

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

Автоматизації і інформаційних технологій

(факультет)

Інформаційних технологій

(назва кафедри)

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО КВАЛІФІКАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ МАГІСТР**

на тему:

«Комп'ютерна система для оцінки якості та ефективності виконання
роботи працівниками підприємства»

Галацан Дмитро Миколайович

(прізвище, ім'я та по батькові студента повністю)

Київ 2024 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

Автоматизації і інформаційних технологій

(факультет)

Інформаційних технологій

(назва кафедри)

ЗАТВЕРДЖУЮ

Завідувачка кафедри

д. т. н., доцент Гончаренко Т. А.

„___” _____ 2024 року

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО КВАЛІФІКАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ НА ЗДОБУТТЯ
ОСВІТНЬОГО СТУПЕНЯ МАГІСТР**
Комп'ютерна система для оцінки якості та ефективності виконання
роботи працівниками підприємства
(назва)

Виконав:

Галацан Дмитро Миколайович

(прізвище, ім'я та по батькові повністю)

122. Комп'ютерні науки

(спеціальність)

Комп'ютерні науки

(освітня програма)

Групи КНМ-23

Керівники Гончаренко Т.А (1),

Мацієвський О. О.(2)

(прізвище та ініціали)

д.т.н., доцент. (1), ас (2)

(вчене звання, науковий ступінь)

Ідентичність підтверджую

Київ 2024 р.

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

БУДІВНИЦТВА І АРХІТЕКТУРИ

Факультет: Автоматизації і інформаційних технологій

Випускова кафедра: Інформаційних технологій

Освітній рівень: «магістр»

Спеціальність: 122 «Комп'ютерні науки»

Освітня програма: Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувачка кафедри

д. т. н., доцент Гончаренко Т. А.

„___” _____ 2024 року

ЗАВДАННЯ

ДО ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ МАГІСТР

Галацан Дмитро Миколайович

(прізвище, ім'я та по батькові студента)

1. Тема роботи

Комп'ютерна система для оцінки якості та ефективності виконання роботи працівниками підприємства

затверджена наказом ректора КНУБА № 2213/2 від 08.10.24

2. Керівник роботи.: Гончаренко Т. А., д. т. н., доцент, Мацієвський О. О., ас
(прізвище, ініціали, науковий ступінь, вчене звання)

3. Строк подання студентом роботи до захисту 01.12.2024

4. Зміст пояснювальної записки за розділами:

Р. 1. Стан проблеми оцінки якості та ефективності роботи працівників

Р. 2. Теоретичні основи поставленої задачі

Р. 3. Проектування системи оцінки ефективності роботи працівників

Р. 4. Практичне впровадження та тестування системи

Р. 5. Економічне обґрунтування проекту

5. Графічний матеріал за розділами:

С. 1. Актуальність дослідження

С. 2. Аналіз існуючих систем

С. 3. Проектування комп'ютерної системи

С. 4. Програмне забезпечення

6. Календарний план виконання роботи:

Види робіт та їх зміст	Дата виконання
Розділ 1	28.07.2024
Розділ 2	23.09.2024
Розділ 3	18.10.2024
Розділ 4	16.11.2024
Остаточне оформлення роботи	26.11.2024
Направлення роботи для перевірки на плагіат	01.12.2024
Попередній захист роботи на випусковій кафедрі	01.12.2024
Направлення роботи на рецензування	02.12.2024

7. Консультанти розділів кваліфікаційної випускної роботи

Розділ	Прізвище, ініціали та посада консультанта	Перевірів	
		дата	підпис
Розділ 1	Гончаренко Т. А.	01.08.2024	
Розділ 2	Гончаренко Т. А.	25.09.2024	
Розділ 3	Мацієвський О. О.	20.10.2024	
Розділ 4	Мацієвський О. О.	20.11.2024	

8. Дата видачі завдання 08.09.2024

Зав. кафедри _____ Гончаренко Т. А.
(підпис) (прізвище та ініціали)

Керівник (1) _____ Гончаренко Т. А.
(підпис) (прізвище та ініціали)

Керівник (2) _____ Мацієвський О. О.
(підпис) (прізвище та ініціали)

Здобувач _____ Галацан Д. М.
(підпис) (прізвище та ініціали)

РЕЗЮМЕ (SUMMARY)

РЕЗЮМЕ (SUMMARY) до кваліфікаційної випускної роботи Здобувач:	Галацан Дмитро Миколайович Galatsan Dmytro Mykolayovych		
ЗВО	Київський національний університет будівництва і архітектури		
Тема (українською та англійською)	Комп'ютерна система для оцінки якості та ефективності виконання роботи працівниками підприємства Computer system for assessing the quality and efficiency of work performed by employees		
Освітній ступінь	Магістр		
Факультет	Автоматизації і інформаційних технологій		
Випускаюча кафедра	Інформаційних технологій		
Спеціальність	122 «Комп'ютерні науки»		
Освітня програма	Комп'ютерні науки		
Керівники	Гончаренко Т. А. та Мацієвський О. О.		
Обсяг роботи:	пояснювальна записка, стор.	розділів	слайди презентації
	104	5	10
Ключові слова:	Комп'ютерна система, Ключові показники ефективності, Big Data		
Keywords:	Computer system, Key performance indicators, Big Data		

Робота присвячена розробці автоматизованої системи для оцінки якості та ефективності виконання роботи працівників підприємства. Розроблена система використовує методи машинного навчання та багатокритеріального аналізу для об'єктивного оцінювання продуктивності працівників, формування персоналізованих рекомендацій і забезпечення прозорості процесів управління.

This paper is devoted to the development of an automated system for assessing the quality and efficiency of the work of enterprise employees. The developed system uses machine learning and multicriteria analysis methods to objectively assess employee performance, generate personalized recommendations, and ensure transparency of management processes.

Здобувач: _____ / Дмитро ГАЛАЦАН /
 Керівник (1): _____ / Тетяна ГОНЧАРЕНКО /
 Керівник (2): _____ / Олексій МАЦІЄВСЬКИЙ /
 “ ____ ” _____ 2024 р.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1. СТАН ПРОБЛЕМИ ОЦІНКИ ЯКОСТІ ТА ЕФЕКТИВНОСТІ РОБОТИ ПРАЦІВНИКІВ	11
1.1. Огляд існуючих систем	12
1.2. Аналіз недоліків сучасних рішень.....	14
1.3. Вимоги до системи оцінки якості та ефективності роботи працівників	1
Ошибка! Закладка не определена.	
1.3.1. Функціональні вимоги системи оцінки	16
1.3.2. Нефункціональні вимоги системи оцінки	18
1.4. Рівні цифрової грамотності.....	19
1.5. Характеристика підприємства як об'єктів автоматизації	20
1.5.1. Особливості процесів управління ефективністю	22
1.5.2. Ключові показники оцінки роботи працівників	23
1.6. Постановка проблеми	25
1.6.1. Основні виклики	25
1.6.2. Обмеження	26
РОЗДІЛ 2. ТЕОРЕТИЧНІ ОСНОВИ ПОСТАВЛЕНОЇ ЗАДАЧІ.....	28
2.1. Моделі оцінки якості роботи працівників	29
2.2. Методології оцінювання ключових показників ефективності....	30
2.3. Моделі багатокритеріального аналізу.....	32
2.4. Методи збору обробки та аналізу даних	33
2.5. Методи збору обробки та аналізу великих даних.....	37
2.6. Стандарти управління якістю та ефективністю.....	39
2.7. Інтеграція автоматизованих систем у бізнес процеси.....	41

