделювання кадрових сценаріїв. Модуль забезпечує:

- зручне введення та збереження даних у текстових файлах;
- генерацію PDF-звітів із QR-кодами для швидкого доступу до інформації;
- інтерактивні візуалізації для аналізу розподілу посад, продуктивності та KPI;
- прогнозування бонусів і моделювання сценаріїв для оптимізації кадрових рішень.

Особистий внесок здобувача. Усі наукові результати, викладені у БКР, отримані автором особисто.

Апробація матеріалів бакалаврської кваліфікаційної роботи. Описані у дослідженні положення доповідались на конференції «LIII Всеукраїнській науково-технічній конференції факультету інформаційних технологій та комп'ютерної інженерії (2025)» та опубліковані в тезах доповіді.

Публікації. За тематикою дослідження опубліковано 1 наукову працю, що доповідалась на конференції «LIII Всеукраїнській науково-технічній конференції факультету інформаційних технологій та комп'ютерної інженерії (2025)» та опубліковано в тезах доповіді [4].

Структура та обсяг БКР. БКР складається зі вступу, чотирьох розділів, висновків, списку літератури що містить 18 найменувань, 4 додатків. Робота містить 25 ілюстрації, 3 таблиці. Загальний обсяг роботи складає 103 сторінок, основний зміст викладено на 64 сторінках друкованого тексту.

1 АНАЛІЗ СУЧАСНОГО СТАНУ ПИТАННЯ ТА ОБҐРУНТУВАННЯ ЗАВДАННЯ НА РОБОТУ

1.1 Аналіз стану систем автоматизованого керування персоналом

У сучасних умовах стрімкого розвитку інформаційних технологій автоматизовані системи керування персоналом стають дедалі популярнішими серед підприємств різного масштабу [5]. Автоматизація HR-процесів відіграє ключову роль у покращенні ефективності управління, забезпечуючи швидкий та точний доступ до кадрової інформації відповідно до потреб керівництва. Розробка програмних модулів для автоматизованих систем керування персоналом є важливим завданням, що дозволяє створити ефективні інструменти для HR-відділів та оптимізувати кадрові процеси.

Однією з головних переваг впровадження систем автоматизованого керування персоналом є оптимізація процесу взаємодії між співробітниками та HR-департаментом, що дозволяє значно скоротити час на обробку кадрової документації та підвищити точність аналітики. Однак розробка таких систем є складним завданням, що потребує ретельного аналізу потреб підприємства, а також створення надійних і масштабованих алгоритмів для обробки великих обсягів персональних даних.

Багато компаній, що займаються розробкою HR-систем, звертаються до новітніх алгоритмів аналізу даних, серед яких особливу увагу привертають технології машинного навчання. Співпраця з досвідченими розробниками дозволяє швидко впроваджувати ефективні та надійні системи керування персоналом, що відповідають потребам сучасних підприємств та трендам галузі управління людськими ресурсами [6]. В результаті впровадження таких систем компанії отримують можливість оптимізувати HR-процеси без зайвих зусиль, а керівництво – приймати обґрунтовані кадрові рішення на основі актуальної аналітики.

Крім того, використання програмного забезпечення для автоматизації HR-процесів надає можливість інтегрувати різні функціональні можливості, такі як

персоналізований підбір навчальних програм, автоматичне оновлення даних співробітників, адаптація під зміни структури підприємства та багато іншого. Це не тільки спрощує процес управління персоналом, але й підвищує задоволеність співробітників, забезпечуючи їм комфортне та прозоре середовище для професійного розвитку. Таким чином, розробка програмних модулів для систем автоматизованого керування персоналом є критично важливою для сучасних підприємств. Інвестування у високоякісні IT-рішення дозволяє організаціям підвищити ефективність своєї роботи, покращити взаємодію з персоналом і забезпечити собі конкурентні переваги в галузі управління людськими ресурсами. У світлі швидкого розвитку HR-технологій важливо слідкувати за новітніми тенденціями та інноваціями в цій області, щоб залишатися на передовій кадрового менеджменту.

1.2 Порівняльний аналіз аналогів

Зі стрімким розвитком технологій сфера керування персоналом зазнає суттєвих змін, особливо у контексті автоматизації HR-процесів відповідно до потреб сучасних підприємств.

Завдяки автоматизованим системам керування персоналом компанії можуть швидко обробляти кадрову інформацію, створювати персоналізовані плани розвитку та отримувати аналітичні звіти відповідно до змін у структурі організації. У цьому контексті з'являється все більше програмних рішень, які дозволяють ефективно управляти персоналом та оптимізувати HR-процеси. Такі системи надають користувачам зручні функції, як-от автоматичний підбір кандидатів на основі професійних навичок, рекомендації щодо навчання та розвитку співробітників, формування аналітичних звітів та багато інших сервісів, що роблять управління персоналом ефективнішим та прозорішим.

Серед численних аналогів розроблюваної системи варто виділити декілька популярних рішень: BambooHR, Workday HCM, PeopleForce, Odoo HR, кожен з яких пропонує свої унікальні можливості та підходи до автоматизації HR-процесів.

На рисунку 1.1 зображено BambooHR – це комплексна HR-платформа, яка робить акцент на управлінні життєвим циклом співробітників [7]. Вона використовує інтуїтивно зрозумілий інтерфейс для відстеження кандидатів, адаптації нових працівників, управління відпустками та оцінки продуктивності персоналу.

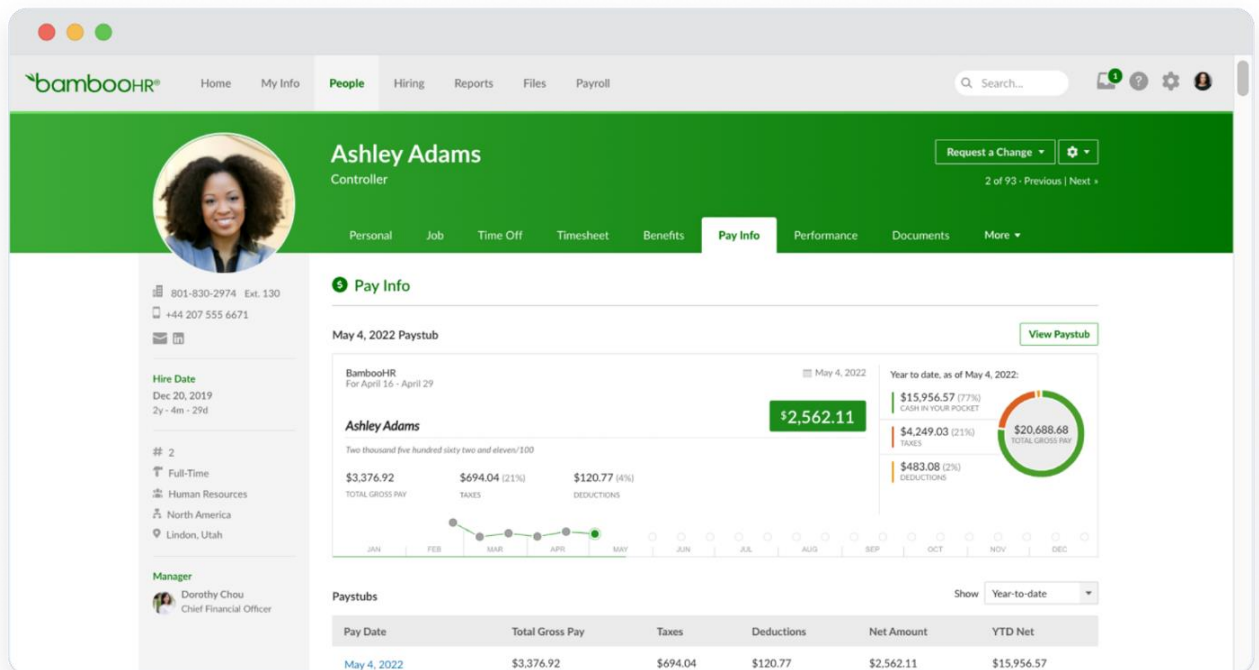


Рисунок 1.1 – Приклад роботи додатку BambooHR

На рисунку 1.2 зображено Workday HCM – потужна система управління людським капіталом, яка використовує алгоритми машинного навчання для аналізу даних про персонал [8]. Вона інтегрується з фінансовими модулями та системами планування ресурсів підприємства, що дозволяє автоматично збирати дані про ефективність роботи та формувати аналітичні звіти. Завдяки цьому система пропонує персоналізовані рекомендації щодо розвитку персоналу на основі аналізу продуктивності та кар'єрних цілей. Система підтримує функціонал прогнозування плинності кадрів і виявлення потенційних ризиків для організації. Крім того, Workday HCM забезпечує гнучке налаштування під потреби різних галузей і масштабів бізнесу.



Рисунок 1.2 – Приклад аналітичного дашборду у Workday HCM

На рисунку 1.3 зображено PeopleForce – це українська HR-система, яка поєднує в собі функції керування персоналом та можливість створення власних кадрових процесів [9]. Основна особливість PeopleForce – використання гнучкої моделі автоматизації, що адаптується під специфіку бізнесу. Система також інтегрується з популярними інструментами комунікації, що дозволяє оптимізувати взаємодію між співробітниками та HR-відділом.

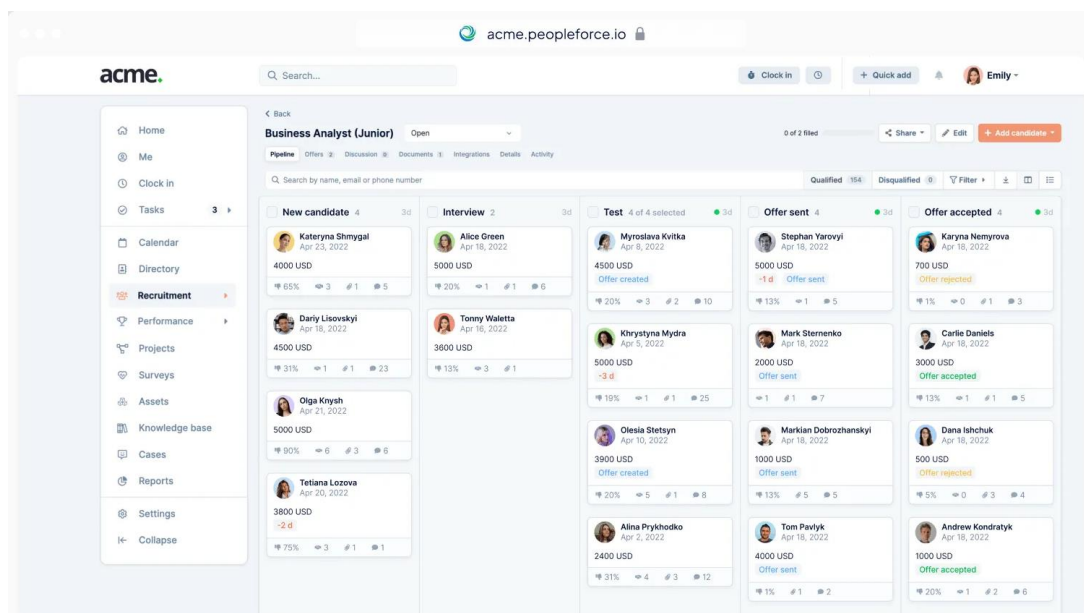


Рисунок 1.3 – Приклад керування персоналом у PeopleForce

На рисунку 1.4 зображено Odoo HR – це модуль управління персоналом у складі платформи Odoo з відкритим кодом, який дозволяє користувачам налаштовувати функціонал під потреби конкретного підприємства [10]. Він забезпечує швидке та зручне управління кадровою інформацією, підтримує синхронізацію з іншими модулями системи та забезпечує резервне копіювання даних, що дозволяє зберігати цілісність інформації навіть при зміні структури підприємства.

Однією з ключових переваг Odoo HR є можливість інтеграції з іншими модулями платформи Odoo. Наприклад, можна налаштувати автоматичне оновлення даних про співробітників у системі бухгалтерського обліку при зміні посади чи заробітної плати. Це дозволяє підтримувати актуальність інформації про персонал у різних системах без зайвих зусиль.

Odoo HR також надає функцію автоматизації документообігу, що дозволяє генерувати та обробляти кадрові документи в електронному вигляді. Це корисно у випадках, коли підприємство має великий штат співробітників і потребує оптимізації процесів оформлення та зберігання документації.

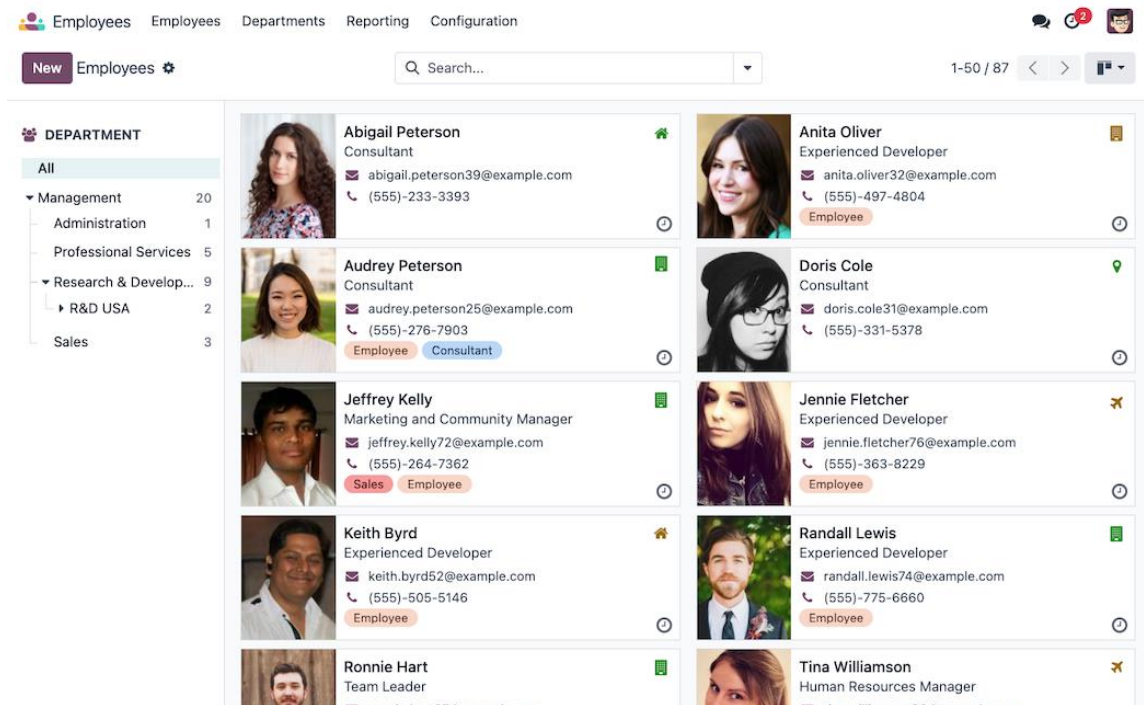


Рисунок 1.4 – Приклад огляду функціональних модулів у Odoo HR

Ці системи демонструють важливість та потенціал використання інформаційних технологій у сфері керування персоналом, пропонуючи підприємствам зручні рішення для автоматизації HR-процесів та оптимізації кадрового менеджменту. Розробка подібних систем вимагає глибокого розуміння потреб сучасних організацій, а також технічної експертизи для створення інтуїтивно зрозумілих, надійних та ефективних програмних модулів, здатних покращити точність і релевантність кадрової аналітики. У результаті аналізу аналогів була створена таблиця 1.1 для порівняння їхнього функціоналу з розроблюваною системою.

Таблиця 1.1 – Порівняльний аналіз характеристик HR-систем

	Bamboo HR	Workday HCM	PeopleForce	Odoo HR	Власна розробка
Інтеграція з іншими системами підприємства	1	1	1	1	1
Функціонал для аналітики персоналу	0	1	1	0	1
Автоматизація кадрового документообігу	0	1	0	1	1
Створення візуалізацій для аналізу персоналу	1	1	1	1	1
Можливість роботи в офлайн-режимі	0	0	1	0	1
Моделювання управлінських сценаріїв	0	1	0	0	1
Сумарний коефіцієнт	32%	84%	66%	50%	100%

Згідно з даними порівняльного аналізу, наведеними в таблиці 1.1, моя розробка модуля інформаційної системи для керування даними персоналу підприємства демонструє суттєві переваги порівняно з існуючими HR-системами, такими як BambooHR, Workday HCM, Odoo HR та PeopleForce. Моя

система досягає 100% функціональності за всіма критеріями оцінювання, тоді як аналоги мають певні обмеження. Workday HCM посідає друге місце з коефіцієнтом функціональності 84%, Odoo HR має 50%, PeopleForce — 66%, а BambooHR демонструє лише 32% необхідної функціональності.

Основними особливостями моєї реалізації є інтерактивне введення та фільтрація даних співробітників (функція `apply_filter`), оцінка ключових показників ефективності (KPI) із візуалізацією (функція `show_kpi_comparison`), прогнозування бонусів і кар'єрного зростання (функція `calculate_bonus_forecast`), моделювання управлінських сценаріїв (функція `simulate_scenarios`), генерація PDF-звітів із QR-кодами (функція `generate_pdf_report`), а також інтерактивні графіки розподілу посад, досвіду проти продуктивності та порівняння KPI (функції `show_position_chart`, `show_exp_vs_performance`). Ці можливості реалізовано за допомогою бібліотек `tkinter`, `matplotlib`, `reportlab` і `mplcursors`.

Кількісно моя система перевершує BambooHR на 68%, Odoo HR на 50%, PeopleForce на 34%, а Workday HCM на 16% за критеріями оцінювання. Ці метрики підтверджують, що моя розробка надає HR-менеджерам кращі можливості для глибокого аналізу даних, прогнозування кадрових рішень і гнучкого управління персоналом.

Отже, за результатами порівняльного аналізу можна стверджувати, що розробка власного модуля інформаційної системи для керування даними персоналу є не лише обґрунтованою, але й необхідною для забезпечення повноцінних управлінських можливостей. Реалізоване програмне забезпечення уможливорює детальний аналіз продуктивності, прогнозування кар'єрного зростання та моделювання сценаріїв завдяки повному набору функціональних характеристик, що робить його оптимальним вибором для практичного застосування в управлінні персоналом [11].

1.3 Аналіз методів розв'язання задачі

Розробка модуля інформаційної системи для керування даними персоналу підприємства вимагає комплексного підходу до вирішення завдань, пов'язаних

із введенням, зберіганням, аналізом і візуалізацією даних про співробітників. Основною задачею, яку має реалізувати програмний модуль, є можливість ефективно управляти даними персоналу, проводити оцінку продуктивності, прогнозувати кар'єрне зростання та забезпечувати інтерактивну візуалізацію результатів. Для досягнення цієї мети існують різні методи, кожен із яких має свої переваги та недоліки. Один із ключових аспектів розробки є визначення параметрів оцінки, таких як рейтинг продуктивності, досвід, освіта та відпустки, що дозволяє в подальшому аналізувати ефективність працівників і прогнозувати їхній розвиток [12].

Одним із можливих підходів є використання систем автоматизації кадрового документообігу, які базуються на шаблонних процесах для генерації звітів. Однак цей метод має певні недоліки: він вимагає значних обчислювальних ресурсів для обробки великих обсягів даних і не забезпечує гнучкості для індивідуального аналізу. Тому такий підхід не варто застосовувати в повному обсязі.

Інший метод полягає у використанні статистичних інструментів для аналізу продуктивності, таких як регресійний аналіз, який дозволяє прогнозувати показники на основі історичних даних. Цей підхід може бути корисним для оцінки продуктивності, але він чутливий до вибору параметрів і не підтримує інтерактивної візуалізації чи моделювання сценаріїв. Через це його використання обмежене.

Традиційні методи управління персоналом, такі як ручна обробка даних у табличних редакторах (наприклад, Excel), є простим і широко застосовуваним підходом. Вони базуються на сортуванні та фільтрації даних, але не здатні обробляти складні аналітичні задачі, такі як прогнозування бонусів чи порівняння KPI. Обмежена функціональність і відсутність інтерактивності роблять цей метод менш ефективним для сучасних потреб.

Ще одним методом є використання хмарних HR-систем, які пропонують ієрархічну структуру управління даними. Цей підхід дозволяє централізувати інформацію, але є обчислювально складним і не завжди підтримує офлайн-

режим, що є критичним для деяких організацій. Крім того, він не забезпечує детального аналізу чи моделювання, що обмежує його застосування.

Найефективнішим підходом є розробка власного модуля з інтерактивними аналітичними інструментами, який дозволяє моделювати дані персоналу та візуалізувати їх у реальному часі. Наприклад, функція `calculate_bonus_forecast` у моєму модулі прогнозує бонуси та кар'єрне зростання на основі рейтингу, досвіду, освіти та відпусток, а функція `show_kpi_comparison` забезпечує порівняння KPI з інтерактивними гістограмами. Цей процес передбачає збереження параметрів аналізу (наприклад, оцінки KPI чи бонусів) для подальшого звітування та візуалізації.

Для оцінки ефективності управління персоналом можна використовувати комбінацію внутрішніх та зовнішніх метрик. Внутрішні метрики, такі як середній рейтинг продуктивності чи відсоток використаної відпустки, оцінюють продуктивність працівників, тоді як зовнішні метрики, такі як порівняння з ринковими стандартами, дозволяють оцінити конкурентоспроможність. Цей підхід потребує реалізації обчислень у програмному модулі, що зменшує похибку та підвищує точність аналізу.

Під час розробки методів слід враховувати особливості даних, які можуть мати складну структуру (наприклад, різний рівень досвіду чи продуктивності). Обчислення метрик для кожного набору даних, як у функції `simulate_scenarios`, дозволяє адаптувати аналіз до специфіки підприємства та забезпечити точні результати.