2.8. Економічний і соціальний вплив автоматизації оцінки роботи працівників	42
2.9. Кількісний аналіз характеристик стандартів оцінки ефективності	43
2.10. Дизайн бази даних	45
РОЗДІЛ 3. Проектування системи оцінки ефективності роботи працівників.....	47
3.1. Постановка задачі для програмної реалізації	Ошибка! Закладка не определена.
3.2. Архітектура системи.....	Ошибка! Закладка не определена.
3.3. Загальна структура.....	49
3.4. Компоненти та їх взаємодія	50
3.5. Вибір технологій для реалізації.....	50
3.5.1. Аналіз мов програмування	51
3.5.2. Середовище розробки	
3.5.3. Спеціалізовані бібліотеки та фреймворки	52
3.6. Проектування бази даних	53
3.6.1. Структура бази даних	53
3.6.2. Моделі даних (ER-діаграми)	54
3.7. Інтерфейс користувача	56
3.7.1. UX/UI-дизайн	57
3.7.2. Створення прототипів	59
3.7.3. UX/UI-прототип	59
3.7.4. Використання HTML-прототипу в системі	61
3.8. Забезпечення безпеки даних	63
3.9. Масштабованість і продуктивність системи	64
3.10. Тестування системи	65
РОЗДІЛ 4 Практичне впровадження та тестування системи	68
4.1. Вибір технологій для реалізації.....	68
4.1.1. Аналіз мов програмування	68

4.1.2. Середовище розробки	70
4.1.3. Спеціалізовані бібліотеки та фреймворки	71
4.1.4. Підсумки вибору технологій	72
4.2. Реалізація компонентів системи.....	73
4.2.1. Опис основного коду	73
4.2.2. Логіка функціонування системи	78
4.3. Інтеграція з іншими системами	80
4.3.1. API та веб-сервіси	80
4.3.2. Методи обміну даними	81
4.4. Тестування системи	85
4.4.1. Методи тестування	85
4.4.2. Аналіз результатів тестування	86
РОЗДІЛ 5 Економічне обґрунтування проекту.....	89
5.1. Вартість розробки та впровадження системи	89
5.2. Оцінка економічного ефекту	90
5.2.1. Підвищення продуктивності	90
5.2.2. Зниження витрат	91
ВИСНОВКИ	93
Список використаних джерел.....	94
Додатки	98
Додаток	А
.....	О

шибка! Закладка не определена.

ВСТУП

У сучасному світі, де цифровізація підприємств стає обов'язковою умовою конкурентоспроможності, ефективна оцінка якості та результативності роботи працівників є одним із ключових чинників успіху організації. Зростання складності бізнес-процесів і обсягів даних, що потребують аналізу, створює додаткове навантаження на керівників і HR-служби, які відповідають за моніторинг продуктивності та прийняття управлінських рішень.

Традиційні методи оцінювання, засновані на ручному зборі даних або суб'єктивній оцінці, стають дедалі менш ефективними. Це зумовлено як зростанням кількості працівників і задач, так і необхідністю швидкої та об'єктивної оцінки. У таких умовах виникає потреба в автоматизованих рішеннях, які здатні збирати, аналізувати й інтерпретувати інформацію про діяльність працівників, забезпечуючи об'єктивність і прозорість процесу оцінювання.

Розробка комп'ютерної системи, що дозволяє автоматизувати процес оцінки якості та ефективності роботи співробітників, є важливим кроком до оптимізації управлінських рішень. Такі системи дозволяють не лише підвищити точність аналізу, але й забезпечити своєчасне реагування на потенційні проблеми, сприяти підвищенню мотивації працівників і зростанню загальної ефективності підприємства.

Актуальність

теми

дослідження

Оцінка якості та ефективності роботи працівників є ключовим завданням сучасного управління підприємствами. У швидкозмінних умовах ринку, коли конкуренція постійно зростає, підприємства змушені впроваджувати інноваційні підходи для забезпечення високої продуктивності та якісного виконання завдань. Традиційні методи оцінювання, засновані на суб'єктивній думці керівництва або періодичних оглядах, втрачають свою ефективність через низьку точність і відсутність оперативності.

Завдяки сучасним інформаційним технологіям з'явилася можливість автоматизації процесів оцінки роботи працівників. Використання комп'ютерних систем дозволяє підвищити об'єктивність аналізу, забезпечити прозорість процесів, а також своєчасно виявляти проблеми й реагувати на них. Розробка такої системи актуальна для підприємств різних галузей, адже вона сприяє оптимізації процесів управління, мотивує працівників до покращення своїх показників і дозволяє приймати обґрунтовані управлінські рішення.

Особливе значення тема набуває в умовах цифровізації, коли дані про діяльність працівників можна збирати, аналізувати та використовувати для підвищення ефективності роботи всього підприємства. Тому розробка комп'ютерної системи, яка здатна автоматизувати оцінку якості та ефективності роботи, є актуальним і практично значущим завданням.

Аналіз останніх досліджень та наукових праць

Останніми роками значну увагу дослідників привертають питання автоматизації процесів оцінки продуктивності працівників і розробка інтелектуальних систем для управління ефективністю. У багатьох працях наголошується, що традиційні методи оцінювання, засновані на опитуваннях, звітах і регулярних оцінках, не відповідають сучасним викликам бізнесу.

Наукові дослідження, присвячені використанню інформаційних технологій для аналізу продуктивності, зосереджені на розробці алгоритмів для обробки великих обсягів даних, моделюванні ключових показників ефективності (KPI) та інтеграції штучного інтелекту для автоматизації оцінки. Роботи провідних учених у цій галузі підкреслюють важливість адаптації систем до потреб конкретних підприємств, зокрема, їх гнучкість і можливість врахування індивідуальних особливостей бізнес-процесів.

Дослідження показують, що застосування автоматизованих систем для моніторингу ефективності сприяє зростанню мотивації працівників, зменшенню ризику суб'єктивності в оцінюванні та підвищенню оперативності управлінських рішень. Проте, у багатьох роботах наголошується на недоліках існуючих рішень, таких як складність впровадження, висока вартість і недостатня інтеграція з

іншими бізнес-системами. Це визначає необхідність подальших досліджень і розробок у цій сфері.

Мета та завдання роботи

Метою даної кваліфікаційної роботи є розробка комп'ютерної системи для оцінки якості та ефективності виконання роботи працівниками підприємства. Система має забезпечувати автоматизацію збору, обробки та аналізу даних про продуктивність працівників, формування звітності, а також створення рекомендацій для підвищення ефективності.

Для досягнення цієї мети необхідно вирішити низку завдань:

- провести аналіз існуючих підходів і систем для оцінки якості роботи;
- визначити вимоги до комп'ютерної системи, що розробляється;
- дослідити теоретичні основи методів оцінювання продуктивності;
- розробити архітектуру системи та спроектувати її основні компоненти;
- реалізувати програмне забезпечення на основі сучасних технологій;
- провести тестування системи для виявлення її переваг і недоліків;
- обґрунтувати економічну ефективність розробленого рішення.

Об'єкт і предмет дослідження

Об'єктом дослідження є процеси управління якістю та ефективністю роботи працівників підприємства. У фокусі дослідження – засоби та методи автоматизації цих процесів за допомогою комп'ютерних систем.

Предметом дослідження виступають алгоритми, моделі та інформаційні технології, що застосовуються для створення комп'ютерних систем оцінки продуктивності працівників, а також особливості їх інтеграції у бізнес-процеси підприємства.

Методи дослідження

Для досягнення мети та виконання поставлених завдань у роботі використовуються комплексні методи дослідження. Теоретичний аналіз включає огляд літератури, нормативних документів і сучасних досліджень з теми автоматизації оцінки продуктивності. Емпіричні методи передбачають вивчення

досвіду використання подібних систем на практиці, аналіз потреб підприємств і моделювання процесів оцінки.

Для проектування та реалізації системи застосовуються методи об'єктно-орієнтованого програмування, моделювання баз даних, а також сучасні фреймворки для розробки веб-додатків. Експериментальні методи включають тестування програмного забезпечення та оцінку його ефективності в реальних умовах.

Методи дослідження

Дослідження базується на використанні комплексу методів, що охоплюють як теоретичний, так і практичний аспекти. У роботі використано:

1. **Аналіз літератури та нормативної бази** для вивчення сучасних підходів до оцінювання ефективності роботи працівників, існуючих моделей і алгоритмів.
2. **Моделювання бізнес-процесів** для визначення вимог до системи та формування ключових функцій, які вона має виконувати.
3. **Методи обробки даних**, зокрема машинного навчання, для аналізу й автоматизації процесів оцінювання.
4. **Методи об'єктно-орієнтованого проектування**, які використовувалися для розробки архітектури програмного забезпечення.
5. **Експериментальні методи**, що включають тестування розробленої системи на реальних даних підприємства.
6. **Порівняльний аналіз**, який дозволив оцінити переваги розробленої системи порівняно з існуючими аналогами.