Розглянувши переваги та недоліки кожного методу, було вирішено розробити модуль із інтерактивним управлінням даними, оцінкою KPI, прогнозуванням і візуалізацією, використовуючи бібліотеки `tkinter`, `matplotlib` і `reportlab`. Цей підхід забезпечує високу ефективність і гнучкість. Також було вирішено реалізувати експорт результатів у PDF (через `generate_pdf_report`) та моделювання сценаріїв, що створює унікальну платформу для HR-аналітики.

1.4 Постановка задач дослідження

Розробка модуля інформаційної системи для керування даними персоналу підприємства передбачає використання мови програмування Python у середовищі Visual Studio Code з бібліотеками tkinter, matplotlib і reportlab для створення інтерактивного графічного інтерфейсу, аналізу даних і генерації звітів. Python є високорівневою мовою з простим синтаксисом, яка широко застосовується для розробки аналітичних систем. Поєднання tkinter для інтерфейсу, matplotlib для візуалізації та reportlab для звітів дозволить створити модуль, що забезпечує ефективне управління персоналом.

Розробка модуля передбачає кілька етапів. По-перше, потрібно створити модуль графічного інтерфейсу, який забезпечить зручну взаємодію з системою для введення, фільтрації та редагування даних співробітників. Цей модуль відповідатиме за навігацію, налаштування фільтрів (наприклад, за посадою через `apply_filter`) і відображення результатів. Далі буде розроблений модуль для збереження та завантаження даних у текстових файлах (`employees`), що використовує функцію `save_employee` для запису даних і `load_employees` для їх відображення.

Наступним етапом буде модуль для оцінки продуктивності та KPI, який реалізує функцію `evaluate_kpi` для введення оцінок за критеріями ("Technical Skills", "Teamwork", "Stress Tolerance") і генерації гістограм. Модуль прогнозування, заснований на `calculate_bonus_forecast`, розраховуватиме бонуси та кар'єрне зростання на основі рейтингу, досвіду, освіти та відпусток. Модуль моделювання сценаріїв (`simulate_scenarios`) дозволить аналізувати вплив звільнень чи скорочення бонусів.

Модуль візуалізації, що використовує `show_position_chart`, `show_exp_vs_performance` і `show_kpi_comparison`, відображатиме дані у вигляді діаграм розподілу, графіків розсіювання та гістограм із підтримкою `mplcursors` для інтерактивного перегляду. Окремий модуль (`generate_pdf_report`) забезпечить експорт результатів у PDF із QR-кодами.

Розробка цих модулів потребує розуміння програмування, аналізу даних і роботи з Python. Використано tkinter для інтерфейсу, matplotlib для візуалізації, reportlab для звітів і numpy для обчислень. Тестування модулів (наприклад, коректність збереження даних і генерації звітів) забезпечить стабільність.

Задачі дослідження:

1. Розробити модуль введення та збереження даних співробітників із фільтрацією за критеріями.
2. Створити модуль оцінки продуктивності та KPI з інтерактивною візуалізацією.
3. Розробити алгоритм прогнозування бонусів і кар'єрного зростання.
4. Реалізувати модуль графічного інтерфейсу для зручного доступу до даних.
5. Забезпечити двовимірну візуалізацію даних (графіки, гістограми).
6. Автоматизувати обчислення метрик продуктивності (рейтинг, бонуси) для точного аналізу.
7. Провести тестування модулів для перевірки функціональності.

Отже, розробка модулів для інформаційної системи з використанням Python і бібліотек tkinter, matplotlib, reportlab є комплексним завданням, що включає проектування, реалізацію та тестування для ефективного управління персоналом.

1.5 Висновки

У першому розділі проведено аналіз існуючих HR-систем, таких як BambooHR, Workday HCM, PeopleForOdo, що дозволило виявити їхні переваги (стабільність, базові функції) та недоліки (обмежена інтерактивність, відсутність офлайн-режиму). Ці системи не забезпечують глибокого аналізу KPI, прогнозування кар'єрного зростання чи моделювання сценаріїв.

Порівняльний аналіз підтвердив потребу в розробці власного модуля інформаційної системи для керування даними персоналу з використанням Python у Visual Studio Code з бібліотеками tkinter, matplotlib і reportlab. Цей модуль

об'єднує найкращі функції аналогів і додає інноваційні можливості: інтерактивне введення даних, оцінку KPI з гістограмами, прогнозування бонусів, моделювання сценаріїв і генерацію PDF-звітів із QR-кодами. Ці функції, реалізовані через зручний інтерфейс, забезпечують гнучкість і ефективність для HR-управління.

2 РОЗРОБКА МОДЕЛІ ТА АЛГОРИТМІВ ПРОГРАМНОГО ПРОДУКТУ

2.1 Аналіз вхідних даних системи

Аналіз вхідних даних є критично важливим етапом у розробці модуля інформаційної системи для керування даними персоналу підприємства. Цей процес передбачає детальне вивчення інформації, яку система буде обробляти, з урахуванням її типів, структури та значення для ефективного управління персоналом. На початковому етапі необхідно визначити, які саме дані будуть збиратися, включаючи персональні дані працівників, їхню професійну активність, ключові показники ефективності (KPI), а також інші аспекти, що відображають робочі процеси та продуктивність.

Ключовим аспектом є розуміння значення кожного типу даних для оцінки ефективності працівників та прийняття управлінських рішень. Наприклад, дані про досвід роботи, освіту чи показники продуктивності можуть свідчити про професійний рівень співробітника, його потенціал для кар'єрного зростання або потребу в додатковому навчанні. Водночас інформація про кількість використаних відпусткових днів або оцінки KPI може вказувати на баланс між роботою та відпочинком, що впливає на загальну продуктивність. Для правильної інтерпретації цих даних система повинна включати алгоритми, які дозволяють категоризувати та аналізувати інформацію, наприклад, оцінювати KPI за критеріями технічних навичок, командної роботи чи стресостійкості, як реалізовано в методі `evaluate_kpi` коду.

Основними типами вхідних даних для модуля інформаційної системи є:

1. Персональні дані працівників: До них належать ім'я, посада, досвід роботи, освіта, кількість відпусткових днів та оцінка продуктивності. Ці дані, які зберігаються в текстових файлах у директорії `EMPLOYEE_DIR`, є основою для ідентифікації працівників та персоналізації їхніх профілів. Наприклад, метод `save_employee` забезпечує збереження цих даних у структурованому форматі.

2. Дані про продуктивність: Оцінка продуктивності (рейтинг від 0 до 100) та KPI, які включають такі критерії, як технічні навички, командна робота та стресостійкість, дозволяють оцінити внесок працівника в діяльність підприємства. Метод `show_kpi_comparison` створює порівняльні графіки для аналізу KPI між працівниками, що допомагає виявити сильні та слабкі сторони.

3. Дані про робочі процеси: Інформація про посади, кількість працівників на кожній посаді та прогнози кар'єрного розвитку допомагає оцінити структуру персоналу та планувати кадрові зміни.

4. Симуляційні дані: Дані, отримані в результаті моделювання сценаріїв, таких як скорочення бонусів чи звільнення працівників, дозволяють прогнозувати вплив управлінських рішень на загальну продуктивність і бюджет підприємства.

Оскільки система працює з персональними даними, такими як імена, посади та оцінки продуктивності, забезпечення їхньої безпеки є пріоритетом. Хоча код не містить явних механізмів шифрування, важливо передбачити захист від несанкціонованого доступу, наприклад, шляхом впровадження автентифікації користувачів та шифрування файлів у директорії `EMPLOYEE_DIR`. Це особливо актуально для звітів у форматі PDF, які генеруються методом `generate_pdf_report` і містять конфіденційну інформацію.

Ретельний аналіз вхідних даних дозволяє виявити потреби працівників і підприємства, а також оптимізувати функціонал системи. Наприклад, методи `show_exp_vs_performance` та `show_kpi_comparison` створюють візуальні звіти, які допомагають HR-менеджерам оцінити залежність продуктивності від досвіду або порівняти KPI між працівниками. Ці аналітичні інструменти сприяють прийняттю обґрунтованих рішень щодо розподілу бонусів, планування кар'єрного зростання чи реорганізації штату.

Аналіз вхідних даних забезпечує основу для створення ефективного модуля інформаційної системи, який підтримує управління персоналом. Процес збору, категоризації та інтерпретації даних, реалізований у таких функціях, як збереження профілів працівників, оцінка KPI та генерація звітів, дозволяє

оптимізувати кадрові процеси, підвищувати продуктивність і покращувати користувацький досвід. Завдяки інтеграції аналітичних інструментів, таких як графіки та симуляції, система стає потужним інструментом для стратегічного управління персоналом підприємства.

2.2 Моделювання архітектури системи

Розробка модуля інформаційної системи для управління даними персоналу підприємства ґрунтується на принципах програмної інженерії, які забезпечують модульність, ефективну обробку даних і гнучкість системи. Модульна архітектура, де кожен компонент відповідає за певну функцію, таку як введення даних, їх аналіз, прогнозування чи створення звітів, сприяє масштабованості та легкості підтримки. Ці принципи є ключовими для роботи з комплексними наборами даних про персонал, генерації аналітичних звітів і створення візуалізацій, що забезпечують надійність і можливість розширення функціоналу. Система, інтегрує обробку даних, аналітику та виведення результатів для підтримки завдань управління персоналом.

Програма, розроблена на Python, використовує модульну архітектуру з інтеграцією бібліотек `tkinter` для створення інтерфейсу, `matplotlib` для візуалізації, `reportlab` для генерації PDF-звітів і стандартних модулів, таких як `os` і `collections`.

Робота системи починається з введення даних працівника через інтерфейс, побудований на `tkinter`. Введені дані перевіряються на валідність і зберігаються у текстових файлах у директорії `EMPLOYEE_DIR`. Далі дані доступні для фільтрації, аналізу чи прогнозування: користувач може вибрати працівника зі списку, оцінити його KPI, згенерувати звіт або створити графік. Оброблені результати спрямовуються до модуля візуалізації для відображення графіків або до модуля звітів для створення PDF-документів. Центральний клас `HRBonusApp` координує взаємодію між модулями, забезпечуючи цілісність і послідовність обробки даних.

Блок-схему загальної роботи системи зображено на рисунку 2.1.



Рисунок 2.1 – Блок-схема загальної роботи системи

Після початкової обробки даних система агрегує результати для комплексного аналізу персоналу. Обчислювальний модуль зберігає проміжні результати, такі як оцінки KPI, розрахунки бонусів і прогнози кар'єрного розвитку, дозволяючи відстежувати історію обробки даних. Ці результати доступні для детального аналізу через методи, такі як бонус і КПІ, які забезпечують збереження й обробку даних для подальшого використання.

Модуль візуалізації, побудований на базі бібліотеки `matplotlib`, генерує різноманітні графіки для ілюстрації даних на різних етапах. Метод створює кругову діаграму розподілу працівників за посадами, `show_exp_vs_performance` відображає залежність продуктивності від досвіду, а `show_kpi_comparison` порівнює КРІ між працівниками. Ці візуалізації супроводжуються числовими підсумками, такими як середні оцінки КРІ чи розміри бонусів, які кількісно оцінюють продуктивність і зберігаються разом із графіками для інтегрованого аналізу.

Для порівняльного аналізу система підтримує альтернативні сценарії, реалізовані в методі симуляція. Наприклад, вона може моделювати звільнення працівників чи скорочення бонусів, створюючи паралельні результати, які порівнюються з основними даними. Ці результати відображаються у текстовому вигляді в інтерфейсі і можуть бути використані для оцінки впливу управлінських рішень.

2.3 Алгоритм прогнозування кар'єрного росту

На рисунку 2.2 блок-схемі представлено логіку роботи модуля керування персоналом, який реалізує функціональність системи HR-менеджменту підприємства. Основна ідея підходу полягає у комплексному обліку даних співробітників та автоматизації процесів розрахунку бонусів і прогнозування кар'єрного розвитку на основі ключових показників ефективності.

Функціональність системи розпочинається з перевірки наявності файлів працівників в директорії. Якщо база даних співробітників порожня, система пропонує створити нові записи для подальшої роботи. У випадку, коли дані про співробітників наявні, система дозволяє здійснювати фільтрацію за посадами, розраховувати бонуси на основі досвіду роботи та оцінки продуктивності, а також формувати звіти з різноманітною аналітичною інформацією.

Якщо рейтинг співробітника перевищує 90 балів і досвід роботи складає більше 5 років, система прогнозує високу ймовірність підвищення. При рейтингу від 80 до 90 балів і досвіді від 3 до 5 років - прогноз "підвищення ймовірне". Для

співробітників з рейтингом від 60 до 80 балів система пропонує прогноз "стабільна позиція", а для тих, хто має нижчий рейтинг - "потрібне вдосконалення".

Загалом, розроблена система демонструє комплексний підхід до автоматизації HR-процесів на підприємстві, дозволяючи ефективно управляти персоналом, здійснювати справедливий розподіл бонусів та прогнозувати кар'єрний розвиток співробітників на основі об'єктивних показників. Вона сприяє підвищенню прозорості управлінських рішень і зменшенню впливу людського фактора на процес оцінювання. Крім того, система забезпечує гнучкість у налаштуванні під потреби конкретного підприємства, що робить її універсальним інструментом для HR-відділу.

Модуль використовує чітку формулу для розрахунку бонусів, яка враховує кілька факторів:

$$B = (3000 + 40 \cdot R) \cdot \left(1 + \frac{E}{10}\right) + \delta \text{ edu} \cdot 300 - 10 \cdot \max(0, V - 20) \quad (2.1)$$

B – розмір бонусу в гривнях; R – рейтинг продуктивності працівника, отриманий з оцінки його ефективності; E – досвід роботи в роках, що відображає стаж працівника; V – кількість використаних відпусткових днів за рік; $\max(0, V - 20)$ – функція, яка обчислює штраф за перевищення ліміту в 20 відпусткових днів.

Ця формула забезпечує справедливий розподіл бонусів, додаючи компоненти за продуктивність і освіту, коригуючи за досвід і штрафуючи за надмірні відпустки. Результати розрахунків виводяться у форматі {bonus:.2f}, що полегшує їх сприйняття.

Інтеграція модуля з інтерфейсом, побудованим на tkinter, дозволяє відображати результати у зрозумілій таблиці, де кожен рядок містить ім'я працівника, розмір бонусу та прогноз кар'єрного розвитку. Це забезпечує HR-менеджерам швидкий доступ до аналітичної інформації для прийняття рішень. Модуль також підтримує обробку винятків, наприклад, пропускаючи файли з некоректними даними, що підвищує його надійність.

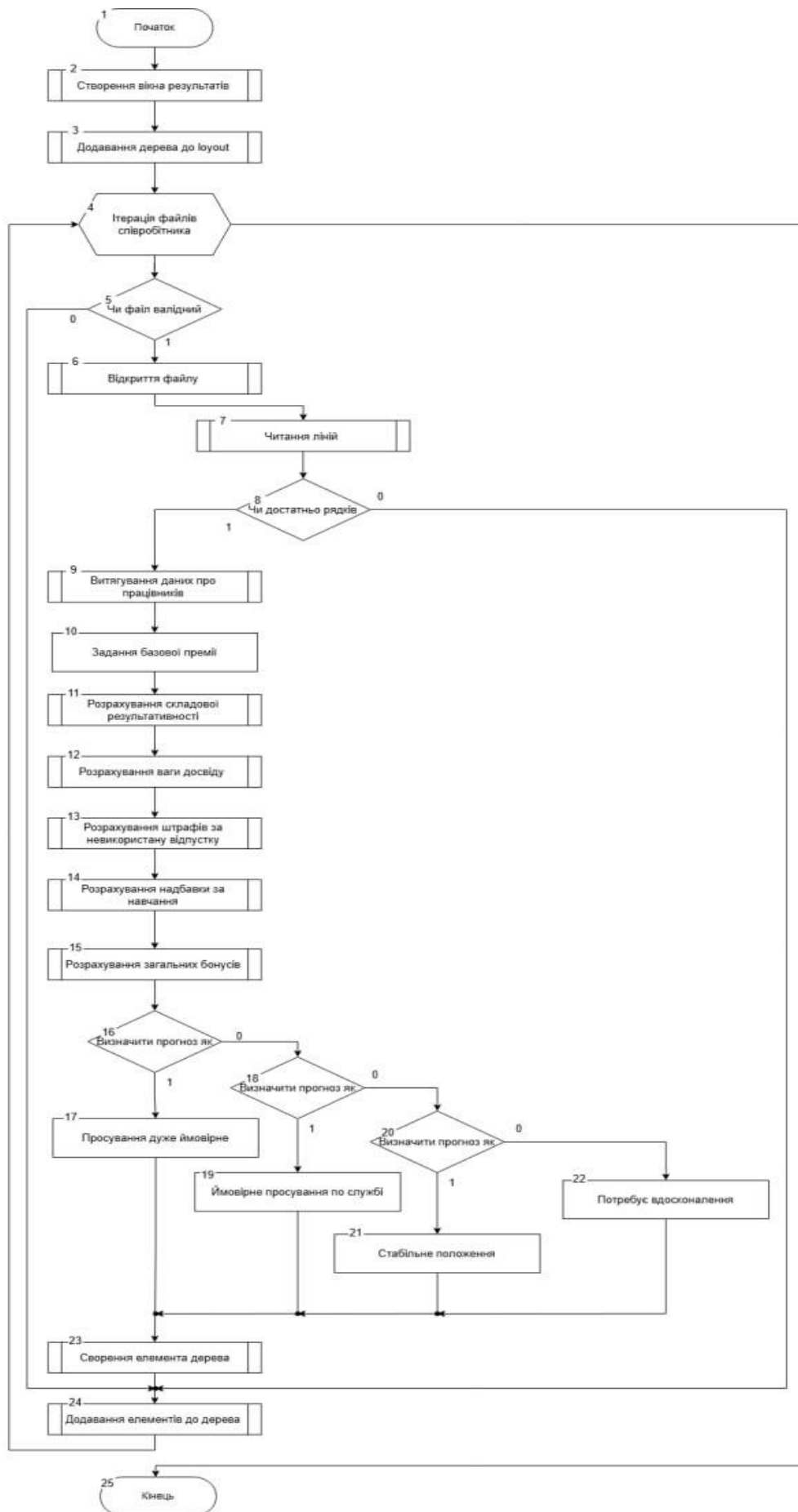


Рисунок 2.2 – Алгоритм роботи модуля calculate_bonus_forecast

Модуль `calculate_bonus_forecast` є прикладом ефективного поєднання простоти та функціональності. Його ізольована структура дозволяє легко модифікувати параметри розрахунків, такі як базовий бонус чи вагові коефіцієнти, для адаптації до специфічних вимог підприємства. Завдяки цьому модуль не лише автоматизує HR-процеси, але й сприяє стратегічному плануванню кар'єрного розвитку працівників, підвищуючи продуктивність управління персоналом.

2.4 Розробка симуляції управлінських сценаріїв

На рисунку 2.3 зображена блок-схема, яка ілюструє логіку роботи модуля симуляції управлінських сценаріїв. Цей модуль є важливим компонентом системи HR-менеджменту підприємства, дозволяючи HR-аналітикам оцінювати наслідки управлінських рішень шляхом моделювання різних сценаріїв, таких як звільнення працівників або скорочення бонусів. Метод забезпечує кількісний аналіз впливу цих рішень на продуктивність і бюджет підприємства, сприяючи обґрунтованому стратегічному плануванню.

Функція `simulate_scenarios` призначена для моделювання двох основних сценаріїв: звільнення одного чи двох працівників із найнижчим рейтингом продуктивності та скорочення бонусів на 20%. Модуль працює з даними працівників, що зберігаються в директорії `EMPLOYEE_DIR`, і використовує рейтинг продуктивності для аналізу. Основні етапи роботи модуля:

1. Сценарій 1: Звільнення працівників:
 - Визначаються один або два працівники з найнижчим рейтингом продуктивності.
 - Розраховується середній рейтинг продуктивності до і після умовного звільнення:
 - Середній рейтинг обчислюється як сума всіх рейтингів, поділена на кількість працівників.
 - Після видалення працівника(ів) середній рейтинг перераховується для решти працівників.

– Результати включають імена звільнених працівників і зміну середнього рейтингу, що допомагає оцінити вплив звільнення на загальну продуктивність.

2. Сценарій 2: Скорочення бонусів:

– Використовується метод `calculate_bonus_forecast` для обчислення бонусів усіх працівників.

– Моделюється скорочення бонусів на 20% для кожного працівника.

– Розраховується загальна економія бюджету як різниця між початковою сумою бонусів і сумою після скорочення.

– Результати включають початкову суму бонусів, суму після скорочення та зекономлені кошти.

Формули для сценаріїв

- Середній рейтинг продуктивності:

$$A = \frac{\left(\sum_{i=1}^n R_i\right)}{n} \quad (2.2)$$

A – середній рейтинг продуктивності; R_i [0,100] – рейтинг продуктивності; n – кількість працівників.

Економія від скорочення бонусів:

$$S = \sum_{i=1}^n B_i - \sum_{i=1}^n (0.8 \times B_i) = 0.2 \times \sum_{i=1}^n B_i \quad (2.3)$$

S – економія від скорочення бонусів; B_i – початковий бонус, i -го працівника, за формолою 2.1.

Модуль `simulate_scenarios` є потужним інструментом для HR-аналітиків, дозволяючи моделювати наслідки управлінських рішень, таких як звільнення працівників або скорочення бонусів.

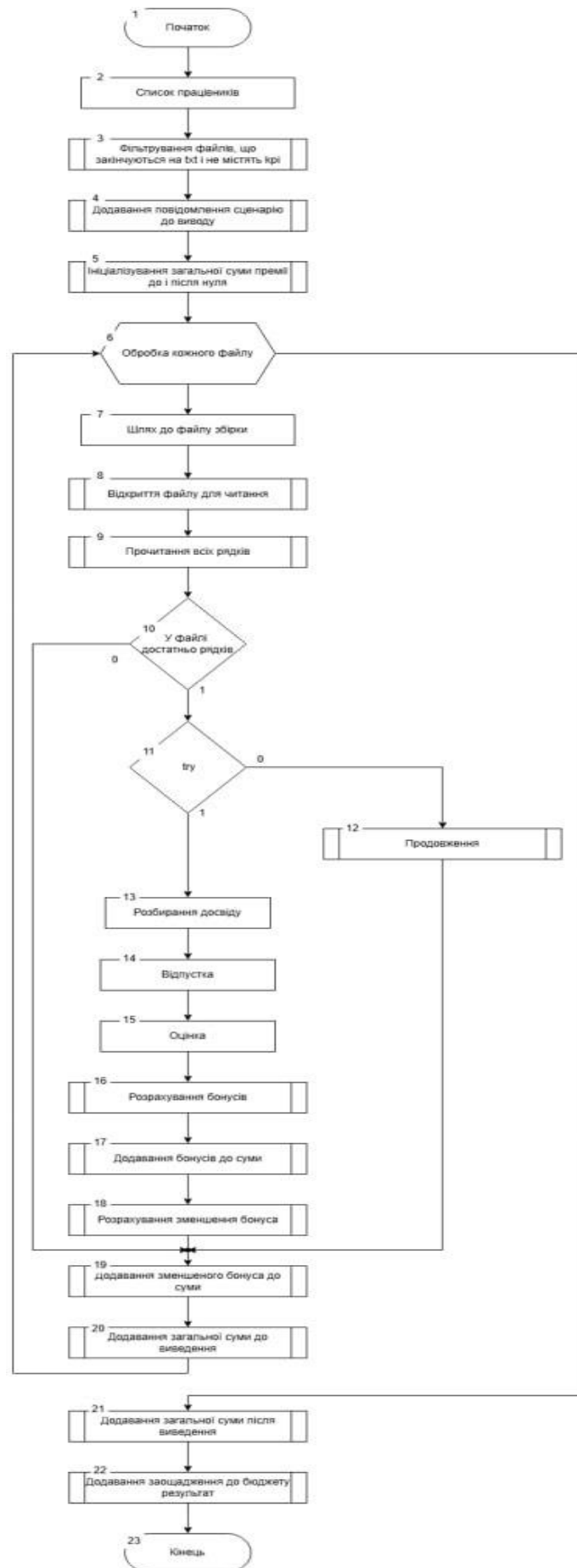


Рисунок 2.3 – Алгоритм роботи модуля simulate_scenarios

2.5 Розробка графічної схеми інтерфейсу

Для створення естетичного та функціонального дизайну модуля інформаційної системи для керування даними персоналу підприємства особлива увага приділяється розробці інтерфейсу. Використання графічних схем дозволяє візуалізувати структуру та зовнішній вигляд інтерфейсу, забезпечуючи зручність, інтуїтивність і привабливий вигляд користувацького середовища. Графічний дизайн відіграє ключову роль у формуванні інтерфейсу, який полегшує взаємодію користувача з системою, тому кожна деталь і елемент вікна програми ретельно продумані, як це реалізовано в коді.

Проектування інтерфейсу модуля інформаційної системи для керування даними персоналу підприємства зосереджено на створенні зручного, інтуїтивного та естетично привабливого середовища. Графічна схема головного вікна, поділена на вкладки для введення даних і аналізу, список працівників, фільтри та аналітичні інструменти, відповідає потребам HR-менеджерів. Реалізація через `tkinter`, `matplotlib` і `reportlab` забезпечує чітку структуру, інтерактивність і професійний вигляд, що сприяє ефективній взаємодії з системою.

Ключовим елементом дизайну є інтерактивність і адаптивність інтерфейсу, що забезпечують комфортну роботу з даними. Наприклад, використання `ttk.Notebook` дозволяє легко перемикатися між вкладками для введення даних і аналітичних функцій, а інтерактивні графіки, створені за допомогою `matplotlib` з підтримкою `mplcursors`, надають HR-менеджерам можливість детального аналізу даних шляхом наведення на елементи діаграм. Система також включає впливаючі повідомлення для зворотного зв'язку, що підвищує зручність і запобігає помилкам під час роботи.

На рисунку 2.4 показано схематичне зображення головного вікна системи з пронумерованими окремими секціями, і описано, за що кожна секція відповідає.

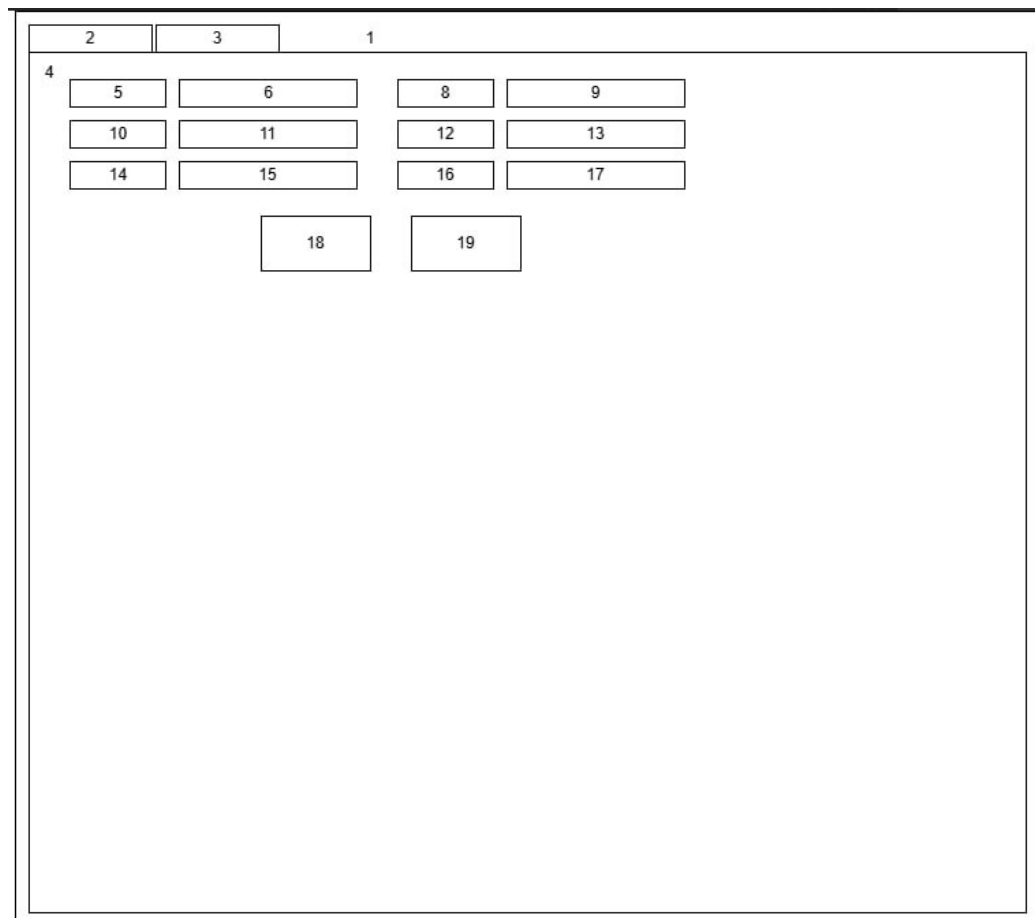


Рисунок 2.4 – Графічна схема інтерфейсу головного вікна системи

1. Головне вікно програми.
2. Вкладка «Введення даних».
3. Вкладка «Управління та аналіз»
4. Частина вікна, що відповідає за візуалізацію даних співробітника
5. Текст «повне Ім'я».
6. Поле для вводу повного імені працівника.
7. Текст «Стаж роботи в роках».
8. Поле для вводу стажу роботи в роках.
9. Текст «Посада працівника».
10. Поле для вводу посади працівника.
11. Текст «Освіта працівника».
12. Поле для вводу освіти працівника.
13. Текст «Відпускні дні».

14. Поле для вводу кількості відпускних днів.
15. Текст «Рейтинг продуктивності від 0 до 100».
16. Поле для вводу рейтингу продуктивності від 0 до 100.
17. Кнопка збереження даних працівника.
18. Кнопка очищення форми.

На рисунку 2.5 показано схематичне зображення вікна програми «Управління та аналіз».

Рисунку 2.5 показує схематичне зображення вікна програми «Управління та аналіз». Вікно має наступну структуру:

- 1. Заголовок вікна.
- 2. Вкладка «Введення даних».
- 3. Вкладка «Управління та аналіз».
- 4. Контейнер для контенту вкладки «Управління та аналіз».
- 5. Заголовок контейнера 4.
- 6. Поле для вводу кількості відпускних днів.
- 7. Поле для вводу рейтингу продуктивності.
- 8. Кнопка збереження даних працівника.
- 9. Кнопка очищення форми.
- 10. Контейнер для таблиці.
- 11-18. Елементи таблиці (11, 12, 13, 14, 15, 16, 17, 18).
- 19. Велике поле для виводу інформації.

Рисунок 2.5 – Графічна схема інтерфейсу управління та аналіз

1. Головне вікно програми.
2. Вкладка «Введення даних».
3. Вкладка «Управління та аналіз».
4. Поле вкладки «Управління та аналіз».

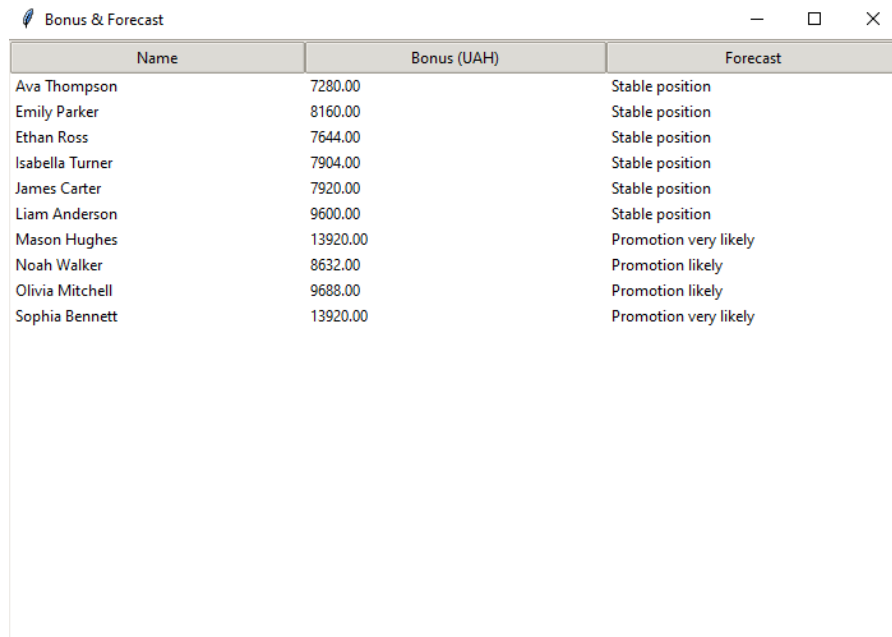
5. Частина вікна, що відповідає за роботу пошуку співробітника.
6. Текст «Фільтрація за посадою»
7. Поле для фільтрації за посадою.
8. Кнопка застосування фільтру.
9. Кнопка відображення всіх працівників.
10. Частина вікна, що відповідає за аналіз програми.
11. Кнопка генерації звіту.
12. Кнопка розрахунку бонусів та прогнозування кар'єрного розвитку.
13. Кнопка відображення діаграми розподілу посад.
14. Кнопка оцінки КРІ працівника.
15. Кнопка симуляції сценаріїв змін у структурі персоналу.
16. Кнопка видалення файлу працівника.
17. Кнопка відображення графіка продуктивності працівників.
18. Кнопка відображення діаграми КПІ працівників.
19. Область відображення списку файлів працівників.

На рисунку 2.6 зображено інтерфейс модуля нарахування бонусів та прогнозування кар'єрного розвитку працівників. Даний модуль є важливою складовою розробленої інформаційної системи для керування персоналом підприємства.

Вікно "Bonus & Forecast" відображає таблицю з трьома стовпцями: «Name», «Bonus» та «Forecast». Система автоматично розраховує бонуси для кожного співробітника на основі закладених алгоритмів, які враховують продуктивність, досвід роботи, освіти та інші показники.

У таблиці представлено дані щодо 10 працівників. Розмір бонусів коливається від 7280.00 до 13920.00 гривень. Система також надає прогноз щодо кар'єрного розвитку кожного працівника. Для більшості співробітників (Ava Thompson, Emily Parker, Ethan Ross, Isabella Turner, James Carter та Liam Anderson) визначено статус «Stable position». Для Noah Walker та Olivia Mitchell система прогнозує «Promotion likely», а для Mason Hughes та Sophia Bennett – «Promotion very likely».

Модуль автоматизує процес оцінки персоналу та надає HR-спеціалістам важливу інформацію для прийняття управлінських рішень щодо матеріального заохочення та кар'єрного планування співробітників.



Name	Bonus (UAH)	Forecast
Ava Thompson	7280.00	Stable position
Emily Parker	8160.00	Stable position
Ethan Ross	7644.00	Stable position
Isabella Turner	7904.00	Stable position
James Carter	7920.00	Stable position
Liam Anderson	9600.00	Stable position
Mason Hughes	13920.00	Promotion very likely
Noah Walker	8632.00	Promotion likely
Olivia Mitchell	9688.00	Promotion likely
Sophia Bennett	13920.00	Promotion very likely

Рисунок 2.6 – Інтерфейс модуля нарахування бонусів та прогнозування кар'єрного розвитку працівників

На рисунку 2.7 продемонстровано результат використання модуля «Simulate» у розробленій системі управління персоналом підприємства. Цей модуль дозволяє HR-менеджерам моделювати різні сценарії змін у структурі персоналу та їх фінансові наслідки.

Вікно «Workforce Simulation» відображає результати двох змодельованих сценаріїв. Перший сценарій моделює ситуацію, коли один працівник залишає компанію. Система автоматично розраховує, що середній рейтинг продуктивності персоналу після такого скорочення становитиме 88.44 бали зі 100 можливих.

Другий сценарій демонструє наслідки скорочення бонусних виплат на 20%. Система показує, що загальна сума бонусів до скорочення складала 47345.00 гривень, а після впровадження змін становитиме 37876.00 гривень. Економія бюджету компанії при цьому становитиме 9469.00 гривень.

Внизу вікна розташовані кнопки, які дозволяють моделювати різні сценарії: «Simulate 1 Resignation», «Simulate 2 Resignations» та «Simulate 20% Bonus Cut».

Цей інструмент надає керівництву компанії можливість прогнозувати наслідки кадрових змін та фінансових рішень, що сприяє прийняттю обґрунтованих управлінських рішень та оптимізації витрат на персонал.

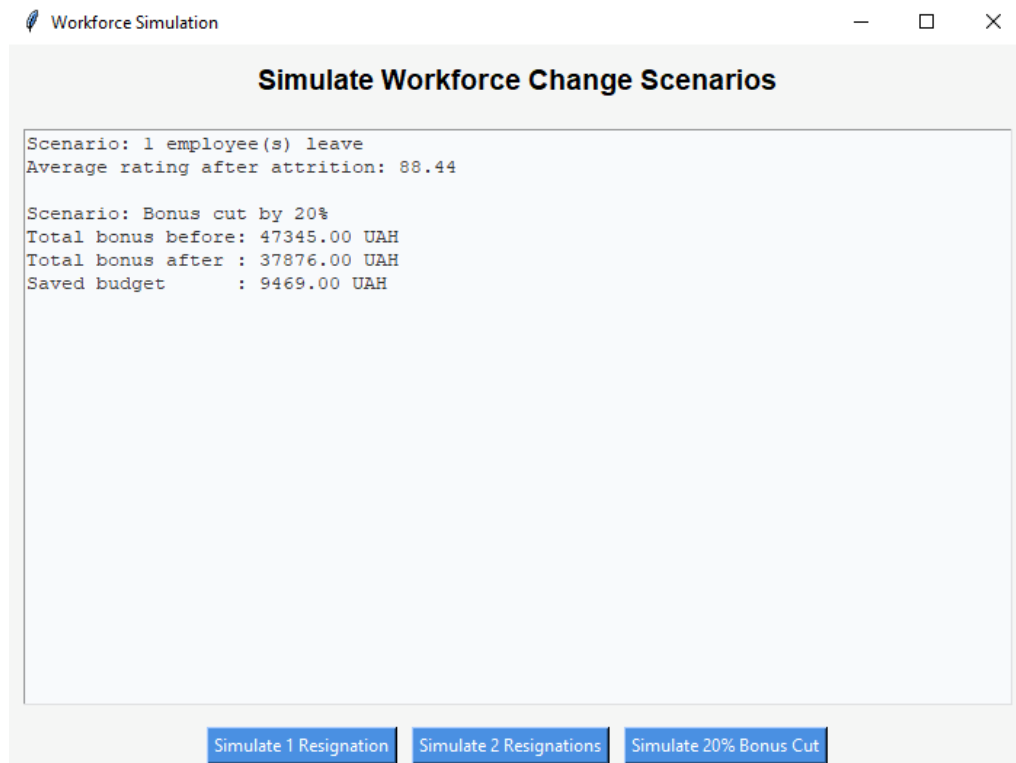


Рисунок 2.7 – Інтерфейс модуля симуляції сценаріїв змін у структурі персоналу

Розробка даної інформаційної системи здійснювалася за допомогою мови програмування Python у інтегрованому середовищі розробки Visual Studio Code. Для реалізації функціоналу були використані бібліотеки Tkinter для створення графічного інтерфейсу, matplotlib для візуалізації даних, reportlab для генерації PDF-звітів, а також стандартні бібліотеки os, io та collections для роботи з файловою системою та обробки даних. Імплементация системи дозволяє не лише оцінювати поточний стан компетенцій працівника, але й відстежувати динаміку змін цих показників з часом.

2.6 Висновки

У другому розділі проведено аналіз і розробку модульної архітектури інформаційної системи для керування даними персоналу підприємства, що забезпечує ефективну обробку даних, прогнозування та візуалізацію результатів. Реалізовано ключові алгоритми, зокрема для оцінки продуктивності, прогнозування кар'єрного розвитку та моделювання управлінських сценаріїв, підкріплені математичними формулами для розрахунку бонусів і середніх показників. Було створено підсистему управління даними, яка використовує текстові файли в директорії EMPLOYEE_DIR для збереження інформації про працівників, включаючи ім'я, посаду, досвід, освіту, відпусткові дні та рейтинг продуктивності.

Розроблено підсистему інтерактивної візуалізації, що відображає аналітичні результати через кругові діаграми, діаграми розсіювання та порівняльні гістограми KPI. Використання бібліотеки matplotlib із підтримкою mplcursors забезпечує інтерактивність, дозволяючи переглядати деталі даних при наведенні на елементи графіків. Графічний інтерфейс, побудований на tkinter, об'єднує введення даних, фільтрацію, оцінку KPI, прогнозування бонусів і моделювання сценаріїв через зручні вкладки, що полегшує навігацію та аналіз.

Сформовано цілісну систему для управління персоналом, яка поєднує обчислювальну точність із зручністю використання. Модульна структура сприяє легкій інтеграції нових функцій, таких як додаткові методи аналізу чи експорт даних у різні формати. Зокрема, реалізовано генерацію PDF-звітів із QR-кодами для швидкого доступу до інформації. Система створює міцну основу для практичного застосування в HR-менеджменті, забезпечуючи гнучкість, наочність і підтримку стратегічного планування кадрових рішень.

3 . РОЗРОБКА ПРОГРАМНИХ МОДУЛІВ РЕАЛІЗАЦІЇ СИСТЕМИ

3.1 Аналіз і обґрунтування вибору засобів для реалізації системи

Розробка модуля інформаційної системи для керування даними персоналу підприємства потребує ретельного вибору засобів, які забезпечують ефективне виконання завдань, модульність і зручність використання. Метою системи є створення платформи для введення, обробки, аналізу та візуалізації даних працівників, а також генерації звітів і прогнозування управлінських рішень. Для цього використано мову програмування Python, бібліотеки tkinter, matplotlib, reportlab і стандартні модулі Python, такі як os і collections. Вибір кожного інструменту ґрунтувався на його продуктивності, функціональних можливостях, сумісності з іншими компонентами та підтримці спільноти.

Python обрано завдяки його універсальності, зрозумілому синтаксису та багатій екосистемі бібліотек для обробки даних і створення інтерфейсів. Ця мова забезпечує швидке створення прототипів, полегшує підтримку коду та підтримує модульну організацію, що критично важливо для системи управління персоналом. Python має розвинену спільноту, велику кількість документації та сумісність із бібліотеками, такими як tkinter для інтерфейсу та matplotlib для візуалізації. У програмі Python об'єднує всі компоненти – від введення даних до генерації звітів [13].

Бібліотека tkinter [14] обрана для створення графічного інтерфейсу завдяки її простоті, вбудованій підтримці в Python і достатньому набору віджетів для реалізації функціоналу системи. Вона забезпечує кросплатформність, дозволяючи запускати систему на різних операційних системах без значних змін. У коді tkinter використовується для створення вкладок ttk.Notebook, полів введення ttk.Entry, списку працівників Listbox і кнопок для аналітичних функцій. Інтеграція з matplotlib через методи, такі як show_position_chart, дозволяє вбудовувати інтерактивні графіки безпосередньо в інтерфейс. У порівнянні з PyQt5, який є складнішим і потребує додаткових залежностей, або wxPython, що

має обмежені можливості дизайну, tkinter пропонує оптимальний баланс між простотою та функціональністю для настільної системи.