Практичне значення роботи

Результати дослідження мають значний практичний потенціал. Розроблена комп'ютерна система дозволяє автоматизувати процеси оцінки якості та ефективності працівників, що:

- скорочує час, необхідний для обробки інформації про діяльність співробітників;
- підвищує точність аналізу та мінімізує вплив суб'єктивного фактору;

- сприяє своєчасному виявленню проблемних зон і запровадженню коригувальних заходів;
- забезпечує прозорість процесу оцінки, що позитивно впливає на мотивацію працівників;
- інтегрується з іншими бізнес-системами, спрощуючи управління процесами на підприємстві.

Система може бути адаптована для використання на підприємствах різного масштабу та галузей, що робить її універсальним інструментом для підвищення ефективності управління людськими ресурсами.

Наукова новизна роботи

Наукова новизна роботи полягає в комплексному підході до автоматизації процесів оцінювання продуктивності працівників. Основні аспекти, що визначають новизну:

- запропонована архітектура комп'ютерної системи, яка поєднує сучасні технології машинного навчання для аналізу великих обсягів даних з інтерактивним інтерфейсом користувача;
- розроблено алгоритми для інтеграції оцінювання KPI з прогнозуванням можливих змін у продуктивності працівників;
- реалізовано підхід до динамічного формування рекомендацій для підвищення ефективності роботи як окремих працівників, так і підрозділів підприємства в цілому;
- запропоновано методику тестування та оцінки ефективності роботи системи, яка враховує специфіку галузі, в якій застосовується система.

РОЗДІЛ 1.

СТАН ПРОБЛЕМИ ОЦІНКИ ЯКОСТІ ТА ЕФЕКТИВНОСТІ РОБОТИ ПРАЦІВНИКІВ

Оцінка якості та ефективності роботи працівників [1] є однією з найважливіших складових управління будь-яким підприємством. У сучасному бізнес-середовищі, що характеризується високою конкуренцією та динамічними змінами, підприємства змушені приділяти все більше уваги оптимізації використання людських ресурсів. Успіх компанії залежить не лише від якісних продуктів або послуг, які вона надає, а й від рівня ефективності її працівників. Саме тому питання автоматизації та вдосконалення процесів оцінки продуктивності співробітників набуває стратегічного значення [2].

Сучасні реалії ставлять перед організаціями нові виклики: необхідність швидкої адаптації до змін, зростання конкуренції на ринку праці та високі вимоги клієнтів до якості послуг і продуктів. В умовах таких змін традиційні методи оцінки роботи працівників, засновані на суб'єктивних оцінках керівництва, стають дедалі менш ефективними. Суб'єктивність, упередженість і нерідко застарілі підходи до оцінювання часто призводять до зниження мотивації співробітників і, як наслідок, до зменшення загальної продуктивності організації [3].

Особливого значення набуває розвиток цифрових технологій, які дають можливість автоматизувати багато рутинних завдань і забезпечити об'єктивний аналіз продуктивності працівників [4]. Використання сучасних комп'ютерних систем для оцінки якості та ефективності роботи дозволяє отримати дані в реальному часі, виявити слабкі місця в роботі окремих співробітників або команд, а також запропонувати рекомендації для підвищення продуктивності.

Проте впровадження таких систем потребує вирішення багатьох проблем, включаючи вибір відповідної методології оцінювання, розробку показників ефективності (KPI) [5], забезпечення прозорості оцінок і збереження конфіденційності даних працівників. Більшість існуючих систем намагаються

вирішити ці питання, однак мають ряд обмежень, які ускладнюють їх впровадження на практиці [6].

Даний розділ присвячено аналізу стану проблеми оцінки роботи працівників у сучасних умовах. У ньому буде розглянуто сучасні підходи до автоматизації цього процесу, проведено огляд популярних систем оцінювання, їх переваг і недоліків, а також сформульовано основні виклики, що стоять перед розробниками таких систем [7].

Таким чином, основна мета цього розділу полягає в ідентифікації проблем, які потребують вирішення для створення ефективної комп'ютерної системи оцінки якості та ефективності роботи працівників [8]. Особлива увага приділяється необхідності поєднання інноваційних технологій, таких як машинне навчання, аналітика великих даних і інтерактивні інтерфейси, з існуючими бізнес-процесами підприємства [9].

1.1.Огляд існуючих систем

Управління ефективністю працівників є однією з ключових задач будь-якої організації, незалежно від її розміру чи галузі діяльності. Сучасні системи оцінки ефективності [10] покликані забезпечити прозорість процесів моніторингу продуктивності, зменшити суб'єктивність у прийнятті рішень і допомогти керівникам своєчасно реагувати на зниження ефективності.

На ринку програмного забезпечення існує широкий спектр систем для автоматизації процесу оцінки роботи працівників [11]. Основними представниками цього сегменту є такі платформи, як **SAP SuccessFactors**, **Oracle HCM**, **Workday**, **BambooHR**, **Zoho People** та інші [12-13].

SAP SuccessFactors – це комплексна HRM-платформа, яка забезпечує управління ефективністю працівників за допомогою аналізу даних у реальному часі. Вона дозволяє відстежувати ключові показники ефективності (KPI), проводити регулярні оцінки та надавати керівникам рекомендації щодо покращення продуктивності. Вбудовані функції прогнозування допомагають

виявляти ризики, пов'язані з плинністю кадрів, і приймати рішення на основі даних [14].

Oracle HCM Cloud [15] є ще однією популярною платформою для управління персоналом. Вона інтегрує в собі функції оцінки продуктивності, навчання, розвитку працівників і планування їхньої кар'єри. Завдяки використанню аналітичних інструментів, система дозволяє прогнозувати продуктивність, оцінювати залученість працівників і пропонувати заходи для підвищення ефективності.

Workday [16]– це сучасна HR-платформа, орієнтована на підприємства середнього та великого бізнесу. Її ключовими особливостями є підтримка мультинаціональних організацій, інтеграція з ERP-системами та можливість створення індивідуальних метрик оцінки продуктивності.

BambooHR [17] зосереджена на автоматизації рутинних завдань, таких як управління даними працівників і відстеження ефективності. Її функціонал включає формування звітів, управління графіками, а також проведення регулярних опитувань для отримання зворотного зв'язку від співробітників.

Zoho People [18] – це більш доступне рішення для невеликих компаній, яке пропонує функції оцінки продуктивності, управління робочим часом і аналізу зворотного зв'язку. Система дозволяє керівникам створювати персоналізовані метрики оцінки, що є важливим для малих підприємств.

Додатково варто згадати спеціалізовані рішення, які інтегрують штучний інтелект у процес управління ефективністю. Наприклад, **Leapsome** та **15Five** активно використовують алгоритми машинного навчання для аналізу продуктивності працівників, виявлення тенденцій і формування рекомендацій.

Також існують окремі інструменти для моніторингу KPI, такі як **Microsoft Power BI** або **Tableau**, які дозволяють керівникам самостійно налаштовувати метрики, візуалізувати дані та проводити глибокий аналіз [19].

Хоча існуючі системи мають значний функціонал, їх використання обмежене певними факторами, які потребують окремого розгляду.

1.2. Аналіз недоліків сучасних рішень

Попри широкі можливості існуючих систем оцінки продуктивності, вони мають низку недоліків, які можуть негативно впливати на їх впровадження та ефективність [20].