Бібліотека matplotlib [15] обрана для створення інтерактивних графіків завдяки її гнучкості та здатності генерувати різноманітні візуалізації, такі як кругові діаграми, діаграми розсіювання і порівняльні гістограми. Вона інтегрується з tkinter, дозволяючи відображати графіки в контексті інтерфейсу, і підтримує інтерактивність через mplcursors для додавання підказок при наведенні. Matplotlib має активну спільноту, численні приклади та детальну документацію, що полегшує вирішення технічних питань. У порівнянні з Seaborn, який менш гнучкий для кастомізації, або Plotly, орієнтованим на веб-додатки, matplotlib краще підходить для настільних систем із локальними візуалізаціями.

Бібліотека reportlab [16] обрана для створення PDF-звітів завдяки її здатності генерувати структуровані документи з текстом, таблицями та графікою. У методі вона використовується для формування звітів із персональними даними, оцінками KPI та QR-кодами, що забезпечує професійний вигляд документів. Reportlab підтримує точне налаштування формату, включаючи шрифти, кольори та розташування елементів, що відповідає потребам системи. Альтернативи, такі як PyPDF2, більше підходять для редагування існуючих PDF, а не для створення нових, тоді як reportlab ідеально підходить для генерації звітів із нуля.

Стандартні модулі Python, такі як os і collections [17], використовуються для роботи з файловою системою та обробки даних. Модуль os забезпечує управління файлами працівників у директорії EMPLOYEE_DIR, а collections.Counter застосовується для підрахунку посад у show_position_chart. Ці модулі є легкими, швидкими та вбудованими в Python, що усуває потребу в додаткових залежностях. Альтернативи, такі як зовнішні бібліотеки для роботи з файлами, додали б непотрібну складність.

Вибір Python, tkinter, matplotlib, reportlab, os і collections зумовлений їхньою відповідністю задачам системи:

- Python забезпечує універсальну платформу для інтеграції всіх компонентів, від обробки даних до інтерфейсу.
- Tkinter пропонує простий і ефективний спосіб створення кросплатформного інтерфейсу з вкладками та аналітичними функціями.
- Matplotlib підтримує гнучку візуалізацію даних працівників із можливістю інтерактивного аналізу.
- Reportlab дозволяє створювати структуровані PDF-звіти, які відповідають професійним стандартам.
- Os і collections забезпечують швидку й ефективну роботу з файлами та даними.

Ці інструменти формують цілісну систему, яка відповідає потребам управління персоналом, забезпечуючи модульність, продуктивність і зручність використання. У порівнянні з альтернативними засобами обрані бібліотеки пропонують оптимальне поєднання простоти, функціональності та підтримки спільноти, що робить їх ідеальними для реалізації модуля інформаційної системи [18].

3.2 Розробка модуля управління персональними файлами співробітників

Блок-схема на рисунку 3.1 описує процес роботи HR-системи, яка завантажує дані співробітників із файлів у директорії employees, застосовує фільтр за посадою для вибірки потрібних записів, відображає детальну інформацію про вибраного співробітника (ім'я, посада, досвід, освіта, відпустка, рейтинг) у відповідних полях інтерфейсу та перевіряє наявність файлу KPI для нього, виводячи ці показники у вигляді графіка чи звіту за їх наявності, або повертається до обробки наступного запису, якщо KPI відсутні. Система забезпечує автоматичне створення директорії для зберігання файлів, дозволяє записувати та оновлювати інформацію про співробітників, підтримує видалення файлів.

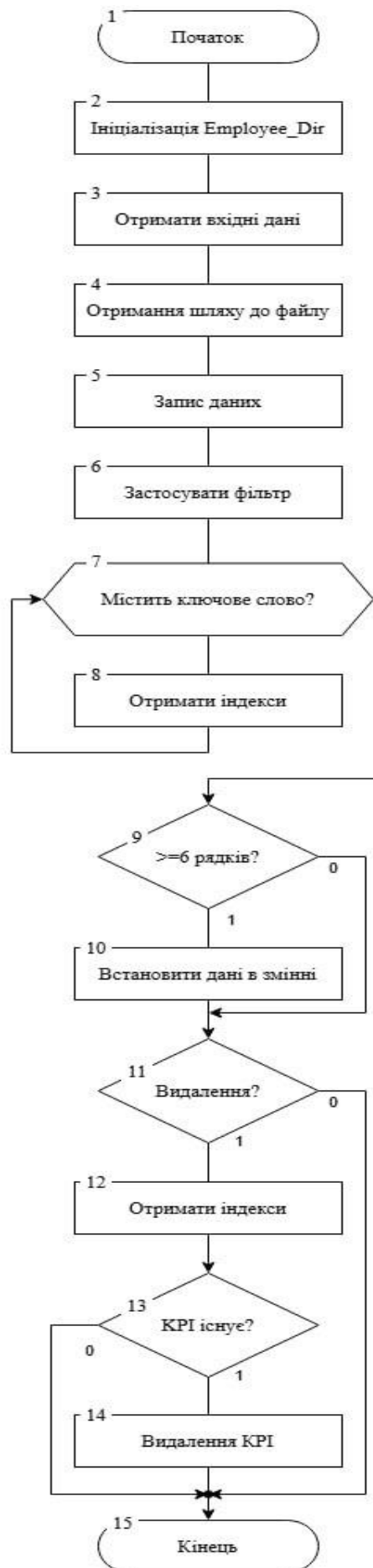


Рисунок 3.1 – Робота класу Employee

Блок-схема на рисунку 3.2 описує детальний процес обробки даних у HR-системі, який охоплює завантаження інформації про співробітників із файлів у директорії, перевірку наявності та повноти цих даних, а також подальше опрацювання або виведення результатів залежно від певних умов. Процес розпочинається з ініціалізації системи, після чого виконується зчитування вихідних даних про співробітників. Далі перевіряється, чи дані існують і чи є достатньо інформації, і за відсутності даних передбачено можливість їх введення або додавання нових записів. Якщо дані присутні, система аналізує їх, зокрема перевіряючи наявність файлу, і за результатами цього аналізу вирішується, чи виводити дані для подальшої роботи, чи завершити процес. Уся логіка супроводжується циклічним поверненням до зчитування даних для забезпечення безперервної обробки, що дозволяє адаптуватися до змін у базі даних.

Вона також підкреслює важливість гнучкості системи у відповідь на різні сценарії роботи з даними. Наприклад, якщо під час перевірки виявляється, що файл даних відсутній або пошкоджений, система пропонує користувачу ввести нові дані вручну, що забезпечує безперервність роботи та мінімізує ризик втрати інформації. Ця функція особливо корисна в умовах динамічного оновлення бази співробітників, коли можуть виникати помилки через видалення чи переміщення файлів. Крім того, після успішного аналізу даних система може автоматично генерувати звіти або візуалізації, такі як графіки продуктивності, що полегшує прийняття рішень менеджерами HR.

Наступним важливим аспектом є інтеграція циклічного підходу, який гарантує, що система постійно моніторить стан даних і готова до обробки нових або оновлених записів. Такий підхід дозволяє підтримувати актуальність інформації в реальному часі, що є критичним для оперативного управління персоналом.

Процес також підтримує інтерактивність, дозволяючи користувачам оперативно реагувати на зміни, наприклад, через графічний інтерфейс, де можна легко оновлювати дані чи запускати аналітичні функції

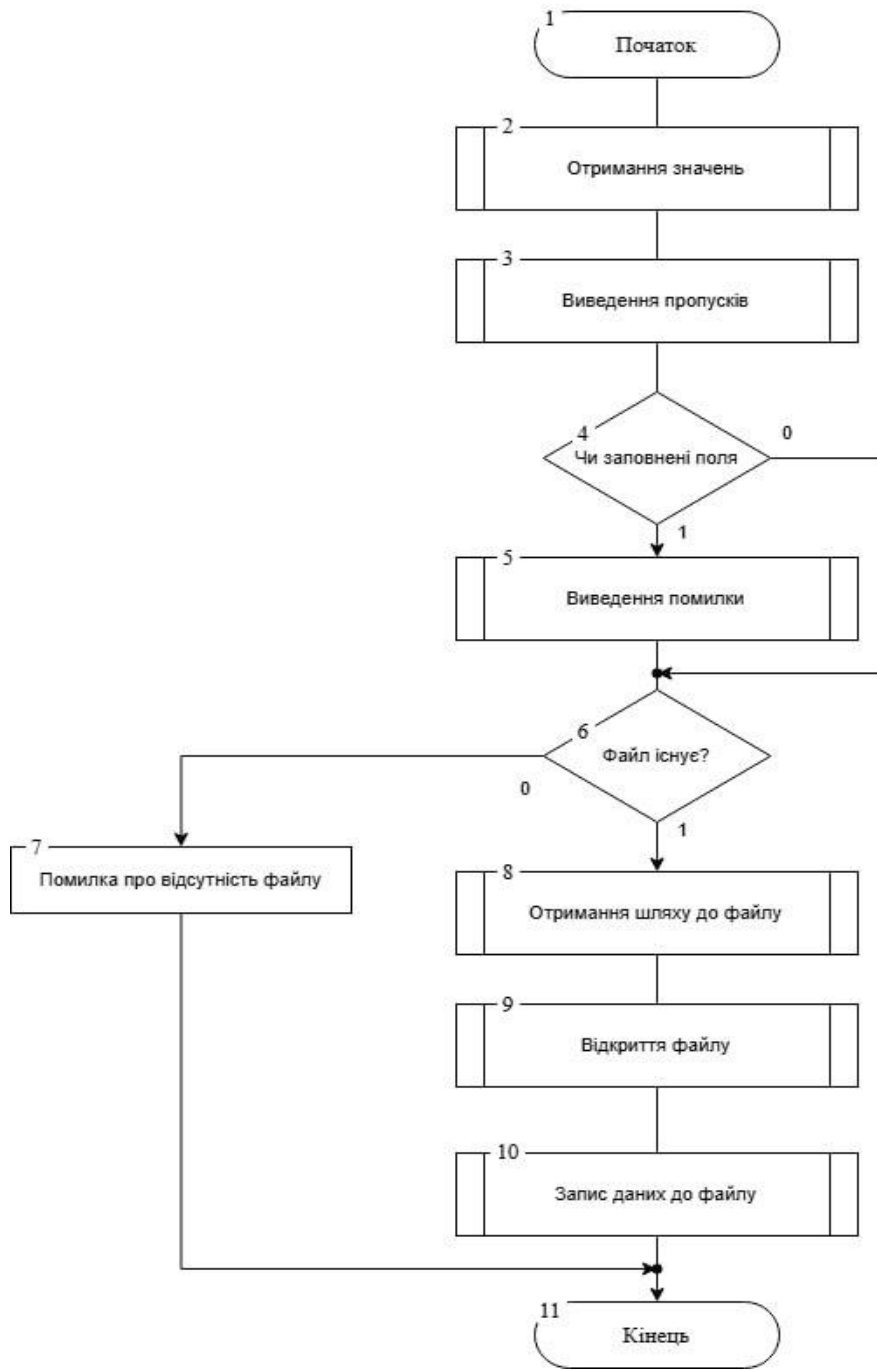


Рисунок 3.2 – Робота модуля save_employee

Блок-схема на рисунку 3.3 описує процес обробки даних у HR-системі, який охоплює ініціалізацію, завантаження інформації про співробітників із файлів, перевірку їх наявності та повноти, а також подальшу обробку. Процес розпочинається з ініціалізації системи, після чого система намагається отримати індекс вибраного елемента. Якщо елемент вибрано, завантажуються відповідні дані з файлу, перевіряється їхня повнота, і за умови достатньої кількості даних

вони передаються до економічного розрахунку. У разі відсутності вибраного елемента або недостатньої кількості даних процес повертається до етапу завантаження файлу, забезпечуючи безперервну обробку.



Рисунок 3.3 – Робота модуля load_employee_data

Блок-схема на рисунку 3.4 описує процес обробки даних у HR-системі, який включає ініціалізацію, завантаження інформації про співробітників, аналіз їхніх даних та подальше опрацювання залежно від умов. Процес розпочинається

з ініціалізації системи, після чого виконується завантаження вихідних даних про співробітників. Далі відбувається очищення елементів списку для підготовки до обробки, а потім перевіряється наявність економічних даних. Якщо такі дані доступні, система завантажує шлях до файлу, опрацьовує ці дані та виконує економічний аналіз. На наступному етапі перевіряється, чи позиція співробітника відповідає заданому фільтру, і за позитивної відповіді дані виводяться для подальшої роботи, в іншому разі процес завершується.

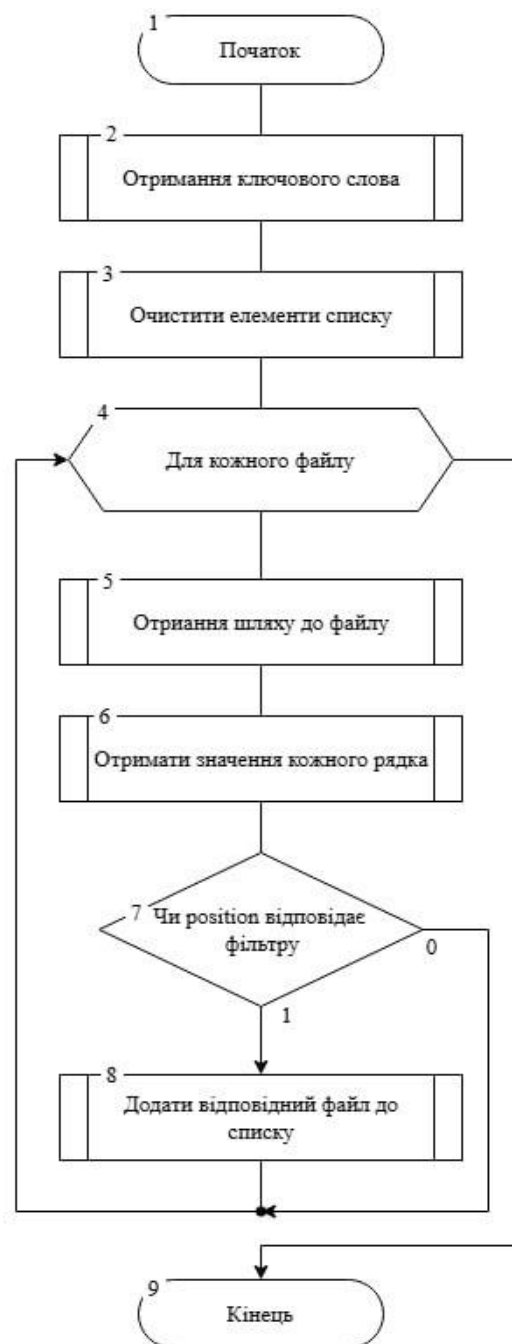


Рисунок 3.4 – Робота модуля `apply_filter`

Цей процес забезпечує гнучкість і адаптивність системи до змін у базі даних, дозволяючи HR-фахівцям швидко реагувати на оновлення інформації. Завдяки структурованому підходу блок-схеми система ефективно справляється з великими обсягами даних, гарантуючи актуальність і точність економічного аналізу, що є вкрай важливим для ухвалення стратегічних рішень у сфері управління персоналом.

3.3 Розробка модуля візуалізації та генерації звітів

Блок-схема на рисунку 3.5 ілюструє деталізований процес обробки даних у HR-системі, який охоплює ініціалізацію, завантаження списків співробітників, їх фільтрацію, аналіз даних та формування робочих графіків. Процес розпочинається з ініціалізації системи, після чого виконується перевірка наявності основного списку співробітників у базі даних. Якщо список існує, система створює попередній графік, завантажує дані з відповідних файлів у директорії, перевіряє їх повноту а також відповідність позиції співробітника заданому фільтру. За позитивного результату перевірки записи додаються до списку, після чого система оцінює наявність уже сформованого списку. Якщо список існує, створюється робочий графік, а за його відсутності додається новий графік для подальшого використання. У разі невідповідності умов фільтрації чи недостатньої кількості даних процес повертається до етапу завантаження файлів, забезпечуючи циклічність і безперервність роботи. Завершальним етапом є показ робочого графіка, що дозволяє HR-фахівцям ефективно управляти розподілом робочих завдань.

Цей метод гарантує високу гнучкість системи, дозволяючи автоматично адаптувати робочі графіки до змін у базі даних у реальному часі. Крім того, поєднання фільтрації з циклічним аналізом дає змогу HR-менеджерам оперативно реагувати на зміни в складі команди. Така функціональність особливо корисна під час масових оновлень або реорганізацій, коли потрібна швидка адаптація. Система також підтримує інтеграцію з іншими аналітичними інструментами, що розширює можливості для детального аналізу.

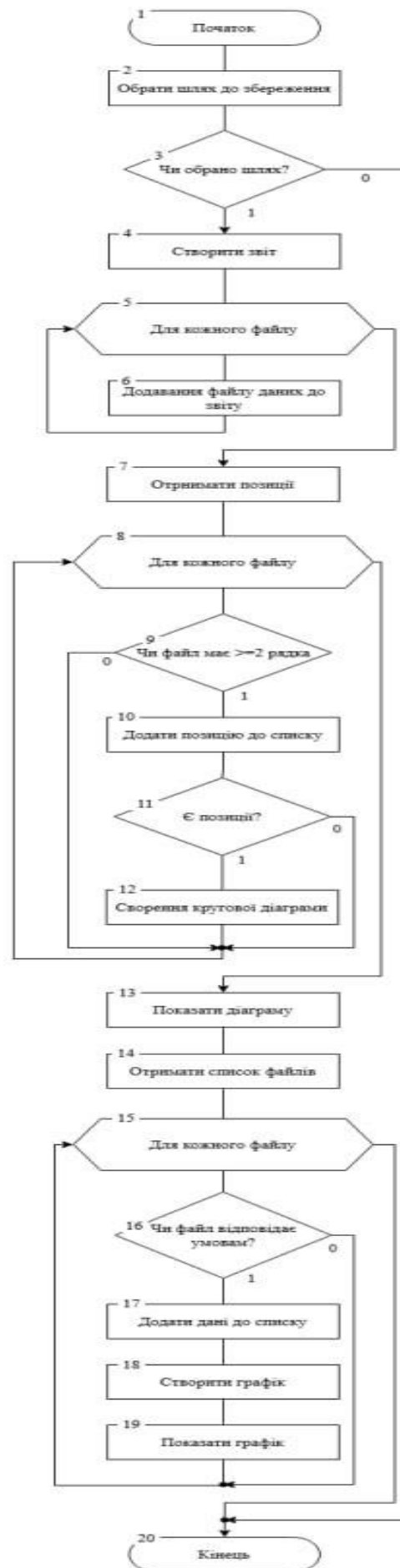


Рисунок 3.5 – Робота класу ReportVisualizer

Блок-схема на рисунку 3.6 описує процес обробки даних у HR-системі, що охоплює ініціалізацію, завантаження списків співробітників, їх аналіз і створення робочих графіків. Процес розпочинається з ініціалізації системи, після чого перевіряється наявність основного списку співробітників. Якщо список існує, система завантажує дані з файлів, очищує робочий список і перевіряє, чи є щонайменше 2 рядки даних для аналізу. Далі перевіряється відповідність позиції співробітника заданому фільтру, і за позитивного результату записи додаються до списку. Якщо список уже сформований, система створює робочий графік, а за його відсутності — додає новий графік. У разі невідповідності умов фільтрації процес повертається до завантаження даних, забезпечуючи циклічність.

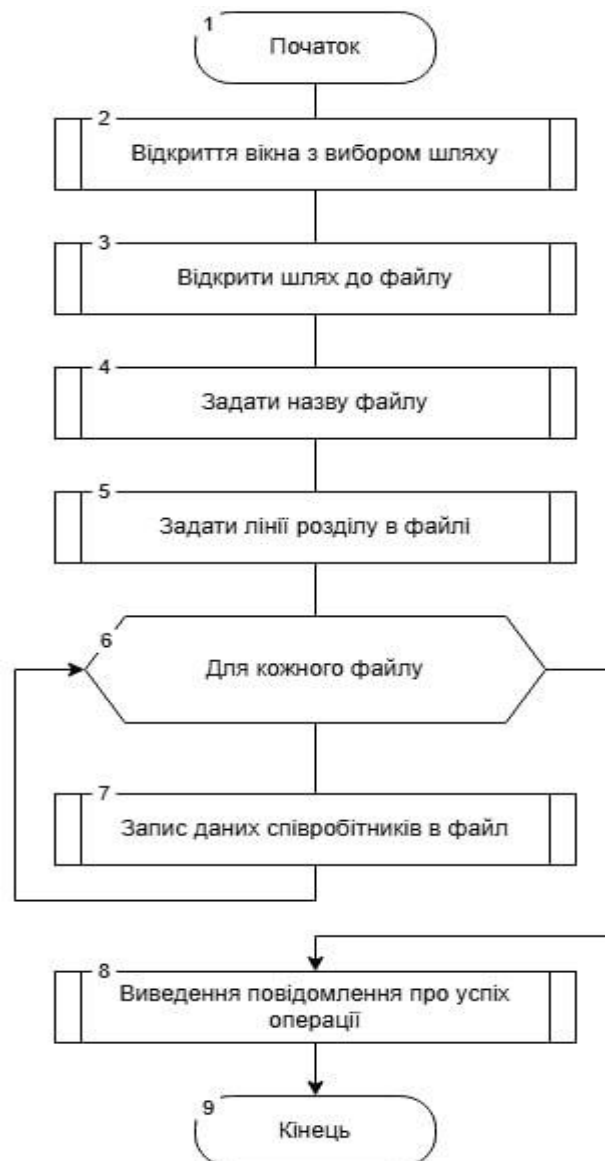


Рисунок 3.6 – Робота модуля `generate_report`

Блок-схема на рисунку 3.7 описує деталізований процес обробки даних у HR-системі, який охоплює ініціалізацію, завантаження списків співробітників, їх фільтрацію, аналіз і подальше створення робочих графіків на основі отриманих результатів. Процес розпочинається з ініціалізації системи, після чого виконується зчитування основного списку співробітників із файлів у відповідній директорії. Далі перевіряється наявність позиції співробітника, і якщо вона є, система завантажує дані з файлу, перевіряючи їх повноту — зокрема, наявність щонайменше 2 рядків. За наявності достатньої кількості даних проводиться економічний аналіз, який враховує різні параметри, такі як продуктивність чи витрати на співробітника. На наступному етапі перевіряється, чи існує сформований список для подальшої обробки, і якщо він є, дані сортуються за заданими критеріями, після чого виводиться повідомлення про успішне виконання операції. Якщо ж список відсутній або даних недостатньо, система повертається до етапу завантаження файлів, забезпечуючи циклічність і безперервність обробки. Завершальним кроком є створення робочого графіка на основі проаналізованих даних, що дозволяє HR-фахівцям ефективно розподіляти завдання, а також відображення результатів аналізу для подальшого використання в управлінні персоналом.