- **Складність інтеграції** Більшість сучасних платформ, таких як **SAP SuccessFactors** або **Oracle HCM**, орієнтовані на великі підприємства. Їх інтеграція з іншими бізнес-системами потребує значних фінансових та часових витрат. Наприклад, синхронізація даних із CRM- або ERP-системами може вимагати участі фахівців із розробки [21]. Це ускладнює використання подібних платформ для малого та середнього бізнесу.
- **Висока вартість** Ліцензування провідних систем є надзвичайно дорогим. Наприклад, вартість підписки на **Workday** або **SAP SuccessFactors** може сягати десятків тисяч доларів на рік. Для середніх і малих підприємств такі витрати є неприйнятними, особливо якщо врахувати необхідність навчання персоналу та адаптації системи [22].
- **Обмежена гнучкість** Деякі системи мають фіксований набір метрик для оцінки продуктивності, які не завжди враховують специфіку діяльності підприємства. Наприклад, компанії з творчою або науковою спрямованістю потребують індивідуального підходу до оцінювання, який часто неможливо реалізувати в рамках стандартних функцій платформи [23].
- **Низька прозорість** Оцінка роботи працівників часто базується на алгоритмах, які є непрозорими для користувачів. Це може викликати недовіру з боку працівників, які не розуміють, як саме оцінюються їхні показники. Зокрема, у великих системах, таких як **Oracle HCM**, обчислення метрик може бути надто складним для пояснення безпосередньо співробітникам [24].
- **Недостатність використання штучного інтелекту** Багато систем обмежуються базовим аналізом даних без використання сучасних алгоритмів машинного навчання. Це не дозволяє виявляти приховані

закономірності в продуктивності працівників і прогнозувати їхню ефективність у майбутньому.

- **Складність адаптації до локальних умов** Інструменти, розроблені для глобального ринку, не завжди враховують особливості законодавства або корпоративної культури в різних країнах. Наприклад, платформи, що впроваджуються в Європі, повинні відповідати вимогам GDPR [25] щодо захисту персональних даних, що не завжди враховується розробниками.
- **Проблеми з масштабованістю** Хоча більшість систем заявляють про можливість роботи з будь-якою кількістю працівників, їхня продуктивність може знижуватися при значному збільшенні обсягу даних [26]. Це особливо стосується платформ, які не підтримують децентралізоване зберігання даних або мають обмеження щодо обчислювальної потужності.
- **Відсутність персоналізованого підходу** Працівники різних посад і спеціалізацій потребують різних критеріїв оцінки. Наприклад, у відділах продажів ключовими показниками є кількість укладених угод, тоді як для відділу розробки ПЗ важливими є дотримання термінів і якість виконаних завдань. Стандартні системи часто не враховують цієї різноманітності.
- **Проблеми з впровадженням і навчанням [27]** Впровадження сучасних систем потребує значного навчання персоналу, що також вимагає додаткових витрат. Крім того, технічна підтримка цих систем не завжди забезпечує належний рівень допомоги на етапах адаптації.

Отже, існуючі системи, хоча й мають багато переваг, не завжди здатні задовольнити специфічні потреби сучасних підприємств, особливо малого та середнього бізнесу. Ці недоліки підкреслюють необхідність створення нових рішень, орієнтованих на індивідуальні потреби, високу доступність і ефективність [28].

1.3. Вимоги до системи оцінки якості та ефективності роботи працівників

Розробка комп'ютерної системи для оцінки якості та ефективності роботи працівників підприємства передбачає врахування різноманітних вимог, які визначають її функціонал і технічні характеристики. Ці вимоги поділяються на функціональні та нефункціональні [29]. Функціональні вимоги описують, які завдання має виконувати система, тобто її основний функціонал. Нефункціональні вимоги встановлюють параметри, обмеження та стандарти, яким повинна відповідати система для забезпечення ефективної роботи та відповідності сучасним умовам використання.

1.3.1. Функціональні вимоги системи оцінки

Функціональні вимоги до системи є ключовими, оскільки вони визначають, що саме система повинна робити, щоб виконувати своє основне призначення. У цьому контексті система повинна забезпечувати автоматизацію процесів оцінки продуктивності працівників, збирання та аналізу даних, а також формування результатів для подальшого використання керівництвом підприємства [30].

Перш за все, система повинна мати можливість автоматичного збору даних про продуктивність працівників. Це може включати інтеграцію з іншими внутрішніми системами підприємства, такими як CRM-системи, ERP-платформи чи програмне забезпечення для управління завданнями. Наприклад, система має отримувати дані про завершені завдання, відгуки клієнтів, дотримання дедлайнів і кількість виконаних проєктів [31]. Водночас важливо забезпечити можливість ручного введення інформації керівниками або самими працівниками для уточнення специфічних аспектів їхньої роботи.

Однією з основних функцій системи повинна бути оцінка ключових показників ефективності (KPI). Це означає, що система має дозволяти створювати, налаштовувати та моніторити персоналізовані KPI для кожного працівника або команди. Крім того, вона повинна автоматично розраховувати

значення цих показників на основі зібраних даних і представляти результати у зручному для аналізу вигляді. Для забезпечення наочності система має підтримувати візуалізацію даних у вигляді графіків, таблиць або інтерактивних інформаційних панелей [32].

Система повинна також надавати інструменти для створення аналітичних звітів. Це можуть бути регулярні звіти про продуктивність працівників, порівняння між різними підрозділами або аналіз тенденцій у змінах ефективності. Важливо, щоб ці звіти були детальними та містили рекомендації для підвищення продуктивності.

Ще однією важливою функцією є прогнозування. Використання алгоритмів машинного навчання дозволить системі передбачати можливі зміни в продуктивності працівників, виявляти потенційні ризики (наприклад, зниження мотивації або ризик звільнення) і допомагати керівництву ухвалювати обґрунтовані рішення на основі таких прогнозів.

Система повинна забезпечувати управління завданнями, включаючи можливість створення, розподілу та контролю їх виконання. Це дозволить керівникам не лише оцінювати ефективність працівників, а й краще координувати роботу команди [33].

Важливо, щоб система пропонувала інструменти для надання індивідуальних рекомендацій працівникам. Наприклад, вона може автоматично формувати поради щодо вдосконалення певних навичок або змін у робочих процесах, які допоможуть підвищити продуктивність.

Не менш важливою функцією є забезпечення прозорості. Система повинна давати можливість працівникам переглядати свої результати, розуміти, як саме вони оцінюються, і отримувати зворотний зв'язок від керівництва. Це сприятиме підвищенню довіри до системи та мотивуватиме співробітників до покращення своєї роботи [34].

Інтеграція з іншими системами підприємства є ще одним важливим аспектом. Наприклад, система повинна взаємодіяти з платформами управління

персоналом (HRM), фінансовими системами (для розрахунку премій), а також із календарями та системами управління проєктами.

1.3.2. Нефункціональні вимоги системи оцінки

Нефункціональні вимоги встановлюють стандарти, які гарантують ефективність, надійність і зручність використання системи. Однією з головних характеристик є продуктивність і швидкодія. Система повинна обробляти великі обсяги даних у режимі реального часу, забезпечуючи швидкий відгук на запити користувачів. Час очікування результатів не повинен перевищувати кількох секунд, навіть за умови одночасного використання системи великою кількістю користувачів.

Надійність також є критично важливою характеристикою. Система повинна гарантувати безперебійну роботу та мати механізми автоматичного резервного копіювання даних, щоб уникнути їх втрати в разі технічних збоїв.

Безпека та конфіденційність даних працівників мають бути пріоритетними. Система повинна відповідати сучасним стандартам захисту інформації, таким як GDPR або ISO 27001. Усі дані мають бути зашифровані під час збереження та передачі, а доступ до конфіденційної інформації повинен бути обмежений відповідно до ролей користувачів у системі [35].

Масштабованість системи є ще одним важливим параметром. Вона повинна підтримувати збільшення обсягів даних або кількості користувачів без значного зниження продуктивності. Це особливо актуально для великих підприємств або тих, що швидко зростають.

Зручність використання є ключовим аспектом, адже система повинна мати інтуїтивний інтерфейс, зрозумілий як для керівників, так і для працівників. Наявність локалізації кількома мовами та адаптації для користувачів із обмеженими можливостями також є важливими елементами.

Вартість володіння системою повинна бути оптимальною. Це стосується як початкових витрат на впровадження, так і витрат на обслуговування, навчання персоналу та оновлення програмного забезпечення.

Гнучкість системи означає її здатність адаптуватися до специфічних потреб підприємства. Наприклад, можливість налаштовувати методи оцінки, додавати нові функції чи інтегрувати систему з іншими платформами забезпечить її універсальність.

Сумісність із різними платформами є обов'язковою умовою. Система повинна працювати на десктопах, планшетах і смартфонах із популярними операційними системами, такими як Windows, macOS, Android і iOS.

Енергоефективність також має враховуватися, особливо якщо система працює в хмарному середовищі. Оптимізація використання обчислювальних ресурсів дозволить знизити експлуатаційні витрати.

Таким чином, функціональні та нефункціональні вимоги разом визначають рамки, у яких система буде працювати, забезпечуючи ефективність і відповідність потребам підприємства. Дотримання цих вимог є запорукою успішної розробки та впровадження такої системи.

1.4. Рівні цифрової грамотності

У рамці визначено лише загальні вимоги щодо рівнів оволодіння цифровою компетентністю, що демонструють широкий спектр і особливості фахових спеціалізацій працівників галузі. Це дозволяє деталізувати ці рівні, коли необхідно встановити професійні стандарти та кваліфікаційні характеристики працівників охорони здоров'я різних спеціалізацій. Таким чином, працівникам певних спеціалізацій можна отримати певні цифрові компетентності на достатньому рівні, тоді як працівникам інших спеціалізацій можна досягти більш високих професійних або високих рівнів, які мають бути визначені у кваліфікаційних вимогах (таблиця 2.1.).

Для розробки рівня володіння цифровою компетентністю, ви використовуєте якісні оцінки (базовий, достатній, фаховий, високий) і кілька оцінок, які використані в бальній системі, починаються від 1 до 100. Результати тестування та виконання практичних завдань розроблені для визначення балів.

Тип тестування, складність, кількість та обсяг завдань виконання кількісної оцінки.

Таблиця 2.1. Якісна шкала вимірювання рівнів кваліфікації щодо цифрової компетентності

Рівні володіння цифровими компетентностями		Складність завдань	Автономність роботи	Пізнавальний домен
Базовий	A	Чітко визначені прості і шаблонні завдання, прості проблеми	Самостійно або (за необхідності) за інструкціями або під керівництвом більш досвідченого експерта	Запам'ятовування
Достатній	B	Завдання та чітко визначені нешаблонні проблеми	Самостійно	Усвідомлення Розуміння
Фаховий	C	Завдання та проблеми різного ступеня складності	Самостійно і відповідно до своїх потреб	Застосування та оцінювання. Вміння обрати найкраще технологічне рішення
Високий	D	Складні завдання з обмеженим колом можливих рішень	Впевнено використовує самостійно та може навчати інших. Здійснює інтегрований внесок у професійну практику та керування іншими користувачами	Критичне оцінювання, інноваційне та творче впровадження цифрових технологій відповідно до посадових обов'язків

Тестування працівників охорони здоров'я за відповідними навчальними програмами, а також виконання ними практичних ситуативних завдань, що стосуються окремих елементів цифрової компетентності, дозволяє застосувати реальний рівень володіння певними компетентностями. такі модулі містять детальну інформацію про відповідні навички відповідно до їх дескрипторів. Працівники охорони здоров'я можуть використовувати тестові завдання для самооцінки своїх компетентностей і вибору навчальних курсів, які підвищують їхню кваліфікацію.

1.5. Характеристика підприємств як об'єктів автоматизації

Підприємства є складними системами, що складаються з великої кількості взаємопов'язаних компонентів: працівників, технологій, ресурсів і процесів. У

контексті автоматизації оцінки ефективності роботи працівників підприємства можна розглядати як об'єкти, які прагнуть досягти гармонійної взаємодії цих компонентів для максимізації своєї продуктивності та конкурентоспроможності. Автоматизація в управлінні ефективною сприяє підвищенню точності оцінок, зменшенню впливу людського фактору і пришвидшенню ухвалення управлінських рішень.

Підприємства різних масштабів — від малого бізнесу до великих корпорацій — мають спільну потребу в об'єктивній оцінці роботи працівників. Однак специфіка їхньої діяльності, кількість працівників і рівень технологічного розвитку суттєво впливають на вимоги до систем автоматизації. Малі підприємства зазвичай прагнуть до простих і доступних рішень, які забезпечують базову оцінку ефективності без складних аналітичних функцій. Натомість великі організації, які мають розгалужену структуру з численними підрозділами, потребують більш комплексних систем із можливістю інтеграції з іншими інформаційними платформами та детального аналізу даних.

Особливо важливим аспектом є галузева специфіка підприємств. Наприклад, виробничі підприємства можуть вимагати оцінки ефективності працівників на основі таких показників, як кількість виробленої продукції, дотримання стандартів якості та мінімізація відходів. У сфері послуг, навпаки, акцент може бути зроблено на задоволенні клієнтів, швидкості обслуговування та комунікативних навичках співробітників.

Автоматизація процесів оцінки є важливим кроком до підвищення прозорості управлінських процесів. Вона дозволяє керівникам зосередитися на стратегічних завданнях, делегуючи рутинні процеси збору й обробки даних інформаційним системам.

Користувачі системи (включаючи звичайних співробітників, керівників відділів і системних адміністраторів) повинні спочатку авторизуватися. Звичайні користувачі заповнюють, надсилають або змінюють форму самоперевірки. Керівники підрозділів розглядають форму саморекомендації. Потім користувачі

можуть запитувати дані відповідно до власних дозволів. Нарешті, їм потрібно вийти з усієї системи. Тому діаграма діяльності системи показана на рисунку 1.

На діаграмі активності системи початок і кінець усіх дозволів облікового запису реалізуються входом і виходом. Кожен обліковий запис має дозвіл на перегляд. Дозволи на перегляд тут різні. Системний адміністратор має дозвіл переглядати всю інформацію, тоді як керівник відділу має дозвіл переглядати інформацію про оцінку ефективності свого відділу, тоді як звичайні співробітники мають дозвіл переглядати лише інформацію про оцінку своєї роботи. Крім того, після входу ви можете виконувати різні операції шляхом модифікації, затвердження, подання тощо відповідно до різних облікових записів.

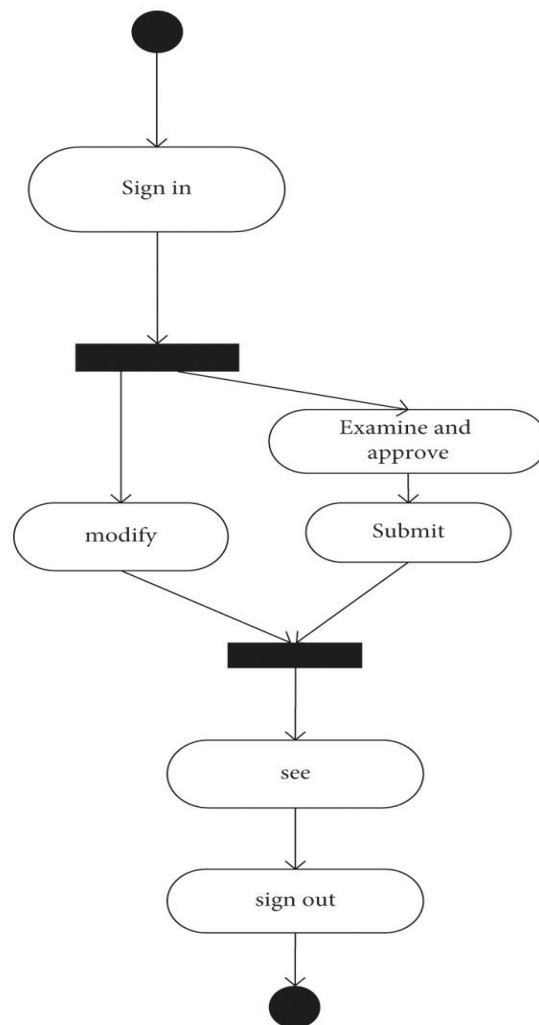


Рисунок 1. Діаграма діяльності системи

1.5.1. Особливості процесів управління ефективністю

Управління ефективністю працівників є складним процесом, який включає декілька взаємопов'язаних етапів. Першим етапом є визначення цілей, яких підприємство прагне досягти. Це можуть бути як стратегічні завдання (зростання ринку, покращення фінансових показників), так і операційні цілі, пов'язані з виконанням конкретних завдань. Відповідно, оцінка ефективності працівників завжди повинна узгоджуватися з цілями організації.