Цей підхід забезпечує високу гнучкість системи, дозволяючи швидко адаптуватися до змін у складі персоналу чи оновленнях даних, що є критично важливим для оперативного реагування на потреби компанії. Крім того, інтеграція економічного аналізу та автоматичного сортування даних сприяє оптимізації ресурсів, допомагаючи менеджерам ухвалювати обґрунтовані рішення щодо розподілу робочого часу та оцінки продуктивності.

Додатково, система підтримує інтерактивність, дозволяючи користувачам вручну коригувати дані чи переглядати проміжні результати через графічний інтерфейс, що значно спрощує роботу з великими обсягами інформації. Така функціональність особливо корисна під час підготовки звітів чи планування масових змін, таких як ротація кадрів чи введення нових посад.

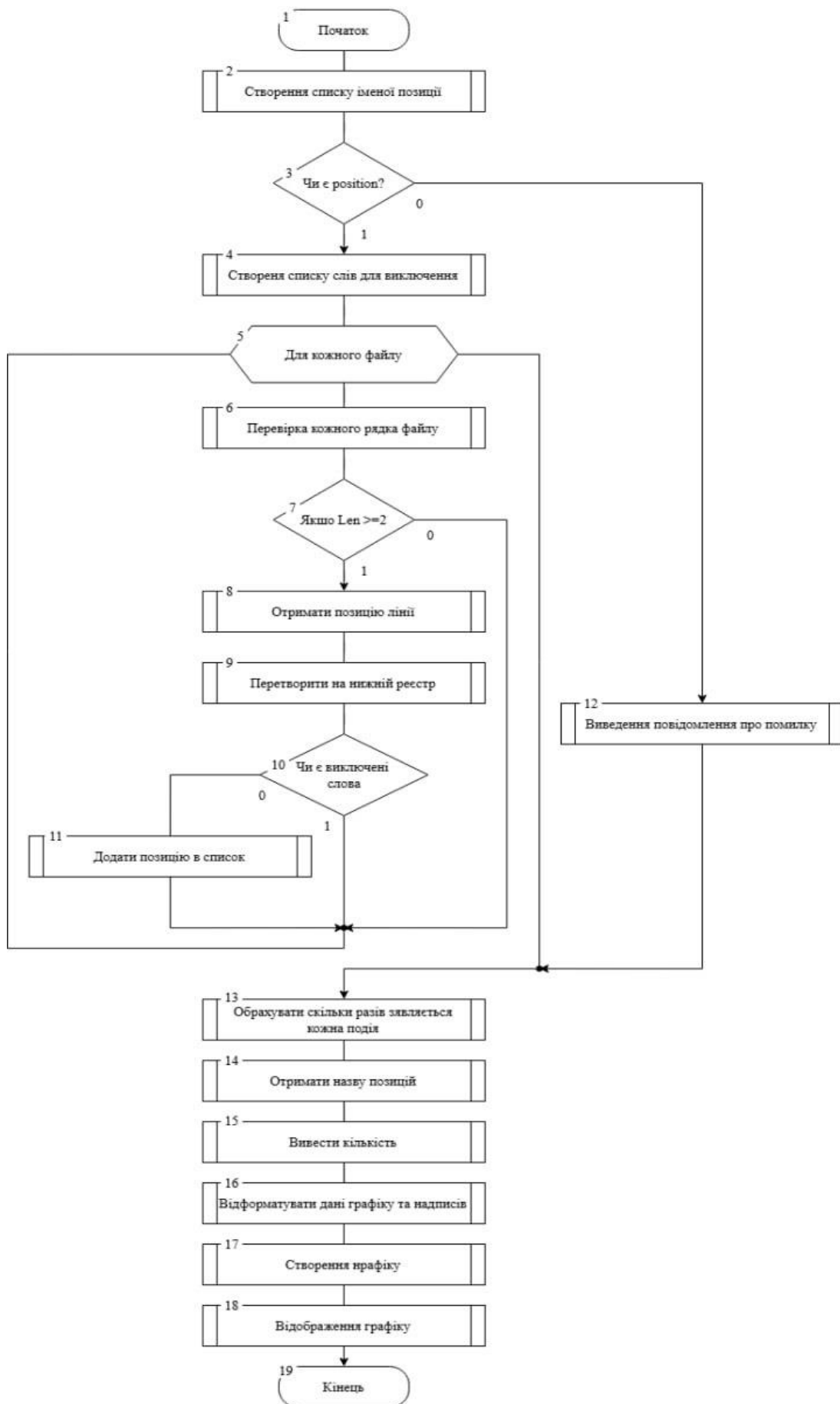


Рисунок 3.7 – Робота модуля show_position_chart

Блок-схема на рисунку 3.8 детально ілюструє процес обробки даних у HR-системі, який охоплює ініціалізацію, завантаження списків співробітників, їх фільтрацію, економічний аналіз і створення робочих графіків для ефективного управління персоналом. Процес розпочинається з ініціалізації системи, після чого виконується перевірка наявності експорту даних, що є ключовим для подальшої роботи з інформацією про співробітників. Якщо експорт доступний, система зчитує списки співробітників із заданої директорії, включаючи детальні дані, такі як ім'я, посада та продуктивність, а також перевіряє наявність текстового файлу KPI для аналізу ключових показників. За умови, що файл містить щонайменше 6 рядків даних, виконується економічний аналіз, який враховує витрати, продуктивність і інші метрики, після чого дані сортуються за заданими критеріями (наприклад, за досвідом чи рейтингом) і виводиться повідомлення про успішне виконання операції. Наступним етапом є перевірка наявності списку даних із застосуванням фільтра за позицією, і за позитивного результату створюється робочий графік, який відображається для використання HR-фахівцями, полегшуючи планування робочого часу. У разі відсутності даних, недостатньої їх кількості чи невідповідності умов фільтрації процес повертається до етапу завантаження файлів, забезпечуючи циклічність і безперервність обробки.

Цей підхід дозволяє системі бути максимально адаптивною до змін у базі даних, що особливо важливо в умовах динамічної роботи HR-відділу, коли інформація про співробітників може оновлюватися щоденно. Наприклад, у разі додавання нових працівників чи зміни їхніх посад система автоматично враховує ці зміни під час наступного циклу обробки, забезпечуючи актуальність робочих графіків і аналітичних даних. Така функціональність значно економить час HR-фахівців, дозволяючи їм зосередитися на стратегічних завданнях, таких як розвиток персоналу чи планування кар'єрного зростання, замість рутинної обробки даних.

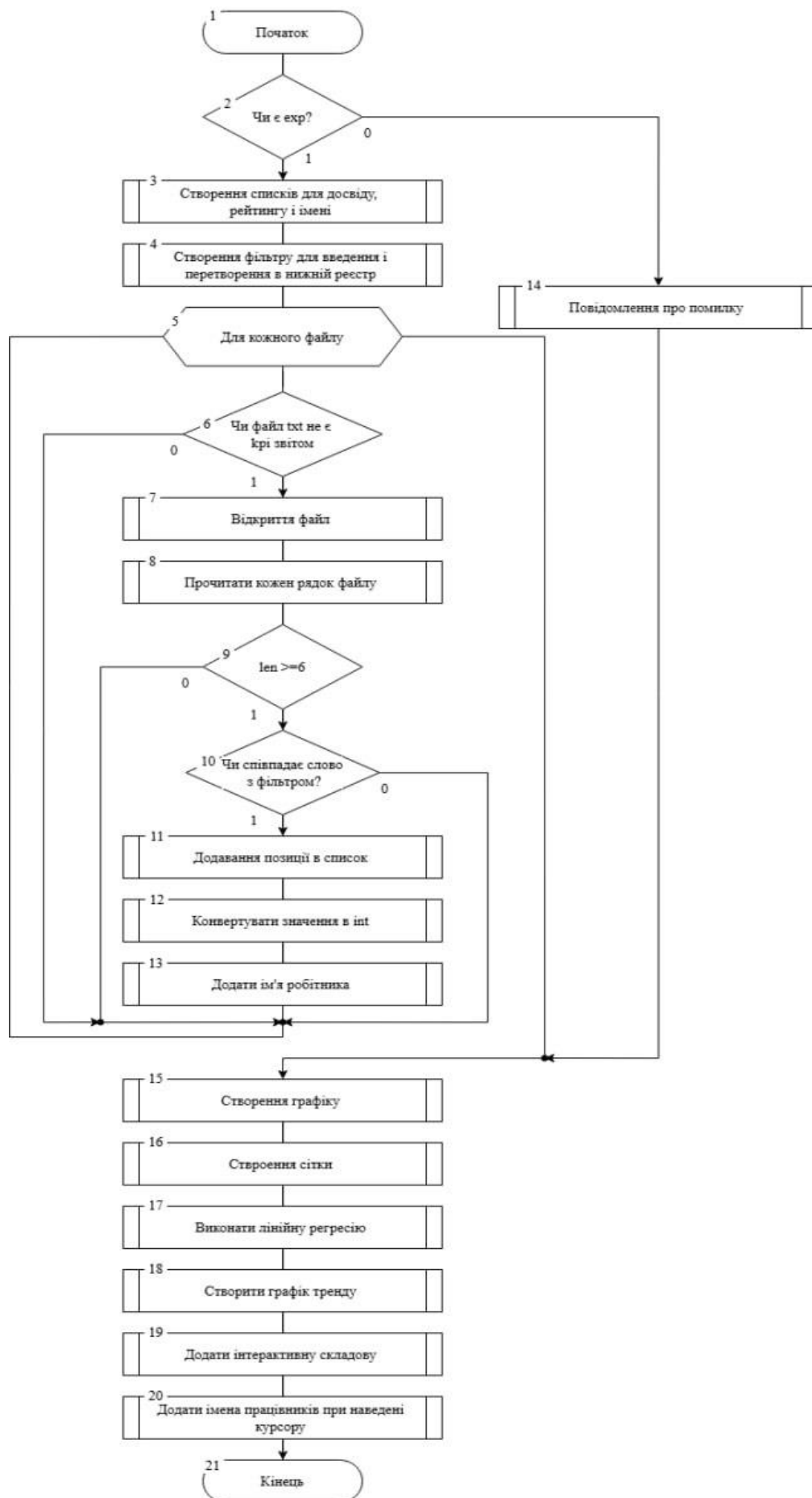


Рисунок 3.8 – Робота модуля show_exp_vs_performance

3.4 Висновки

У третьому розділі було обґрунтовано вибір програмних засобів для реалізації інформаційної системи управління персоналом підприємства. З огляду на специфіку задачі та потребу в гнучкій обробці даних, було обрано мову програмування Python, яка відзначається простотою, широким набором бібліотек і зручністю інтеграції з графічними інтерфейсами. Розробку було здійснено у середовищі Visual Studio Code, що забезпечує комфортну роботу з кодом, зручне налагодження та розширення функціональності.

Система реалізує функціонал управління особовими справами співробітників, оцінки KPI, розрахунку бонусів і прогнозування кар'єрного розвитку. Замість використання бази даних було обрано просту обробку текстових файлів, що містять дані про співробітників та їхні оцінки KPI. Для візуалізації даних використовуються бібліотеки matplotlib та networkx, які дозволяють наочно відображати розподіл персоналу за посадами, оцінки KPI та інші статистичні дані.

Система також надає можливість генерації професійних PDF-звітів за допомогою бібліотеки reportlab, що включають усю необхідну інформацію про співробітника, його оцінки KPI та навіть QR-код для швидкого доступу до основних даних. Модуль моделювання та прогнозування дозволяє HR-менеджерам аналізувати різні сценарії, такі як звільнення співробітників або скорочення бонусного бюджету, і оцінювати їхній вплив на компанію.

Такий підхід забезпечує ефективну та наочну взаємодію з користувачем без необхідності складної серверної інфраструктури або баз даних, що є оптимальним рішенням для малих та середніх підприємств або для використання на ранніх етапах впровадження системи управління персоналом.

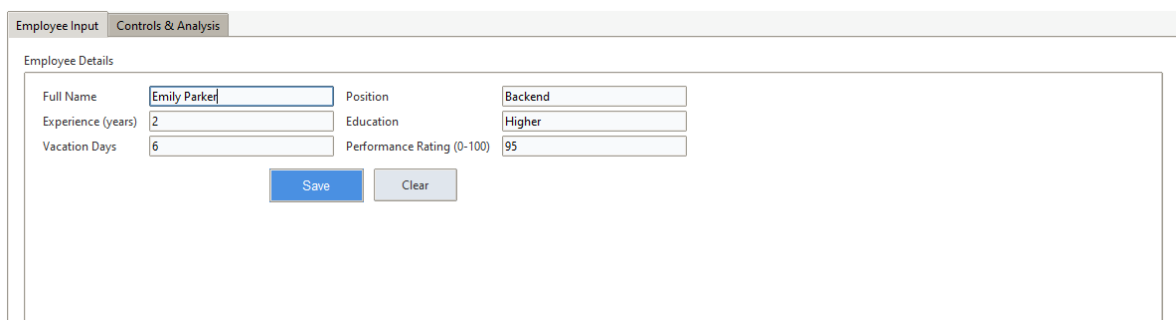
4 ТЕСТУВАННЯ ПРОГРАМИ

4.1 Тестування системи

Тестування інформаційної системи керування персоналом на платформі ПК – це процес перевірки її функціональності, стабільності та безпеки під час взаємодії з користувачем через графічний інтерфейс. Основна мета тестування – впевнитися, що система правильно реагує на введені дані, коректно обробляє інформацію про співробітників і надає очікувану відповідь без збоїв чи втрати даних.

Першим кроком є встановлення та налаштування тестової версії системи на комп'ютері та запуск її в робочому середовищі. Далі проводиться серія функціональних тестів, які охоплюють обробку основних функцій системи, таких як збереження та завантаження даних співробітників, фільтрація за посадою, генерація звітів, розрахунок бонусів та прогнозування, а також створення діаграм розподілу посад.

Під час тестування було перевірено коректність збереження даних співробітника з використанням відповідної форми введення. Система перевіряє наявність обов'язкових полів. Після успішного збереження дані зберігаються у текстовому файлі у директорії employees. Відповідну поведінку системи наведено на рисунку 4.1.



The screenshot displays a web application interface with two tabs: 'Employee Input' (active) and 'Controls & Analysis'. Below the tabs is a section titled 'Employee Details' containing a form with the following fields and values:

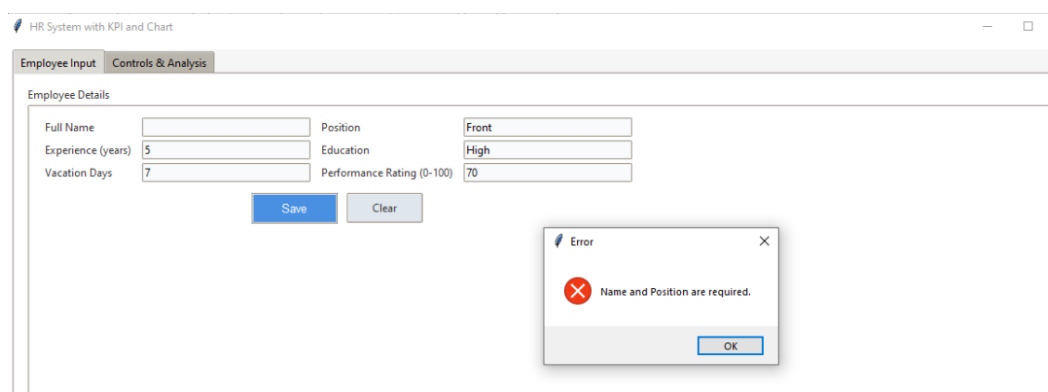
Field	Value
Full Name	Emily Parker
Position	Backend
Experience (years)	2
Education	Higher
Vacation Days	6
Performance Rating (0-100)	95

At the bottom of the form are two buttons: 'Save' (blue) and 'Clear' (grey).

Рисунок 4.1 – Результат введення додавання співробітника

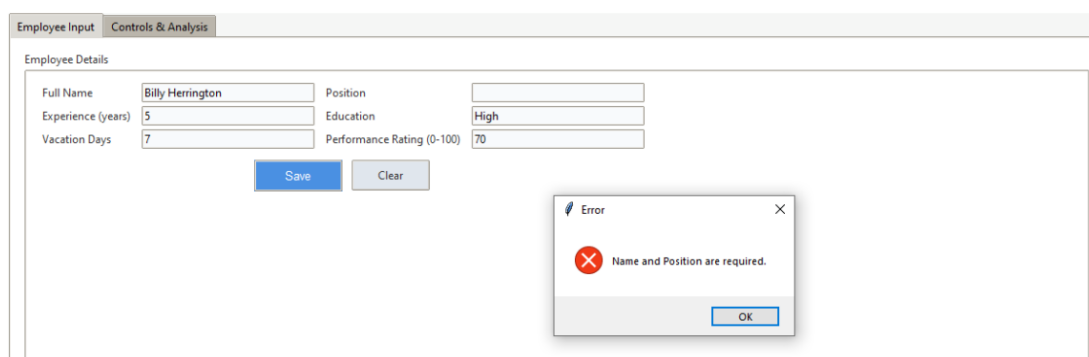
На рисунку 4.2 демонструє ситуацію, коли користувач натиснув кнопку «Save», але система видала помилку через відсутність обов'язкових даних.

З'явилося спливаюче вікно з повідомленням «Error» із червоним знаком «X» і текстом «Name and Position are required», що вказує на валідацію введених даних. Це свідчить про реалізацію механізму перевірки в методі `save_employee`, який вимагає заповнення полів «Full Name» і «Position» перед збереженням. Кнопка «OK» у спливаючому вікні дозволяє користувачу закрити повідомлення та виправити помилку, повернувшись до форми для введення відсутньої інформації. Такий підхід підвищує надійність системи, запобігаючи збереженню неповних або некоректних даних.



На рисунку 4.2 – Інтерфейс вкладки введення даних із повідомленням про помилку не заповненого поля «Full Name»

На рисунку 4.3 зображено не заповнено поле Position і також спливає вікно «Error» із повідомленням «Name and Position are required» і кнопкою "OK", вказуючи на необхідність заповнення обов'язкових полів.



На рисунку 4.3 – Інтерфейс вкладки введення даних із повідомленням про помилку не заповненого поля «Position»

Наступним етапом тестування був перевірка можливості створення PDF-звітів для обраних співробітників. Система збирає всі доступні дані про співробітника, включаючи основну інформацію та оцінки KPI, та формує добре структурований звіт з логотипом компанії, QR-кодом та місцем для підпису HR-спеціаліста на рисунку 4.4.

Employee Report

Name: Ava Thompson
Position: Frontend
Experience: 3 years
Education: Secondary
Vacation Days: 2
Performance Rating: 65/100

KPI Evaluation:

- Technical Skills: 7/10
- Teamwork: 7/10
- Stress Tolerance: 9/10
- Average Score: 7.67/10

HR Signature _____

Date _____

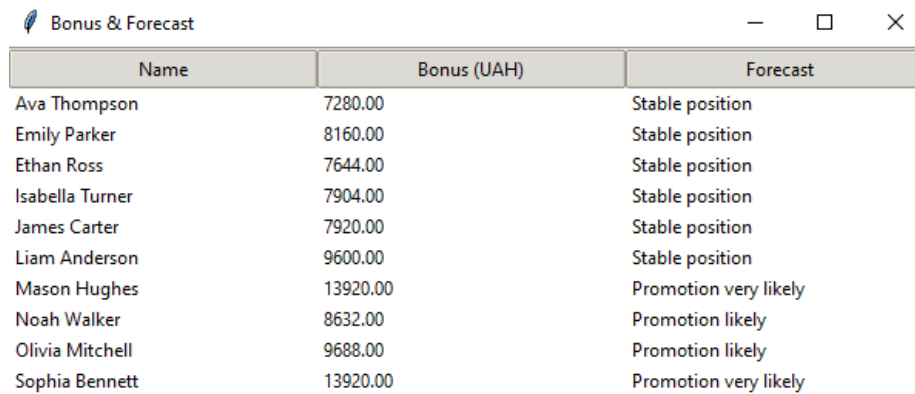


Рисунок 4.4 – Результат створення PDF-звіту

На рисунку 4.5 тестування інформаційної системи керування персоналом є перевірка коректності виконання функції розрахунку бонусів та прогнозування розвитку кар'єри. Ця функція призначена для генерації аналітичної інформації на основі показників продуктивності співробітників.

Після успішного тестування базової функціональності, розпочинається перевірка аналітичних можливостей системи. Під час виконання функції

розрахунку бонусів система збирає інформацію з файлів співробітників, формує статистичну вибірку та обчислює розмір бонусу з урахуванням багатьох факторів, зокрема базової суми, ефективності роботи, досвіду, освіти та використаних днів відпустки.



Name	Bonus (UAH)	Forecast
Ava Thompson	7280.00	Stable position
Emily Parker	8160.00	Stable position
Ethan Ross	7644.00	Stable position
Isabella Turner	7904.00	Stable position
James Carter	7920.00	Stable position
Liam Anderson	9600.00	Stable position
Mason Hughes	13920.00	Promotion very likely
Noah Walker	8632.00	Promotion likely
Olivia Mitchell	9688.00	Promotion likely
Sophia Bennett	13920.00	Promotion very likely

Рисунок 4.4 – Результат розрахунків бонусів

На рисунку 4.5 зображено інтерактивність графіка, забезпечена бібліотекою `mplcursors`, дає змогу ідентифікувати конкретних працівників, що полегшує аналіз окремих випадків. Трендова лінія, побудована за допомогою лінійної регресії, допомагає визначити загальну залежність між досвідом і продуктивністю, підтримуючи стратегічні рішення, такі як планування кар'єрного зростання, виявлення працівників для підвищення або додаткового навчання. Діаграма сприяє прийняттю обґрунтованих кадрових рішень у контексті управління персоналом.

Крім того, діаграма може слугувати інструментом для моніторингу ефективності HR-політик підприємства. Аналізуючи розподіл точок і нахил трендової лінії, менеджмент може оцінити, наскільки поточні програми навчання чи мотивації впливають на продуктивність, а також виявити можливі аномалії, такі як працівники з низьким рейтингом при значному досвіді, що може

вказувати на потребу в додаткових заходах підтримки чи перекваліфікації. Це робить графік цінним ресурсом для адаптивного управління персоналом у реальному часі

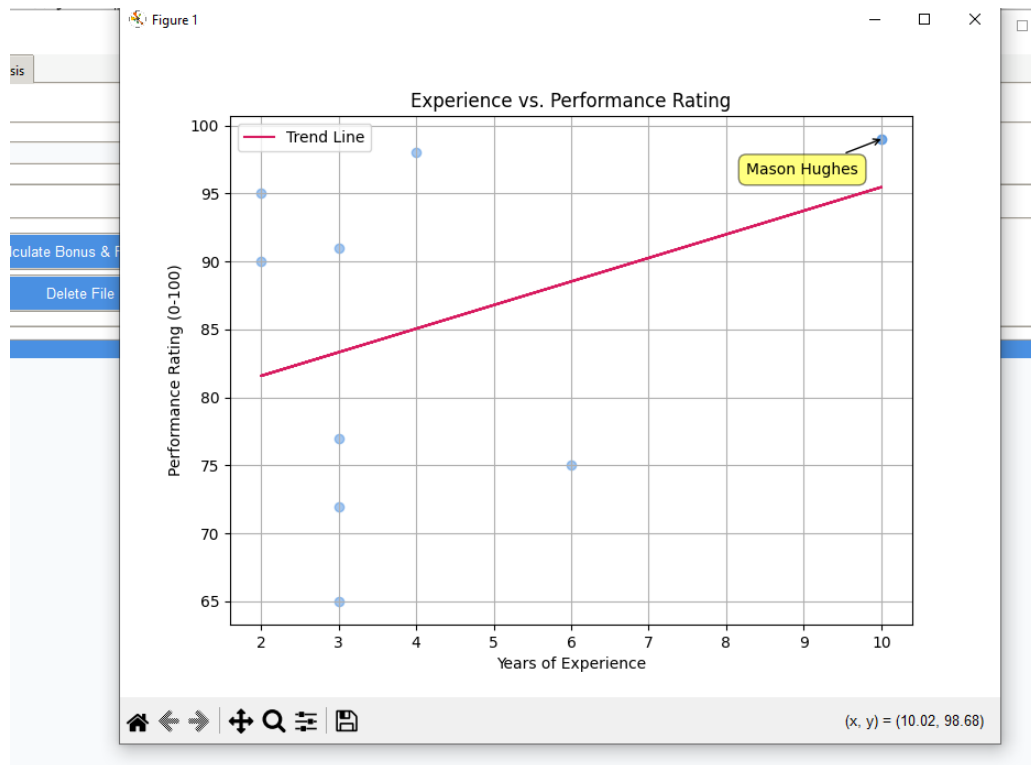


Рисунок 4.5 – Діаграма розсіювання продуктивності від досвіду

Комплексне тестування відіграє вирішальну роль у забезпеченні надійного функціонування модулів інформаційної системи для керування персоналом підприємства. Поетапні перевірки дозволяють оперативно виявляти потенційні помилки у взаємодії компонентів, аналізувати реакцію системи на нестандартні кадрові ситуації та контролювати відповідність алгоритмів реальним HR-процесам. Багаторівневе випробування функціоналу – від базових операцій з обліку співробітників до генерації складної аналітики та звітності – підтверджує гнучкість і стабільність програмного забезпечення.

Усі проведені перевірки спрямовані на досягнення високої якості інформаційної системи, яка не лише автоматизує рутинні кадрові процеси, але й забезпечує позитивний досвід для HR-фахівців. Безвідмовна робота програмних модулів гарантує, що співробітники відділу персоналу зможуть ефективно

керувати даними про співробітників через зручний інтерфейс та покладатися на систему в питаннях збереження й обробки конфіденційної інформації. Таким чином, тестування стає невід'ємною складовою розробки ефективної, захищеної та зручної інформаційної системи управління персоналом.

4.2 Розробка інструкції користувача

Керівництво користувача включає опис технічних умов, необхідних для стабільної роботи програмного продукту. У ньому зазначено параметри пристроїв, на яких застосунок може бути встановлений та ефективно функціонувати. Детальну інформацію щодо мінімальних та рекомендованих характеристик мобільного пристрою для запуску застосунку наведено у таблицях 4.1 та 4.2.

Таблиця 4.1 – Мінімальна конфігурація:

Компонент	Вимоги
Процесор	Intel Core i3 або AMD Ryzen 3
Об'єм оперативної пам'яті	4 ГБ для 64-разрядної системи
Місце на жорсткому диску	500 МБ
Операційна система	Windows 7/10 або Linux

Таблиця 4.2 – Рекомендована конфігурація:

Компонент	Вимоги
Процесор	Intel Core i5 або AMD Ryzen 5
Об'єм оперативної пам'яті	8 ГБ для 64-разрядної системи
Місце на жорсткому диску	2-4 ГБ
Операційна система	Windows 10/11 або Linux

Після запуску інформаційної системи HR Management користувач отримує доступ до повнофункціонального інтерфейсу для керування електронними особистими справами співробітників. Система забезпечує створення та збереження індивідуальних профілів працівників із детальною інформацією:

ПІБ, посада, досвід роботи, освіта, кількість днів відпустки та показники ефективності.

Основний функціонал системи включає додавання нових співробітників, редагування існуючих даних, фільтрацію записів за критеріями, розрахунок преміальних виплат та прогнозування кар'єрного зростання. Перед виконанням складних операцій з даними, наприклад, формування звітів чи оцінювання KPI, система перевіряє правильність і повноту введеної інформації.

Після створення особистої справи співробітника система автоматично розраховує рекомендовані преміальні виплати на основі комплексного аналізу показників: рівня кваліфікації, стажу роботи, освіти та використаних днів відпустки. Зібрані дані використовуються для генерації різноманітних звітів, доступних через функцію «Generate Report», а також для створення графічних візуалізацій за допомогою інструменту «Position Chart».

Модуль оцінки KPI дозволяє проводити багатокритеріальне оцінювання продуктивності працівників за шкалою від 0 до 10 за категоріями «Технічні навички», «Командна робота» та «Стресостійкість». Результати оцінювання зберігаються в системі та використовуються для формування профільних звітів у PDF-форматі з QR-кодом для швидкої ідентифікації. Система також підтримує функціонал симуляції сценаріїв змін у кадровому складі для моделювання впливу плинності персоналу чи скорочення преміального фонду на загальні показники підприємства.

4.3 Висновки

У четвертому розділі було проведено всебічне тестування програмного забезпечення «HR System», що функціонує на настільних комп'ютерах з графічним інтерфейсом Tkinter. Основний фокус тестування був спрямований на перевірку коректності обробки дій користувача, зокрема валідації введених даних про співробітників, та належного функціонування ключових компонентів системи, таких як збереження особистих справ, розрахунок бонусів, оцінювання KPI та генерація звітів.

Додатково було сформовано детальне керівництво користувача, що описує процедуру роботи з системою управління персоналом. Посібник охоплює етапи первинного налаштування системи, процес додавання та редагування даних співробітників, застосування фільтрів, формування PDF-звітів, побудову діаграм розподілу посад та проведення сценарного моделювання змін у кадровому складі підприємства.

ВИСНОВКИ

У бакалаврській кваліфікаційній роботі було розроблено методи та програмні засоби для створення модуля інформаційної системи керування даними персоналу підприємства. Проведено аналіз існуючих систем автоматизації управління персоналом, таких як BambooHR, Workday HCM, PeopleForce та Odoo HR, із виявленням їхніх сильних і слабких сторін, а також обґрунтовано потребу в розробці інтерактивного модуля для аналізу, прогнозування та візуалізації даних співробітників.

Розроблено метод інтерактивного введення та обробки даних персоналу, особливістю якого є динамічне оновлення інформації в реальному часі з можливістю фільтрації за посадами та перегляду результатів, що спрощує управління персоналом. Подальшого розвитку набув метод оцінки ключових показників ефективності (KPI) і прогнозування бонусів, який базується на комбінації параметрів, таких як досвід, освіта, відпустки та рейтинг продуктивності, забезпечуючи точний і обґрунтований аналіз. Також удосконалено метод моделювання управлінських сценаріїв, який дозволяє прогнозувати вплив звільнень чи скорочення бонусів на продуктивність і бюджет підприємства.

Практична цінність отриманих результатів полягає в розробці модуля інформаційної системи, що інтегрує введення даних, оцінку KPI, прогнозування кар'єрного зростання та генерацію аналітичних звітів у PDF із QR-кодами. Розроблені методи підвищення інтерактивності та наочності, зокрема через інтерактивні графіки розподілу посад і залежності продуктивності від досвіду, створені за допомогою бібліотеки Matplotlib, полегшують аналіз даних HR-менеджерами. Проведене тестування підтвердило стабільність роботи модуля, коректність обчислень і відповідність функціональним вимогам.

Розроблений модуль може бути використаний підприємствами для оптимізації управління персоналом, аналізу продуктивності та планування кадрових рішень. Він також придатний для освітнього процесу в галузі інженерії

програмного забезпечення для демонстрації практичного застосування програмних рішень у HR-управлінні.

Ключові слова: Python, tkinter, Matplotlib, reportlab, HR-система, управління персоналом, KPI, прогнозування, візуалізація, PDF-звіти.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Competing on Analytics: The New Science of Winning. Harvard Business School Press, 2007. 240 p.
2. Burns M. M., Mercer L. Creating Value Through People: Discussions with Talent Leaders. Wiley & Sons, Limited, John, 2015.
3. Sen S. Digital HR Strategy: Achieving Sustainable Transformation in the Digital Age. Kogan Page, Limited, 2020.
4. Новицький О.М. Розробка програмних модулів інформаційної системи для керування персоналом підприємства. Матеріали LIV Всеукраїнської науково-технічної конференції професорсько-викладацького складу, науковців, аспірантів та студентів підрозділів університету з участю працівників підприємств (НТКП ВНТУ-2025). Вінниця, 2025. URL: <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2025/paper/view/24291> (дата звернення 14.04.2025).
5. Fulcher J., Marasco K., Cote T. People Operations: Automate HR, Design a Great Employee Experience, and Unleash Your Workforce. Wiley & Sons, Incorporated, John, 2021. – 272 p.
6. Bussin M., Diez F., Lee V. Fundamentals of HR Analytics: A Manual on Becoming HR Analytical. Emerald Publishing Limited, 2019. – 280 p.
7. BambooHR: The Complete HR Software for People, Payroll & Benefits URL: https://www.bamboohr.com/p1?utm_source (date of access: 15.04.2025).
8. Workday Platform HR, Finance, Planning, Spend URL: https://www.workday.com/?utm_source=chatgpt.com (date of access 15.04.2025).
9. HRM система управління персоналом PeopleForce URL: https://lp.peopleforce.io/uk/hrm?utm_source (дата звернення 15.04.2025).
10. Кадри Odoo - Функції | Odoo URL: https://www.odoo.com/uk_UA/app/employees-features?utm_source (дата звернення 17.04.2025).

11. Decision Science for Human Resources (HR) Analytics URL: <https://graphite-note.com/decision-science-for-human-resources-hr-analytics/> (date of access 17.04.2025).
12. Webex Contact Center URL: <https://surl.li/zajsmv> (date of access 17.04.2025).
13. Pandas – Python Data Analysis Library. URL: <https://pandas.pydata.org/> (date of access: 17.04.2025).
14. Python Tkinter – GeeksforGeeks URL: https://www.geeksforgeeks.org/python-gui-tkinter/?utm_source (date of access: 10.05.2025).
15. Interactive data selection cursors for Matplotlib URL: https://mplcursors.readthedocs.io/en/stable/?utm_source (date of access: 10.05.2025).
16. ReportLab Docs URL: https://docs.reportlab.com/demos/hello_world (date of access: 10.05.2025).
17. Grad URL: <https://www.upgrad.com/tutorials/software-engineering/python-tutorial/counter-in-python> (date of access: 10.05.2025).
18. Building HR Management Systems URL: <https://www.codingwithcookie.com/> (date of access: 10.05.2025).

ДОДАТКИ

Додаток А – Технічне завдання

65

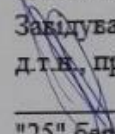
Додаток А – Технічне завдання

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії

ЗАТВЕРДЖУЮ

Завідувач кафедри ІТЗ

д.т.н., проф.

 О. Н. Романюк

"25" березня 2025 р.

Технічне завдання
на бакалаврську кваліфікаційну роботу «Розробка модулю інформаційної
системи для керування даними персоналу підприємства» за спеціальністю
121 Інженерія програмного забезпечення

Керівник бакалаврської кваліфікаційної роботи:

 к.т.н., доцент Г.Б. Ракитянська

"24" 03 2025 р.

Виконав:

 студент гр. 4ПІ-216 О. М. Новицький

"24" 03 2025 р.

Вінниця – 2025 року

1. Найменування та галузь застосування

Бакалаврська кваліфікаційна робота: «Розробка модулю інформаційної системи для керування даними персоналу підприємства».

Галузь застосування – управління персоналом на підприємствах та оптимізація HR-процесів.

2. Підстава для розробки.

Підставою для розробки бакалаврської кваліфікаційної роботи є індивідуальне завдання на БКР та наказ від 20 березня 2025 р. №97 ректора по ВНТУ про закріплення тем БКР.

3. Мета та призначення розробки.

Метою роботи є підвищення ефективності управління персоналом через створення інтерактивного модуля для аналізу, прогнозування та візуалізації даних співробітників.

Призначення роботи – розробка методів і засобів для автоматизації процесів оцінки KPI, прогнозування бонусів і моделювання кадрових сценаріїв.

4. Вихідні дані для проведення НДР

Перелік основних літературних джерел, на основі яких буде виконуватись БКР.

1. Liu S., Zhou Z.-H. Machine Learning. Singapore: Springer, 2021 – 578 p.
2. Madala H. R. Inductive Learning Algorithms for Complex Systems Modeling. Taylor & Francis Group, 2019. 380 p.
3. Krishnan T., McLachlan G. J. EM Algorithm and Extensions. Wiley & Sons, Incorporated, John, 2007. 384 p.

5. Технічні вимоги

Методи управління персоналом – оцінка KPI, прогнозування бонусів, моделювання сценаріїв; набори даних – текстові файли з інформацією про співробітників; параметри алгоритму – рейтинг продуктивності, досвід роботи, освіта, відпусткові дні; методи візуалізації – Matplotlib; програмне середовище – Python, tkinter; середовище розробки – Visual Studio Code; вихідні результати – аналітичні звіти в PDF із QR-кодами, інтерактивні графіки розподілу посад і продуктивності; вхідні дані: ім'я, посада, досвід, освіта, рейтинг, фільтри за посадою.

6. Конструктивні вимоги.

Графічна та текстова документація повинна відповідати діючим стандартам України.

7. Перелік технічної документації, що пред'являється по закінченню робіт:

- пояснювальна записка до БКР;
- технічне завдання;
- лістинги програми.

8. Вимоги до рівня уніфікації та стандартизації

При розробці програмних засобів слід дотримуватися уніфікації і ДСТУ.

9. Стадії та етапи розробки:

№ з/п	Назва етапів бакалаврської кваліфікаційної роботи	Строк виконання етапів роботи
1	Аналіз стану систем автоматизації управління персоналом	25.03.25 – 07.04.25
2	Розробка підсистеми введення та збереження даних співробітників	08.04.25 – 20.04.25
3	Розробка підсистеми інтерактивної візуалізації даних	21.04.25 – 03.05.25
4	Розробка модуля оцінки KPI та прогнозування	04.04.25 – 15.05.025
5	Тестування роботи застосунку	16.05.25 – 24.05.25
6	Оформлення матеріалів до захисту БКР	25.05.25 – 30.05.25

10. Порядок контролю та прийняття.

Виконання етапів бакалаврської кваліфікаційної роботи контролюється керівником згідно з графіком виконання роботи. Захист бакалаврської кваліфікаційної роботи здійснюється ДЕК, затвердженою зав. кафедрою згідно з графіком.

Додаток Б – Протокол перевірки на плагіат

69

Додаток Б – Протокол перевірки на плагіат

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: Розробка модулю інформаційної системи для керування даними персоналу підприємства

Тип роботи: бакалаврська кваліфікаційна робота

(бакалаврська кваліфікаційна робота / магістерська кваліфікаційна робота)

Підрозділ кафедра програмного забезпечення, ФІТКІ, 4ПІ-21Б

(кафедра, факультет, навчальна група)

Коефіцієнт подібності текстових запозичень, виявлених у роботі
системою StrikePlagiarism 6,3 %

Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне)

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

Експертна комісія:

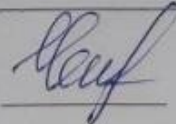
_____ (прізвище, ініціали, посада)

_____ (підпис)

_____ (прізвище, ініціали, посада)

_____ (підпис)

Особа, відповідальна за перевірку



Черноволик Г. О.

З висновком експертної комісії ознайомлений(-на)

Керівник _____
(підпис)

Ракитянська Г.Б. к.т.н., доц. каф. ПЗ
(прізвище, ініціали, посада)

Здобувач _____
(підпис)

Новицький О.М.
(прізвище, ініціали)

Додаток В – Лістинг програми

```

import os
import tkinter as tk
from tkinter import ttk, messagebox, filedialog
import matplotlib.pyplot as plt
from collections import Counter
import numpy as np
import mplcursors
from reportlab.lib.pagesizes import A4
from reportlab.lib.units import cm
from reportlab.lib.utils import ImageReader
from reportlab.pdfgen import canvas
import qrcode
import io
from datetime import datetime

EMPLOYEE_DIR = "employees"

class EmployeeFileManager:
    def __init__(self):
        if not os.path.exists(EMPLOYEE_DIR):
            os.makedirs(EMPLOYEE_DIR)

    def get_file_path(self, name):
        return os.path.join(EMPLOYEE_DIR, f"{name.replace(' ', '_')}.txt")

    def save_employee(self, name_var, position_var, experience_var, education_var,
vacation_days_var, rating_var):
        name = name_var.get().strip()
        position = position_var.get().strip()
        exp = experience_var.get().strip()
        edu = education_var.get().strip()
        vac = vacation_days_var.get().strip()
        rating = rating_var.get().strip()

```

```
if not name or not position:
```

```
    messagebox.showerror("Error", "Name and Position are required.")
```

```
    return False
```

```
try:
```

```
    path = self.get_file_path(name)
```

```
    with open(path, "w", encoding="utf-8") as f:
```

```
        f.write(f"{name}\n{position}\n{exp}\n{edu}\n{vac}\n{rating}")
```

```
    messagebox.showinfo("Saved", f"Data for {name} saved.")
```

```
    return True
```

```
except Exception as e:
```

```
    messagebox.showerror("Error", str(e))
```

```
    return False
```

```
def load_employees(self, listbox):
```

```
    listbox.delete(0, tk.END)
```

```
    for file in os.listdir(EMPLOYEE_DIR):
```

```
        if file.endswith(".txt") and "_kpi" not in file:
```

```
            listbox.insert(tk.END, file)
```

```
def load_employee_data(self, listbox, name_var, position_var, experience_var, education_var,
vacation_days_var, rating_var):
```

```
    idx = listbox.curselection()
```

```
    if not idx:
```

```
        return
```

```
    filename = listbox.get(idx[0])
```

```
    try:
```

```
        with open(os.path.join(EMPLOYEE_DIR, filename), "r", encoding="utf-8") as f:
```

```
            lines = f.readlines()
```

```
            if len(lines) >= 6:
```

```
                name_var.set(lines[0].strip())
```

```
                position_var.set(lines[1].strip())
```

```
                experience_var.set(lines[2].strip())
```

```
                education_var.set(lines[3].strip())
```

```

        vacation_days_var.set(lines[4].strip())
        rating_var.set(lines[5].strip())
except Exception as e:
    messagebox.showerror("Error", str(e))

def apply_filter(self, listbox, filter_position_var):
    keyword = filter_position_var.get().strip().lower()
    listbox.delete(0, tk.END)
    for file in os.listdir(EMPLOYEE_DIR):
        path = os.path.join(EMPLOYEE_DIR, file)
        with open(path, "r", encoding="utf-8") as f:
            lines = f.readlines()
            if len(lines) >= 2 and keyword in lines[1].lower():
                listbox.insert(tk.END, file)

def delete_employee_file(self, listbox):
    selected_idx = listbox.curselection()
    if not selected_idx:
        messagebox.showwarning("Warning", "Select an employee first.")
        return
    filename = listbox.get(selected_idx[0])
    if not filename.endswith(".txt") or "_kpi" in filename:
        messagebox.showwarning("Warning", "You can only delete main employee files.")
        return
    full_path = os.path.join(EMPLOYEE_DIR, filename)
    confirm = messagebox.askyesno("Confirm", f"Delete employee file: {filename}?")
    if confirm and os.path.exists(full_path):
        os.remove(full_path)
        messagebox.showinfo("Deleted", f"{filename} deleted.")
        self.load_employees(listbox)

def delete_kpi(self, listbox):
    selected_idx = listbox.curselection()
    if not selected_idx:
        messagebox.showwarning("Warning", "Select an employee first.")