Наступним кроком є встановлення ключових показників, за якими буде здійснюватися оцінка. Процес управління ефективністю передбачає регулярний моніторинг цих показників, щоб керівники могли оперативної реагувати на зниження продуктивності або виявлення проблемних зон. Автоматизація цього процесу дозволяє мінімізувати витрати часу та ресурсів, забезпечуючи своєчасне отримання необхідної інформації.

Важливою особливістю є залучення працівників до процесу управління ефективністю. Успішна оцінка повинна бути прозорою та зрозумілою для всіх учасників, що забезпечує довіру до результатів і підвищує мотивацію працівників до самовдосконалення. Наприклад, якщо працівник бачить, що його роботу оцінюють за зрозумілими й об'єктивними критеріями, він більш схильний сприймати результати як справедливі.

Ще однією особливістю є необхідність врахування людського фактора. Хоча автоматизація дозволяє зменшити вплив суб'єктивності, вона не може повністю виключити його. Наприклад, навіть найточніші системи оцінки можуть потребувати втручання керівника для інтерпретації результатів або врахування обставин, які вплинули на продуктивність (наприклад, хвороба або сімейні обставини).

Процеси управління ефективністю часто включають формування планів розвитку для працівників. Це можуть бути рекомендації щодо підвищення кваліфікації, участі у тренінгах або зміни підходів до виконання завдань. Автоматизація цього аспекту дозволяє генерувати персоналізовані рекомендації, враховуючи результати оцінки.

1.5.2. Ключові показники оцінки роботи працівників

Оцінка ефективності роботи працівників базується на використанні ключових показників (KPI), які є кількісними та якісними характеристиками продуктивності. KPI дозволяють об'єктивно оцінити внесок кожного працівника у досягнення загальних цілей підприємства.

Продуктивність є одним із найважливіших показників, який вимірюється через кількість виконаних завдань або проєктів за певний період. Наприклад, для виробничого підприємства це може бути кількість вироблених одиниць продукції, а для працівників сфери послуг – кількість оброблених клієнтських запитів.

Якість роботи є іншим критично важливим показником. Вона може оцінюватися через кількість допущених помилок, відповідність виконаної роботи стандартам або задоволеність клієнтів. Наприклад, у сфері програмного забезпечення якість роботи може вимірюватися кількістю помилок у коді або часом, витраченим на їх виправлення.

Час виконання завдань також є важливим KPI. Він визначає, наскільки ефективно працівник використовує свій робочий час і чи дотримується він встановлених дедлайнів. У цьому контексті система може автоматично відстежувати витрати часу на кожне завдання.

Залученість працівника до роботи відображає його участь у проєктах, ініціативах або командній роботі. Це складний показник, який може враховувати кількість запропонованих ідей, участь у тренінгах або активність на внутрішніх корпоративних платформах.

Окремо варто виділити задоволеність клієнтів, яка може бути ключовим показником у сервісно-орієнтованих галузях. Вона вимірюється через оцінки, залишені клієнтами, кількість скарг або відгуків.

Комбінація кількісних і якісних показників дозволяє сформулювати комплексну оцінку ефективності працівників. Автоматизація цього процесу забезпечує не лише точність розрахунків, але й можливість аналізу змін у

показниках протягом часу, що допомагає виявляти тенденції та планувати подальші дії.

1.6. Постановка проблеми

Оцінка якості та ефективності роботи працівників є одним із ключових завдань сучасного підприємства. В умовах глобалізації, цифрової трансформації та зростання конкуренції успіх організації значною мірою залежить від здатності швидко адаптуватися до змін і максимально ефективно використовувати наявні ресурси. Зокрема, людський капітал, як основа будь-якої компанії, потребує постійного моніторингу та вдосконалення.

Однак процес оцінки ефективності працівників стикається з численними труднощами, пов'язаними як з методологічними аспектами, так і з технічними й організаційними викликами. Традиційні методи оцінки, такі як періодичні огляди роботи керівником або опитування, часто є застарілими, суб'єктивними та нездатними забезпечити повну картину про продуктивність працівників. Це створює значні ризики для підприємства, включаючи зниження мотивації співробітників, збільшення плинності кадрів і втрату конкурентних переваг.

Крім того, сучасні підприємства стикаються з необхідністю обробки великої кількості різноманітної інформації: від даних про виконання завдань до показників задоволеності клієнтів. Ручна обробка цих даних не лише займає багато часу, але й підвищує ризик помилок. Автоматизація процесу оцінки є перспективним вирішенням цієї проблеми, проте розробка відповідної системи вимагає врахування численних викликів і обмежень, з якими стикаються підприємства.

1.6.1. Основні виклики

Одним із ключових викликів у розробці комп'ютерної системи для оцінки якості та ефективності працівників є інтеграція різноманітних джерел даних. Сучасне підприємство використовує багато різних платформ для управління завданнями, збору інформації про продуктивність, взаємодії з клієнтами тощо.

Інтеграція цих платформ в одну систему потребує розробки гнучких і масштабованих рішень, які дозволять збирати, обробляти й аналізувати дані в реальному часі.

Ще одним важливим викликом є забезпечення прозорості оцінок. Працівники часто недовіряють автоматизованим системам через відсутність розуміння, за якими критеріями оцінюється їхня робота. Це може призводити до конфліктів і зниження мотивації. Тому система повинна бути не лише точною, але й зрозумілою для всіх учасників.

Використання інноваційних технологій, таких як штучний інтелект і машинне навчання, також ставить перед розробниками нові завдання. Хоча ці технології дозволяють автоматизувати складні процеси та забезпечити прогнозування продуктивності, їх впровадження потребує високої кваліфікації розробників, значних ресурсів і часу.

Значний виклик становить і забезпечення конфіденційності даних. У процесі оцінки система збирає та обробляє багато конфіденційної інформації, зокрема персональні дані працівників, інформацію про їхню продуктивність і навіть зворотний зв'язок від клієнтів. Відповідність сучасним стандартам захисту даних, таким як GDPR, є обов'язковою умовою для розробки та впровадження такої системи.

Не менш важливим викликом є адаптивність системи до потреб різних підприємств. Кожна організація має свої унікальні особливості, які потребують врахування під час проектування системи. Це стосується як специфіки галузі, так і внутрішніх бізнес-процесів. Створення універсальної платформи, яка одночасно враховувала б специфіку малого бізнесу та великих корпорацій, є складним, але необхідним завданням.

1.6.2. Обмеження

Одним із найбільш вагомих обмежень у розробці системи є фінансові ресурси. Впровадження автоматизованої системи оцінки потребує значних початкових витрат, включаючи розробку програмного забезпечення, закупівлю

обладнання, інтеграцію з існуючими платформами та навчання персоналу. Для малих і середніх підприємств ці витрати можуть бути надто високими, що обмежує їх доступ до сучасних технологій.

Технічні обмеження також відіграють важливу роль. Система повинна забезпечувати високу продуктивність навіть за умови обробки великої кількості даних у режимі реального часу. Недостатня потужність серверів або відсутність хмарної інфраструктури можуть значно впливати на швидкість і якість роботи системи.

Організаційні обмеження включають низький рівень готовності персоналу до змін. Впровадження нової системи може зустріти опір з боку працівників, які звикли до традиційних методів оцінки. Крім того, недостатня кваліфікація співробітників у роботі з новими технологіями може уповільнити процес інтеграції та зменшити ефективність використання системи.

Юридичні аспекти також є суттєвим обмеженням. Система повинна відповідати вимогам законодавства щодо захисту персональних даних у різних країнах. Це може ускладнювати її впровадження на міжнародному рівні, особливо для компаній, які працюють у кількох юрисдикціях.

Культурні та соціальні фактори можуть впливати на ефективність впровадження системи. У різних регіонах існують різні підходи до управління працівниками та оцінки їхньої продуктивності. Наприклад, у країнах із жорсткою ієрархією співробітники можуть сприймати автоматизовані системи як спосіб посилення контролю, тоді як у більш гнучких організаціях вони можуть бути інструментом розвитку.

Таким чином, розробка та впровадження системи для оцінки якості й ефективності працівників є складним завданням, яке потребує врахування численних викликів і обмежень. Успішне вирішення цих питань можливе лише за умови комплексного підходу, що поєднує технологічні інновації, глибокий аналіз потреб підприємств і залучення працівників до процесу оцінки.

РОЗДІЛ 2.

ТЕОРЕТИЧНІ ОСНОВИ ПОСТАВЛЕНОЇ ЗАДАЧІ

Розробка комп'ютерної системи для оцінки якості та ефективності роботи працівників базується на низці теоретичних концепцій, які охоплюють різні аспекти автоматизації управлінських процесів, аналізу даних і підвищення продуктивності. Основою для вирішення поставленої задачі є поєднання сучасних підходів до управління людськими ресурсами, математичного моделювання, машинного навчання та інформаційних технологій.

Одним із базових елементів є поняття ключових показників ефективності (KPI), які використовуються для кількісного оцінювання продуктивності працівників. KPI є метриками, що відображають досягнення конкретних цілей. Ці показники можуть включати, наприклад, кількість виконаних завдань, швидкість обробки запитів клієнтів, якість виконаної роботи чи рівень задоволеності клієнтів. KPI базуються на принципах SMART: вони повинні бути конкретними (Specific), вимірюваними (Measurable), досяжними (Achievable), релевантними (Relevant) і обмеженими в часі (Time-bound).

Ще однією теоретичною основою є управління ефективністю працівників, що охоплює процеси планування, моніторингу та оцінювання результатів роботи. Цей підхід тісно пов'язаний із теорією мотивації, зокрема з моделями Маслоу, Герцберга та Врума. Теорія Маслоу вказує на важливість задоволення базових і вищих потреб працівників, тоді як двофакторна теорія Герцберга підкреслює роль гігієнічних і мотиваційних факторів у підвищенні продуктивності.

Для аналізу даних, які збирає система, широко використовуються статистичні методи та алгоритми машинного навчання. До прикладу, кластерний аналіз дозволяє групувати працівників за рівнем їхньої продуктивності, а регресійний аналіз використовується для прогнозування майбутніх результатів на основі поточних даних. Методи класифікації, такі як дерева рішень або випадкові ліси, дозволяють автоматично визначати працівників із високою чи низькою ефективністю.

Інтеграція теоретичних підходів до аналізу даних із практичними методами управління персоналом стала можливою завдяки розвитку інформаційних технологій. Використання сучасних платформ для обробки великих даних, таких як Apache Hadoop або Spark, дозволяє ефективно працювати з великими обсягами інформації, які генеруються в рамках діяльності підприємства. У поєднанні з бібліотеками для машинного навчання, такими як TensorFlow або Scikit-learn, це відкриває нові можливості для автоматизації процесів управління.

Автоматизація процесу оцінки працівників вимагає розробки системи, яка враховує архітектурні принципи проектування програмного забезпечення. Архітектура системи повинна бути гнучкою, модульною і масштабованою. Найчастіше для побудови таких систем використовуються мікросервісні архітектури, які дозволяють розділити функціонал на незалежні компоненти. Це спрощує обслуговування та інтеграцію з іншими системами, такими як CRM або ERP.

Особливе значення має візуалізація даних. Для ефективного аналізу продуктивності необхідно надавати керівникам підприємства зрозумілі й інформативні графіки, які відображають динаміку змін, розподіл показників і порівняння між працівниками. Візуалізація може реалізовуватися за допомогою спеціалізованих інструментів, таких як Power BI, Tableau або бібліотеки для розробки графіків у Python (Matplotlib, Seaborn).

1.4. Моделі оцінки якості роботи працівників

Оцінка якості роботи працівників є складним завданням, яке потребує системного підходу, що враховує не лише результати діяльності, але й поведінкові аспекти, професійні компетенції та внесок у досягнення стратегічних цілей підприємства. Моделі оцінки якості роботи будуються на основі різних методологій, включаючи ключові показники ефективності, багатокритеріальний аналіз та підходи, які враховують специфіку діяльності конкретного підприємства (рис. 2).

Основою оцінки якості роботи є формування системи метрик, які дозволяють кількісно і якісно оцінити результати праці. Це можуть бути показники продуктивності, точності виконання завдань, відповідності роботи стандартам, а також фактори, що відображають особисті якості працівника, такі як комунікабельність, здатність до навчання чи стресостійкість.

Для забезпечення об'єктивності та зменшення впливу суб'єктивних оцінок використовуються стандартизовані методології, які враховують основні фактори, що впливають на результативність працівника. Зокрема, моделі можуть бути засновані на поведінкових індикаторах, таких як відповідність корпоративним цінностям, або на основі результатів, які працівник досяг за певний період.

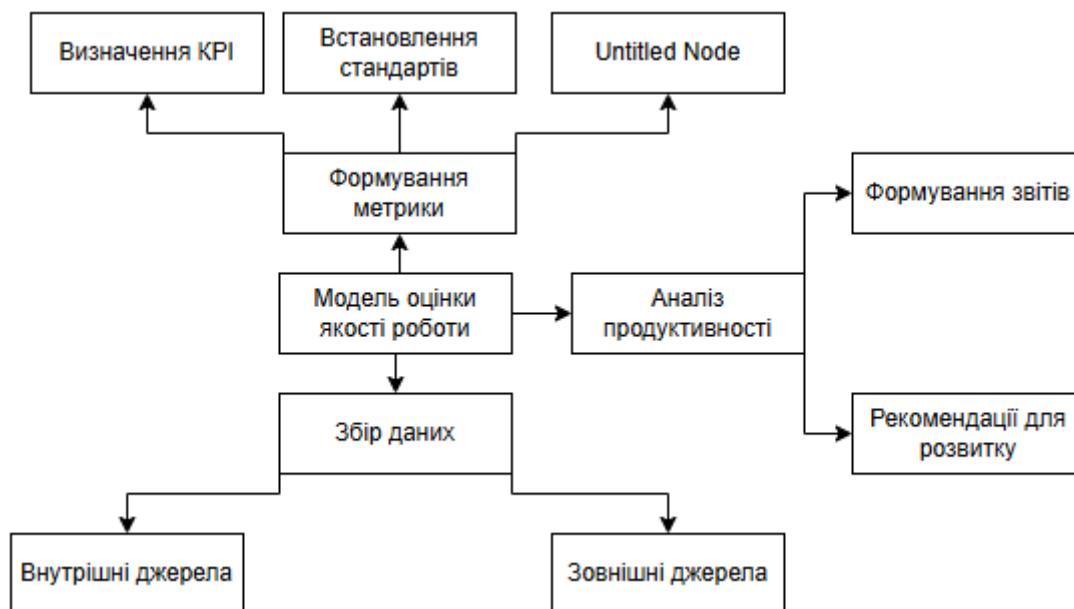


Рисунок 2. Структура моделі оцінки

1.5. Методології оцінювання ключових показників ефективності

Ключові показники ефективності (КПЕ) є однією з найпоширеніших методологій оцінювання роботи працівників. КПЕ забезпечують кількісну оцінку результатів роботи, допомагаючи керівникам приймати обґрунтовані рішення щодо розвитку персоналу, преміювання або корекції бізнес-процесів.

Методологія оцінювання КПЕ базується на принципах SMART, які гарантують, що показники будуть чіткими, вимірюваними, досяжними,

релевантними та обмеженими в часі. Наприклад, для працівника відділу продажів КРІ може включати кількість укладених угод за місяць, середній чек клієнта або відсоток успішно завершених переговорів.

Процес впровадження КРІ (рис. 3) починається з аналізу цілей підприємства, після чого визначаються ключові результати, досягнення яких є критичним для успіху організації. Далі ці результати трансформуються у метрики, які відображають внесок кожного працівника в досягнення цілей. Важливо, щоб КРІ були гнучкими та адаптувалися до змін у бізнес-середовищі.