```

```

    return
filename = listbox.get(selected_idx[0])
kpi_path = os.path.join(EMPLOYEE_DIR, filename.replace(".txt", "_kpi.txt"))
if os.path.exists(kpi_path):
    os.remove(kpi_path)
    messagebox.showinfo("Deleted", "KPI file deleted.")
else:
    messagebox.showinfo("Not Found", "No KPI file found for this employee.")

```

```
class ReportVisualizer:
```

```
    def generate_report(self):
```

```

        path = filedialog.asksaveasfilename(defaultextension=".txt", filetypes=[("Text File", "*.txt")])
        if not path:
            return
        try:
            with open(path, "w", encoding="utf-8") as report:
                report.write("FULL EMPLOYEE REPORT\n")
                report.write("="*60 + "\n\n")
                for file in os.listdir(EMPLOYEE_DIR):
                    with open(os.path.join(EMPLOYEE_DIR, file), "r", encoding="utf-8") as f:
                        report.write(f.read() + "\n" + "-"*50 + "\n")
                messagebox.showinfo("Done", "Report generated.")
        except Exception as e:
            messagebox.showerror("Error", str(e))

```

```
    def generate_pdf_report(self, listbox):
```

```

        selected_idx = listbox.curselection()
        if not selected_idx:
            messagebox.showwarning("Warning", "Select an employee first.")
            return

```

```

        filename = listbox.get(selected_idx[0])
        base = filename.replace(".txt", "").replace("_kpi", "")
        file_path = os.path.join(EMPLOYEE_DIR, filename)
        kpi_path = os.path.join(EMPLOYEE_DIR, base + "_kpi.txt")

```

```

with open(file_path, "r", encoding="utf-8") as f:
    lines = f.readlines()
if len(lines) < 6:
    messagebox.showwarning("Error", "Employee file is incomplete.")
    return

name = lines[0].strip()
position = lines[1].strip()
experience = lines[2].strip()
education = lines[3].strip()
vacation = lines[4].strip()
rating = lines[5].strip()

kpi_data = []
if os.path.exists(kpi_path):
    with open(kpi_path, "r", encoding="utf-8") as f:
        kpi_data = f.readlines()

pdf_name = filedialog.asksaveasfilename(defaultextension=".pdf", filetypes=[("PDF files",
"*.*.pdf")])
if not pdf_name:
    return

c = canvas.Canvas(pdf_name, pagesize=A4)
width, height = A4

logo_path = "company_logo.png"
if os.path.exists(logo_path):
    c.drawImage(logo_path, 2 * cm, height - 3.5 * cm, width=3 * cm, preserveAspectRatio=True)
else:
    c.rect(2 * cm, height - 3.5 * cm, 3 * cm, 2 * cm)

c.setFont("Helvetica-Bold", 18)
c.drawString(6 * cm, height - 2 * cm, "Employee Report")

```

```

c.setLineWidth(1)
c.setStrokeColorRGB(0.29, 0.56, 0.89)
c.line(2 * cm, height - 4 * cm, width - 2 * cm, height - 4 * cm)

y = height - 5 * cm
c.setFont("Helvetica", 12)
spacing = 1 * cm

c.drawString(2 * cm, y, f"Name: {name}")
y -= spacing
c.drawString(2 * cm, y, f"Position: {position}")
y -= spacing
c.drawString(2 * cm, y, f"Experience: {experience} years")
y -= spacing
c.drawString(2 * cm, y, f"Education: {education}")
y -= spacing
c.drawString(2 * cm, y, f"Vacation Days: {vacation}")
y -= spacing
c.drawString(2 * cm, y, f"Performance Rating: {rating}/100")

y -= spacing * 0.8
c.setFont("Helvetica-Bold", 13)
c.drawString(2 * cm, y, "KPI Evaluation:")
y -= spacing * 0.6
c.setFont("Helvetica")
for line in kpi_data:
    c.drawString(2.5 * cm, y, "-" + line.strip())
    y -= spacing * 0.6

c.setLineWidth(0.5)
c.setStrokeColorRGB(0, 0, 0)
c.line(2 * cm, 3 * cm, 8 * cm, 3 * cm)
c.setFont("Helvetica", 10)
c.drawString(2 * cm, 2.5 * cm, "HR Signature")

```



```
c.drawString(width - 2 * cm, 2.5 * cm, "Date: " + datetime.today().strftime("%Y-%m-%d"))
```

```
qr = qrcode.make(f'{name}, {position}, Rating: {rating}')
buffer = io.BytesIO()
qr.save(buffer)
buffer.seek(0)
qr_img = ImageReader(buffer)
c.drawImage(qr_img, width - 4 * cm, 1.5 * cm, 2 * cm, 2 * cm)

c.showPage()
c.save()
messagebox.showinfo("Report", f'Saved to: {pdf_name}')
```

```
def show_position_chart(self, filter_position_var):
    positions = []
    exclude_keywords = ["teamwork", "stress tolerance"]
    for file in os.listdir(EMPLOYEE_DIR):
        path = os.path.join(EMPLOYEE_DIR, file)
        with open(path, "r", encoding="utf-8") as f:
            lines = f.readlines()
            if len(lines) >= 2:
                position = lines[1].strip()
                lowered = position.lower()
                if not any(bad in lowered for bad in exclude_keywords):
                    positions.append(position)

    if not positions:
        messagebox.showwarning("No data", "No valid positions found.")
        return

    counter = Counter(positions)
    labels = list(counter.keys())
    sizes = list(counter.values())
```

```

def make_autopct(values):
    def my_autopct(pct):
        total = sum(values)
        count = int(round(pct * total / 100.0))
        return f'{pct:.1f}%({count})'
    return my_autopct

plt.figure(figsize=(6, 6))
plt.pie(sizes, labels=labels, autopct=make_autopct(sizes), startangle=140, colors=['#4A90E2',
'#50C878', '#F7B731', '#D81B60'])
plt.title("Employee Distribution by Position")
plt.axis('equal')
plt.show()

def show_exp_vs_performance(self, listbox, filter_position_var):
    experience = []
    ratings = []
    names = []
    keyword = filter_position_var.get().strip().lower()
    for file in os.listdir(EMPLOYEE_DIR):
        if file.endswith(".txt") and "_kpi" not in file:
            path = os.path.join(EMPLOYEE_DIR, file)
            with open(path, "r", encoding="utf-8") as f:
                lines = f.readlines()
                if len(lines) >= 6:
                    if keyword and keyword not in lines[1].lower():
                        continue
                    try:
                        experience.append(int(lines[2].strip()))
                        ratings.append(int(lines[5].strip()))
                        names.append(lines[0].strip())
                    except ValueError:
                        continue
    if not experience:
        messagebox.showwarning("No Data", "No valid data found for the selected filter.")

```

```

        return

plt.figure(figsize=(8, 6))
scatter = plt.scatter(experience, ratings, color='#4A90E2', alpha=0.5)
plt.title("Experience vs. Performance Rating")
plt.xlabel("Years of Experience")
plt.ylabel("Performance Rating (0-100)")
plt.grid(True)
try:
    z = np.polyfit(experience, ratings, 1)
    p = np.poly1d(z)
    plt.plot(experience, p(experience), "#D81B60", label="Trend Line")
    plt.legend()
except:
    pass
cursor = mplcursors.cursor(scatter, hover=True)
cursor.connect("add", lambda sel: sel.annotation.set_text(names[sel.index]))
plt.show()

def show_kpi_comparison(self, filter_position_var):
    kpi_data = {}
    keyword = filter_position_var.get().strip().lower()
    for file in os.listdir(EMPLOYEE_DIR):
        if not file.endswith("_kpi.txt"):
            continue
        employee_file = file.replace("_kpi.txt", ".txt")
        if keyword:
            try:
                with open(os.path.join(EMPLOYEE_DIR, employee_file), "r", encoding="utf-8") as ef:
                    lines = ef.readlines()
                    if len(lines) < 2 or keyword not in lines[1].lower():
                        continue
            except:
                continue
        try:
            with open(os.path.join(EMPLOYEE_DIR, file), "r", encoding="utf-8") as f:

```

```

name = file.replace("_kpi.txt", "").replace("_", " ")
lines = f.readlines()
scores = {}
for line in lines:
    if ":" in line and "Average" not in line:
        try:
            crit, score = line.split(":", 1)
            score_value = int(score.split("/")[0].strip())
            scores[crit.strip()] = score_value
        except (ValueError, IndexError):
            continue
    if scores:
        kpi_data[name] = scores
except:
    continue
if not kpi_data:
    messagebox.showwarning("No Data", "No KPI data found for the selected filter.")
    return
criteria = ["Technical Skills", "Teamwork", "Stress Tolerance"]
x = np.arange(len(criteria))
width = 0.15
plt.figure(figsize=(10, 6))
bars = []
colors = ['#4A90E2', '#50C878', '#F7B731', '#D81B60']
for i, (name, scores) in enumerate(kpi_data.items()):
    values = [scores.get(crit, 0) for crit in criteria]
    bar = plt.bar(x + i * width, values, width, label=name, color=colors[i % len(colors)])
    bars.append(bar)
plt.xlabel("KPI Criteria")
plt.ylabel("Score (0-10)")
plt.title("KPI Scores Comparison")
plt.xticks(x + width * (len(kpi_data) - 1) / 2, criteria)
plt.legend()
cursor = mplcursors.cursor(bars, hover=True)
cursor.connect("add", lambda sel: sel.annotation.set_text(

```

```

        f'{list(kpi_data.keys())[sel.artist.get_offset_position() // len(criteria)]}\n'
        f'{criteria[sel.index % len(criteria)]}: {sel.target[1]}'
    ))
    plt.tight_layout()
    plt.show()

```

```
class HRAppCore:
```

```

    def __init__(self, root):
        self.root = root
        self.root.title("HR System with KPI and Chart")
        self.root.geometry("1200x700")
        self.root.configure(bg="#F5F6F5")

```

```

        self.name_var = tk.StringVar()
        self.position_var = tk.StringVar()
        self.experience_var = tk.StringVar()
        self.education_var = tk.StringVar()
        self.vacation_days_var = tk.StringVar()
        self.rating_var = tk.StringVar()
        self.filter_position_var = tk.StringVar()

```

```

        self.file_manager = EmployeeFileManager()
        self.visualizer = ReportVisualizer()

```

```
        self.setup_ui()
```

```
def setup_ui(self):
```

```

    style = ttk.Style()
    style.theme_use('clam')
    style.configure("TNotebook", background="#F5F6F5")
    style.configure("TFrame", background="#FFFFFF")
    style.configure("TLabelframe", background="#FFFFFF", foreground="#333333")
    style.configure("TLabelframe.Label", background="#FFFFFF", foreground="#333333")
    style.configure("TLabel", background="#FFFFFF", foreground="#333333")
    style.configure("TEntry", fieldbackground="#F8F8F8")

```

```

style.configure("Accent.TButton",
                background="#4A90E2",
                foreground="white",
                padding=6,
                font=('Arial', 10))
style.map("Accent.TButton",
          background=[('active', '#357ABD')],
          foreground=[('active', 'white')])
style.configure("Secondary.TButton",
                background="#E0E6ED",
                foreground="#333333",
                padding=6)
style.map("Secondary.TButton",
          background=[('active', '#C5CED9')],
          foreground=[('active', '#333333')])

main_frame = ttk.Frame(self.root, padding=10)
main_frame.pack(fill="both", expand=True)

notebook = ttk.Notebook(main_frame)
notebook.pack(fill="both", expand=True)

input_frame = ttk.Frame(notebook, padding=10, style="TFrame")
notebook.add(input_frame, text="Employee Input")

entry_frame = ttk.LabelFrame(input_frame, text="Employee Details", padding=10)
entry_frame.grid(row=0, column=0, sticky="nsew", padx=5, pady=5)
entry_frame.configure(style="TLabelframe")

fields = [
    ("Full Name", self.name_var),
    ("Position", self.position_var),
    ("Experience (years)", self.experience_var),
    ("Education", self.education_var),
    ("Vacation Days", self.vacation_days_var),

```

```

        ("Performance Rating (0-100)", self.rating_var)
    ]

    for i, (label, var) in enumerate(fields):
        ttk.Label(entry_frame, text=label).grid(row=i//2, column=(i%2)*2, sticky="w", padx=5,
        pady=2)
        ttk.Entry(entry_frame, textvariable=var, width=30).grid(row=i//2, column=(i%2)*2+1,
        padx=5, pady=2)

    button_frame = ttk.Frame(entry_frame)
    button_frame.grid(row=3, column=0, columnspan=4, pady=10)
    ttk.Button(button_frame, text="Save", command=self.save_employee,
    style="Accent.TButton").pack(side="left", padx=5)
    ttk.Button(button_frame, text="Clear", command=self.clear_form,
    style="Secondary.TButton").pack(side="left", padx=5)

    controls_frame = ttk.Frame(notebook, padding=10, style="TFrame")
    notebook.add(controls_frame, text="Controls & Analysis")

    filter_frame = ttk.LabelFrame(controls_frame, text="Filter Employees", padding=10)
    filter_frame.grid(row=0, column=0, sticky="ew", padx=5, pady=5)

    ttk.Label(filter_frame, text="Filter by Position:").pack(side="left", padx=5)
    ttk.Entry(filter_frame, textvariable=self.filter_position_var, width=30).pack(side="left",
    padx=5)
    ttk.Button(filter_frame, text="Apply Filter", command=self.apply_filter,
    style="Accent.TButton").pack(side="left", padx=5)
    ttk.Button(filter_frame, text="Show All", command=self.load_employees,
    style="Secondary.TButton").pack(side="left", padx=5)

    analysis_frame = ttk.LabelFrame(controls_frame, text="Analysis Tools", padding=10)
    analysis_frame.grid(row=1, column=0, sticky="ew", padx=5, pady=5)

    analysis_buttons = [
        ("Generate Report", self.visualizer.generate_report),

```

```

("Calculate Bonus & Forecast", self.calculate_bonus_forecast),
("Position Chart", lambda: self.visualizer.show_position_chart(self.filter_position_var)),
("Evaluate KPI", self.evaluate_kpi),
("Simulate Scenarios", self.simulate_scenarios),
("Delete File", lambda: self.file_manager.delete_employee_file(self.listbox)),
("Exp vs. Performance", lambda: self.visualizer.show_exp_vs_performance(self.listbox,
self.filter_position_var)),
("KPI Comparison", lambda: self.visualizer.show_kpi_comparison(self.filter_position_var))
]

```

```

for i, (text, command) in enumerate(analysis_buttons):
    ttk.Button(analysis_frame, text=text, command=command,
style="Accent.TButton").grid(row=i//4, column=i%4, padx=5, pady=2, sticky="ew")

```

```

list_frame = ttk.Frame(controls_frame, style="TFrame")
list_frame.grid(row=2, column=0, sticky="nsew", padx=5, pady=5)
self.listbox = tk.Listbox(list_frame, height=15, bg="#F8FAFC", fg="#333333",
selectbackground="#4A90E2", selectforeground="white")
self.listbox.pack(fill="both", expand=True)
self.listbox.bind("<<ListboxSelect>>", self.load_employee_data)

```

```

controls_frame.grid_rowconfigure(2, weight=1)
controls_frame.grid_columnconfigure(0, weight=1)
input_frame.grid_rowconfigure(0, weight=1)
input_frame.grid_columnconfigure(0, weight=1)

```

```

self.load_employees()

```

```

def save_employee(self):
    if self.file_manager.save_employee(
        self.name_var, self.position_var, self.experience_var,
        self.education_var, self.vacation_days_var, self.rating_var
    ):
        self.clear_form()
        self.load_employees()

```



```

def clear_form(self):
    self.name_var.set("")
    self.position_var.set("")
    self.experience_var.set("")
    self.education_var.set("")
    self.vacation_days_var.set("")
    self.rating_var.set("")
    self.filter_position_var.set("")

def load_employees(self):
    self.file_manager.load_employees(self.listbox)

def load_employee_data(self, event):
    self.file_manager.load_employee_data(
        self.listbox, self.name_var, self.position_var, self.experience_var,
        self.education_var, self.vacation_days_var, self.rating_var
    )

def apply_filter(self):
    self.file_manager.apply_filter(self.listbox, self.filter_position_var)

def calculate_bonus_forecast(self):
    result = tk.Toplevel(self.root)
    result.title("Bonus & Forecast")
    result.geometry("600x400")
    result.configure(bg="#F5F6F5")
    tree = ttk.Treeview(result, columns=("Name", "Bonus", "Forecast"), show="headings")
    tree.heading("Name", text="Name")
    tree.heading("Bonus", text="Bonus (UAH)")
    tree.heading("Forecast", text="Forecast")
    tree.pack(fill="both", expand=True)

    for file in os.listdir(EMPLOYEE_DIR):
        path = os.path.join(EMPLOYEE_DIR, file)

```

```

try:
    with open(path, "r", encoding="utf-8") as f:
        lines = f.readlines()
        if len(lines) >= 6:
            name = lines[0].strip()
            exp = int(lines[2])
            edu = lines[3].strip()
            vac = int(lines[4])
            rating = int(lines[5])

            base_bonus = 3000
            performance_component = rating * 40
            experience_weight = 1 + (exp / 10)
            vacation_penalty = max(0, vac - 20) * 10
            education_bonus = 300 if 'master' in edu.lower() or 'phd' in edu.lower() else 0

            bonus = (base_bonus + performance_component) * experience_weight +
education_bonus - vacation_penalty

            if rating >= 90 and exp >= 5:
                forecast = "Promotion very likely"
            elif rating >= 80 and exp >= 3:
                forecast = "Promotion likely"
            elif rating >= 60:
                forecast = "Stable position"
            else:
                forecast = "Improvement needed"

            tree.insert("", "end", values=(name, f"{bonus:.2f}", forecast))
except Exception:
    continue

def evaluate_kpi(self):
    kpi_win = tk.Toplevel(self.root)
    kpi_win.title("Employee KPI Evaluation")

```

```

kpi_win.geometry("400x400")
kpi_win.configure(bg="#F5F6F5")

selected_idx = self.listbox.curselection()
if not selected_idx:
    tk.Label(kpi_win, text="Select an employee first.", bg="#F5F6F5").pack(pady=10)
    return

filename = self.listbox.get(selected_idx[0])
employee_path = os.path.join(EMPLOYEE_DIR, filename)
with open(employee_path, "r", encoding="utf-8") as f:
    name = f.readline().strip()

tk.Label(kpi_win, text=f"KPI Evaluation for {name}", font=("Arial", 12, "bold"),
bg="#F5F6F5").pack(pady=10)

criteria = ["Technical Skills", "Teamwork", "Stress Tolerance"]
self.kpi_vars = {}

for crit in criteria:
    frame = tk.Frame(kpi_win, bg="#F5F6F5")
    frame.pack(pady=5)
    tk.Label(frame, text=crit + ": ", bg="#F5F6F5").pack(side="left")
    var = tk.IntVar(value=5)
    self.kpi_vars[crit] = var
    tk.Scale(frame, from_=0, to=10, orient="horizontal", variable=var, bg="#F5F6F5",
highlightthickness=0).pack(side="left")

def save_kpi():
    scores = {k: v.get() for k, v in self.kpi_vars.items()}
    avg = sum(scores.values()) / len(scores)
    base = filename.replace("_kpi", "").replace(".txt", "")
    kpi_report_path = os.path.join(EMPLOYEE_DIR, base + "_kpi.txt")
    try:
        with open(kpi_report_path, "w", encoding="utf-8") as f:

```

```

        for k, v in scores.items():
            f.write(f"{k}: {v}/10\n")
        f.write(f"Average Score: {avg:.2f}/10\n")
    messagebox.showinfo("Saved", f"KPI saved for {name} at {kpi_report_path}")
    plt.figure(figsize=(6,4))
    plt.bar(scores.keys(), scores.values(), color="#4A90E2")
    plt.ylim(0, 10)
    plt.title(f'KPI Scores: {name}')
    plt.ylabel('Score')
    plt.tight_layout()
    plt.show()
except Exception as e:
    messagebox.showerror("Error", f"Failed to save KPI: {e}")

tk.Button(kpi_win, text="Save KPI", command=save_kpi, bg="#4A90E2", fg="white",
activebackground="#357ABD").pack(pady=15)

def simulate_scenarios(self):
    sim_win = tk.Toplevel(self.root)
    sim_win.title("Workforce Simulation")
    sim_win.geometry("700x500")
    sim_win.configure(bg="#F5F6F5")

    tk.Label(sim_win, text="Simulate Workforce Change Scenarios", font=("Arial", 14, "bold"),
bg="#F5F6F5").pack(pady=10)

    output_box = tk.Text(sim_win, height=20, wrap="word", bg="#F8F8F8", fg="#333333")
    output_box.pack(fill="both", expand=True, padx=10, pady=10)

    def simulate_attrition(count=1):
        files = [f for f in os.listdir(EMPLOYEE_DIR) if f.endswith(".txt") and "_kpi" not in f]
        files.sort()
        affected = files[:count]
        output_box.insert(tk.END, f"Scenario: {count} employee(s) leave\n")
        total_rating = 0

```

```

remaining = 0
for file in files:
    if file in affected:
        continue
    with open(os.path.join(EMPLOYEE_DIR, file), "r", encoding="utf-8") as f:
        lines = f.readlines()
        if len(lines) >= 6:
            try:
                rating = int(lines[5])
                total_rating += rating
                remaining += 1
            except:
                continue
    avg_rating = total_rating / remaining if remaining else 0
    output_box.insert(tk.END, f"Average rating after attrition: {avg_rating:.2f}\n\n")

def simulate_bonus_cut(percent=20):
    files = [f for f in os.listdir(EMPLOYEE_DIR) if f.endswith(".txt") and "_kpi" not in f]
    output_box.insert(tk.END, f"Scenario: Bonus cut by {percent}%\n")
    total_bonus_before = 0
    total_bonus_after = 0
    for file in files:
        path = os.path.join(EMPLOYEE_DIR, file)
        with open(path, "r", encoding="utf-8") as f:
            lines = f.readlines()
            if len(lines) >= 6:
                try:
                    exp = int(lines[2])
                    vac = int(lines[4])
                    rating = int(lines[5])
                    bonus = (rating * 50) + (exp * 100) - (5 * vac)
                    total_bonus_before += bonus
                    total_bonus_after += bonus * (1 - percent / 100)
                except:
                    continue

```

```

diff = total_bonus_before - total_bonus_after
output_box.insert(tk.END, f"Total bonus before: {total_bonus_before:.2f} UAH\n")
output_box.insert(tk.END, f"Total bonus after : {total_bonus_after:.2f} UAH\n")
output_box.insert(tk.END, f"Saved budget      : {diff:.2f} UAH\n\n")

btn_frame = tk.Frame(sim_win, bg="#F5F6F5")
btn_frame.pack(pady=5)

tk.Button(btn_frame, text="Simulate 1 Resignation", command=lambda: simulate_attrition(1),
bg="#4A90E2", fg="white", activebackground="#357ABD").pack(side="left", padx=5)
tk.Button(btn_frame, text="Simulate 2 Resignations", command=lambda: simulate_attrition(2),
bg="#4A90E2", fg="white", activebackground="#357ABD").pack(side="left", padx=5)
tk.Button(btn_frame, text="Simulate 20% Bonus Cut", command=lambda:
simulate_bonus_cut(20), bg="#4A90E2", fg="white",
activebackground="#357ABD").pack(side="left", padx=5)

if __name__ == "__main__":
    root = tk.Tk()
    app = HRAppCore(root)
    root.mainloop()

```

Додаток Г – Графічна частина

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення

Розробка модулю інформаційної системи для керування даними персоналу підприємства

Виконав:
студент 4 курсу групи 4ПІ-216 Новицький О.М.
Керівник:
к.т.н., доц. каф. ПЗ Ракитянська Г.Б.

Рисунок Г.1 – Титульний слайд

Мета, об'єкт і предмет дослідження

- **Мета дослідження.** Метою роботи є підвищення ефективності управління персоналом через створення інтерактивного модуля для аналізу, прогнозування та візуалізації даних співробітників.
- **Об'єкт дослідження.** Процес управління даними персоналу в інформаційних системах.
- **Предмет дослідження.** Методи та засоби інтерактивного управління даними співробітників, оцінки KPI, прогнозування бонусів і візуалізації результатів.

Рисунок Г.2 – Мета, об'єкт і предмет дослідження

Новизна і практична цінність

- **Новизна отриманих результатів.**

1. Алгоритм прогнозування кар'єрного росту працівників на основі продуктивності та досвіду. Функція calculate bonus forecast реалізує модель, що враховує рейтинг продуктивності, досвід, освіту та відпустки для розрахунку бонусу та прогнозу на майбутнє. Такий підхід дозволяє приймати обґрунтовані HR-рішення з використанням простого евристичного аналізу.
2. Симуляція управлінських сценаріїв (звільнення працівників, скорочення бонусів). Функція simulate scenarios дозволяє HR-аналітикам змодельовати наслідки управлінських рішень та прогнозувати як зміниться середній рейтинг після звільнення працівників або який буде ефект від скорочення бонусів.

- **Практична цінність отриманих результатів.** Полягає в тому, що на основі отриманих в бакалаврській кваліфікаційній роботі теоретичних положень запропоновано оптимізацію управління персоналом, аналізу продуктивності, прогнозування кадрових рішень і створення аналітичних звітів.

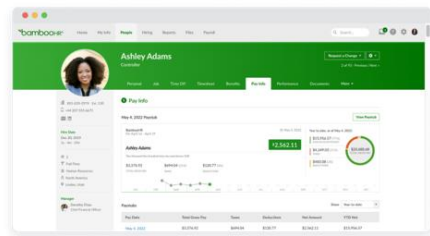
Рисунок Г.3 – Наукова новизна та практична цінність

ЗАДАЧІ БАКАЛАВРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ

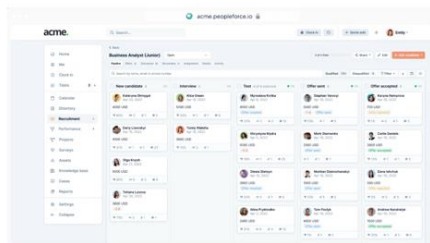
- Запропонувати нові методи підвищення інтерактивності введення, фільтрації та аналізу даних співробітників.
- Створити наочні візуалізації (графіки розподілу посад, досвіду проти продуктивності, порівняння KPI).
- Розробити програмні компоненти та модуль інформаційної системи.
- Провести тестування розроблених засобів.
- Провести аналіз існуючих методів і засобів управління даними персоналу для визначення напрямків підвищення інтерактивності.

Рисунок Г.4 – Задачі бакалаврської кваліфікаційної роботи

АНАЛОГИ



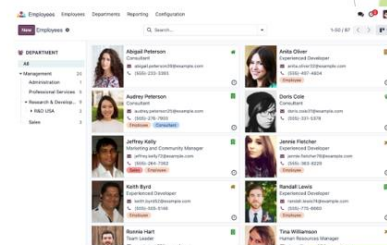
BambooHR



People Force



Workday HCM



Odoo HR

Рисунок Г.5 – Аналоги

ПОРІВНЯЛЬНИЙ АНАЛІЗ АНАЛОГІВ

	Bamboo HR	Workday HCM	PeopleFor ce	Odoo HR	Власна розробка
Інтеграція з іншими системами підприємства	1	1	1	1	1
Функціонал для аналітики персоналу	0	1	1	0	1
Автоматизація кадрового документообігу	0	1	0	1	1
Створення візуалізацій для аналізу персоналу	1	1	1	1	1
Можливість роботи в офлайн-режимі	0	0	1	0	1
Моделювання управлінських сценаріїв	0	1	0	0	1
Сумарний коефіцієнт	32%	84%	66%	50%	100%

Рисунок Г.6 – Порівняльний аналіз аналогів

ОБРАНІ ЗАСОБИ ДЛЯ РЕАЛІЗАЦІЇ

- Python - універсальність, простота синтаксису, широка бібліотечна база.
- tkinter - створення графічного інтерфейсу, кросплатформність.
- Matplotlib - інтерактивні візуалізації (графіки, гістограми).
- reportlab - генерація PDF-звітів із QR-кодами.
- Причини вибору: Модульність, продуктивність, інтеграція компонентів, велика кількість доступних бібліотек для обробки даних та створення візуалізацій, підтримка кросплатформного виконання.

Рисунок Г.7 – Порівняльний аналіз аналогів

ФОРМУЛИ РОБОТИ СИСТЕМИ

1. Розрахунок бонусу

$$B = (3000 + 40 \cdot R) \cdot \left(1 + \frac{E}{10}\right) + \delta \cdot edu \cdot 300 - 10 \cdot \max(0, V - 20) \quad (2.1)$$

B – розмір бонусу в гривнях; R – рейтинг продуктивності працівника, отриманий з оцінки його ефективності; E – досвід роботи в роках, що відображає стаж працівника; V – кількість використаних відпусткових днів за рік; $\max(0, V - 20)$ – функція, яка обчислює штраф за перевищення ліміту в 20 відпусткових днів.

3. Економія від скорочення бонусів

$$S = \sum_{i=1}^n B_i - \sum_{i=1}^n (0.8 \times B_i) = 0.2 \times \sum_{i=1}^n B_i \quad (2.3)$$

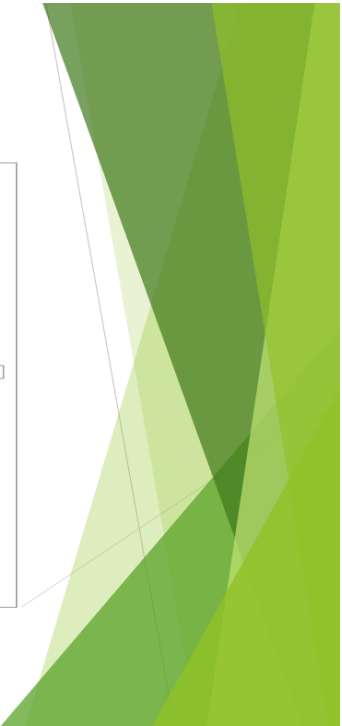
S – економія від скорочення бонусів; B_i – початковий бонус, i -го працівника, за формолою 2.1.

2. Середній рейтинг

$$A = \frac{\left(\sum_{i=1}^n R_i\right)}{n} \quad (2.2)$$

A – середній рейтинг продуктивності; R_i [0,100] – рейтинг продуктивності; n – кількість працівників.

Рисунок Г.8 – Формули роботи системи



СТАРТОВЕ ВІКНО ЗАСТОСУНКУ

HR System with KPI and Chart

Employee Input Controls & Analysis

Employee Details

Full Name		Position	
Experience (years)		Education	
Vacation Days		Performance Rating (0-100)	

Рисунок Г.13 – Стартове вікно застосунку

Інтерфейс модуля симуляції сценаріїв змін у структурі персоналу

Workforce Simulation

Simulate Workforce Change Scenarios

Scenario: 1 employee(s) leave
Average rating after attrition: 88.44

Scenario: Bonus cut by 20%
Total bonus before: 47345.00 UAH
Total bonus after : 37876.00 UAH
Saved budget : 9469.00 UAH

Рисунок Г.14 – Інтерфейс модуля симуляції

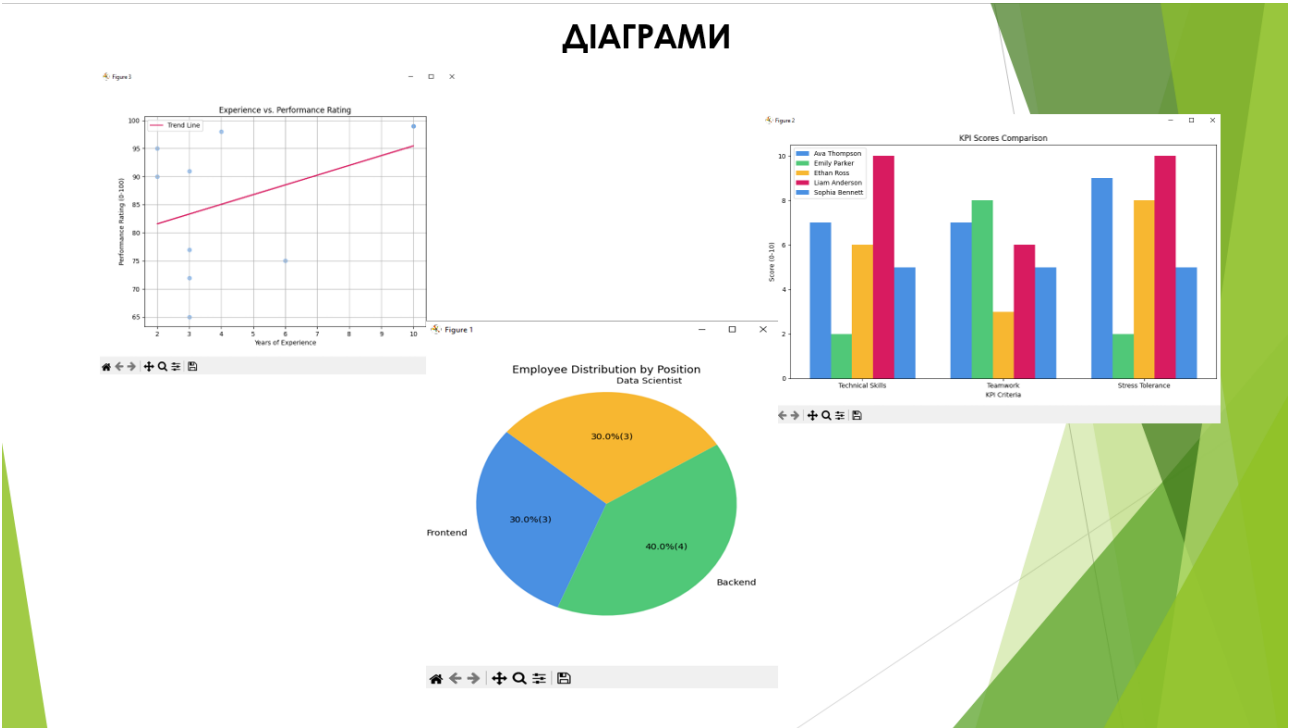


Рисунок Г.15 – Діаграми

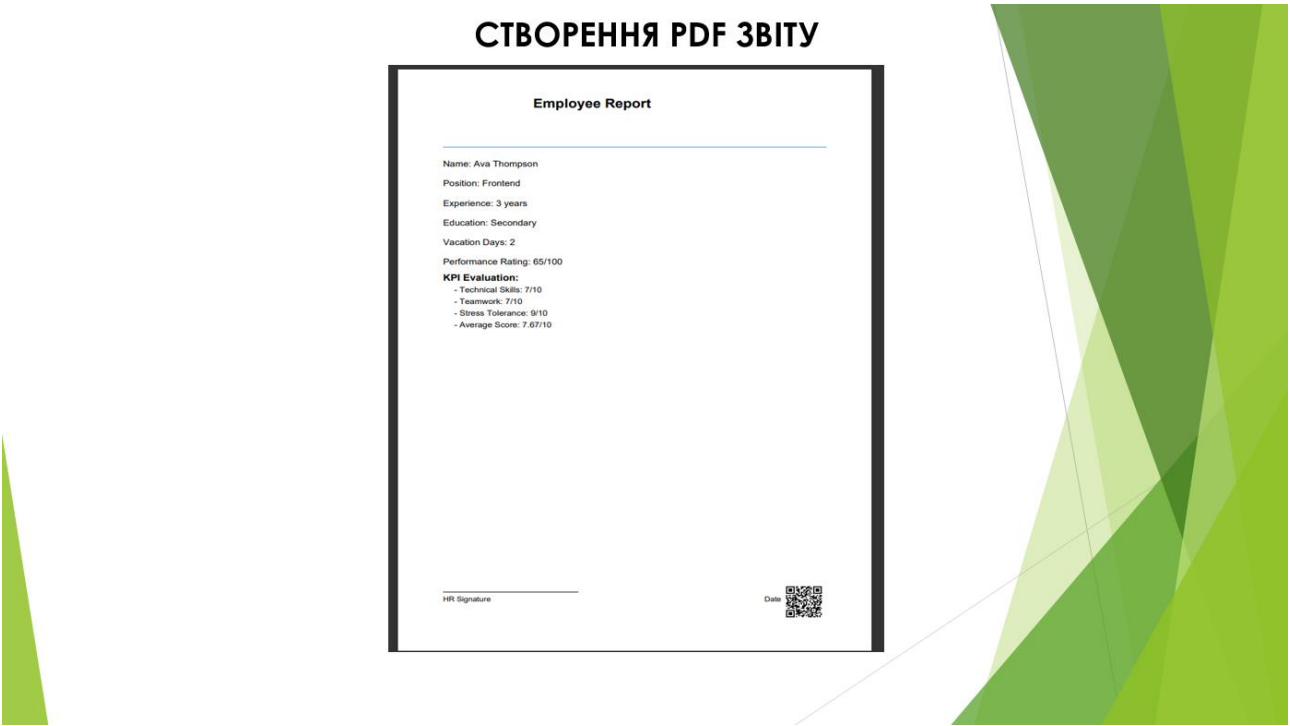


Рисунок Г.16 – Створення PDF звіту

ПУБЛІКАЦІЇ

- Основні результати досліджень опубліковано в матеріалах конференції «LIV Всеукраїнська науково-технічна конференція підрозділів Вінницького національного технічного університету (2025)»
- Новицький О.М. Розробка програмних модулів інформаційної системи для керування персоналом підприємства. Матеріали LIV Всеукраїнської науково-технічної конференції професорсько-викладацького складу, науковців, аспірантів та студентів підрозділів університету з участю працівників підприємств (НТКП ВНТУ-2025). Вінниця, 2025. URL: <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2025/paper/view/24291>

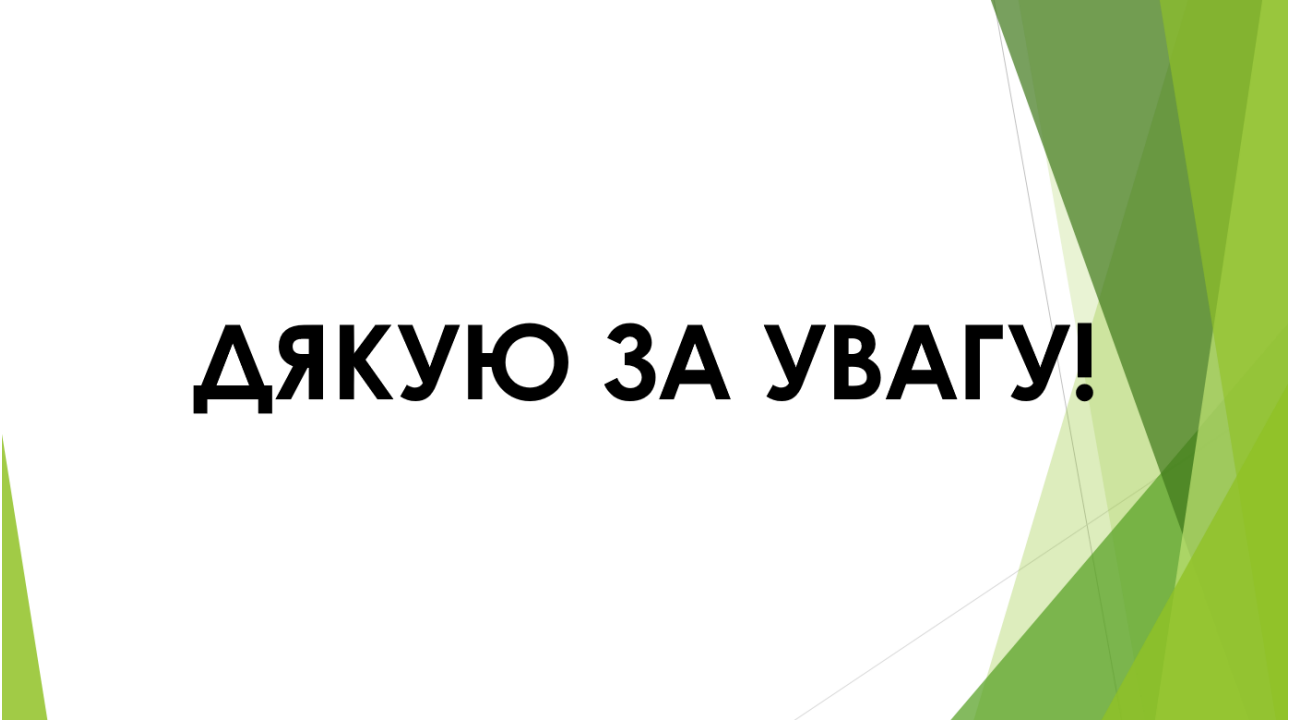
Рисунок Г.17 – Публікації

ВИСНОВКИ

Було виконано такі задачі:

- Проведено грунтовний аналіз існуючих HR-систем, що дозволило обґрунтувати розробку власного модуля з урахуванням їхніх сильних і слабких сторін.
- Успішно розроблено підсистеми для введення даних, оцінки KPI, прогнозування бонусів і створення інтерактивних візуалізацій, які значно полегшують процес управління персоналом.
- Тестування системи підтвердило її стабільність, коректність роботи та високу продуктивність у реальних умовах, що свідчить про готовність до практичного застосування.
- Модуль продемонстрував свою практичну цінність, забезпечуючи оптимізацію управління персоналом, точне прогнозування кадрових рішень і підтримку освітнього процесу шляхом інтеграції аналітичних інструментів.

Рисунок Г.18 – Висновки



ДЯКУЮ ЗА УВАГУ!

Рисунок Г.19 – Кінець презентації