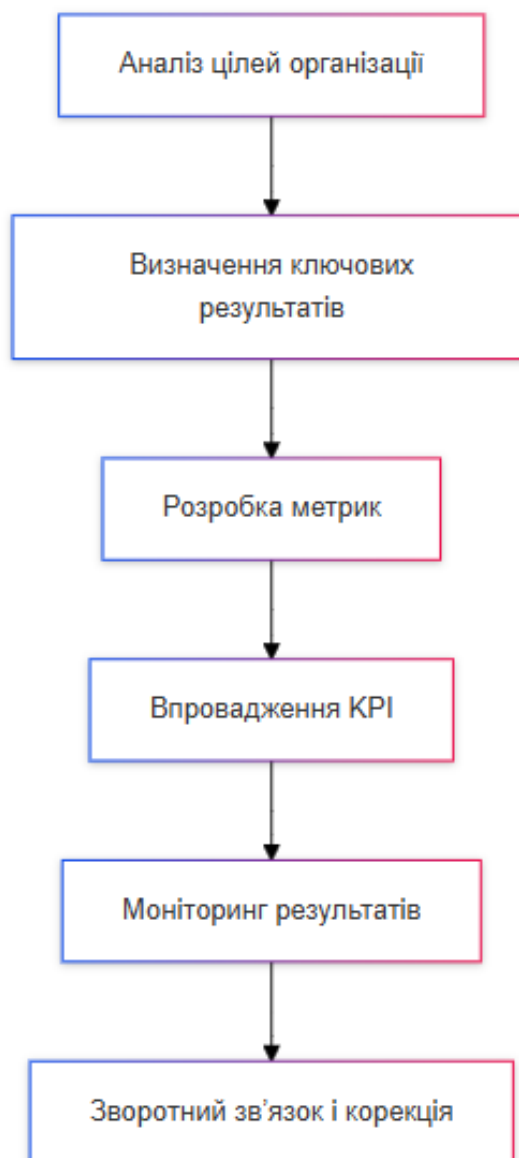


Рисунок 3. Процес впровадження ключових показників ефективності

Для аналізу КРІ можуть використовуватися різні інструменти, такі як дашборди, які відображають ключові показники в режимі реального часу, або автоматизовані системи для збору й аналізу даних.

1.6. Моделі багатокритеріального аналізу

Багатокритеріальний аналіз є підходом, який дозволяє враховувати кілька різних критеріїв при оцінці якості роботи працівників. Цей метод є особливо корисним у ситуаціях, коли один показник не може повністю відобразити внесок працівника в успіх організації.

Моделі багатокритеріального аналізу часто використовують для поєднання кількісних і якісних показників. Наприклад, для оцінки роботи співробітника можуть враховуватися такі критерії, як продуктивність, якість виконаних завдань, відповідність дедлайнам, рівень командної взаємодії та інноваційність.

Одним із поширених інструментів для багатокритеріального аналізу є метод аналізу ієрархій (АНР). Цей метод дозволяє розділити загальну задачу на кілька підзадач, кожна з яких має свої критерії. Вага кожного критерію визначається експертами, після чого для кожного працівника обчислюється загальний бал на основі його результатів.

Ще одним популярним методом є TOPSIS (Technique for Order Preference by Similarity to Ideal Solution). Цей підхід порівнює оцінювані варіанти з ідеальним і найгіршим варіантом, дозволяючи визначити, який із варіантів є найкращим.

Для візуалізації багатокритеріального аналізу часто використовуються діаграми Парето, графіки важливості критеріїв або радари ефективності.

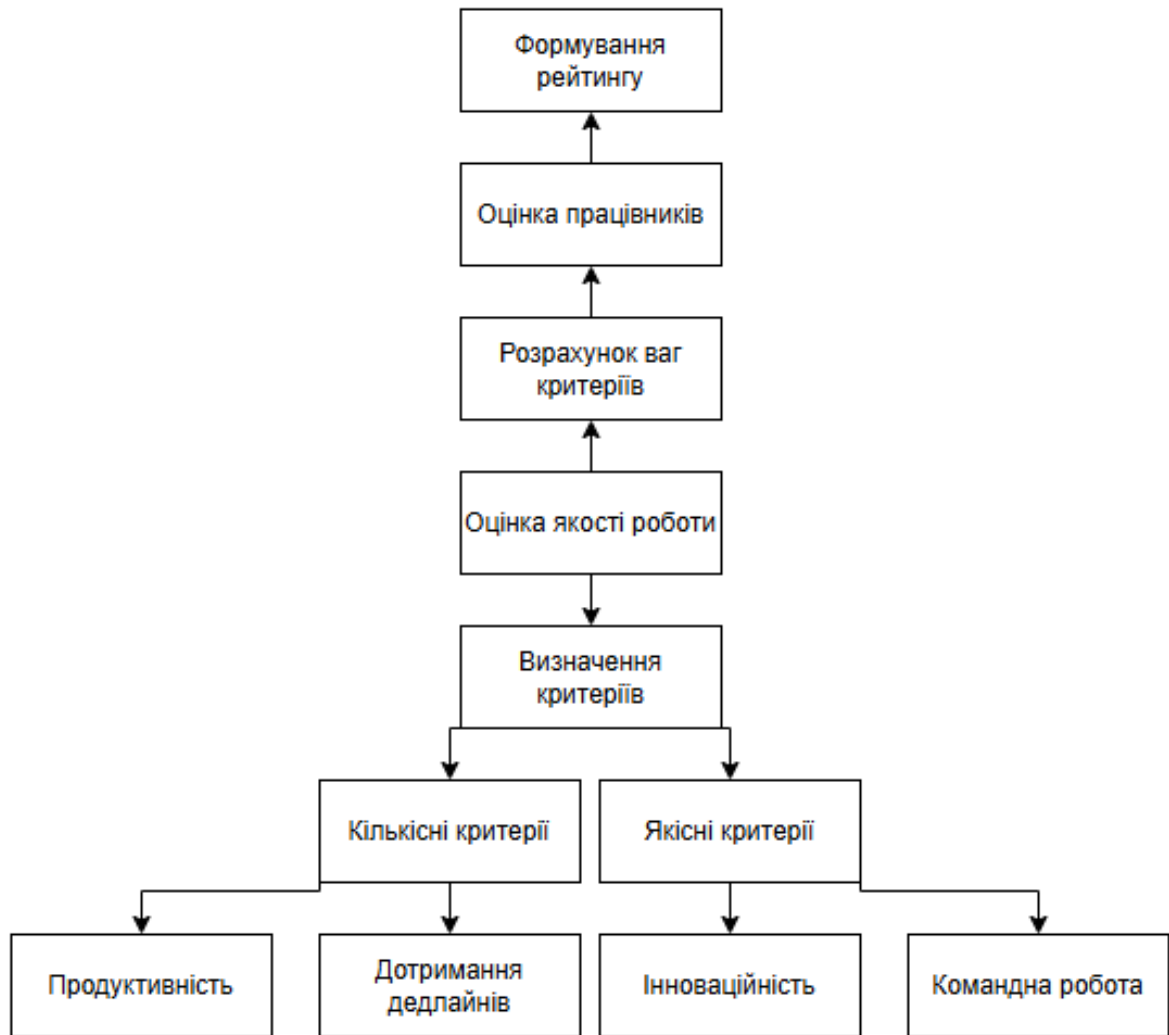


Рисунок 4. Модель багатокритеріального аналізу

Таким чином, поєднання моделей оцінки, методологій KPI та багатокритеріального аналізу дозволяє створити комплексну систему оцінки роботи працівників, яка враховує як кількісні, так і якісні аспекти їхньої діяльності. Це забезпечує об'єктивність, прозорість і можливість адаптації до специфічних умов конкретного підприємства.

1.7. Методи збору, обробки та аналізу даних

Збір, обробка та аналіз даних є критично важливими етапами у створенні системи оцінки якості та ефективності роботи працівників, оскільки на основі зібраних даних приймаються управлінські рішення, формуються звіти та рекомендації для підвищення продуктивності. Враховуючи великі обсяги даних, які генеруються в процесі діяльності підприємства, застосування передових

методів збору та аналізу даних є необхідним для отримання точної та своєчасної інформації.



Рисунок 5. Діаграма для збору та обробки даних у комп'ютерній системі

Ця діаграма відображає процес збору даних з різних джерел (CRM, ERP, соціальні мережі), їх очищення та нормалізацію, а також передачу в аналітичну платформу для аналізу та прогнозування продуктивності. Всі ці етапи є важливими для створення точних і надійних оцінок ефективності працівників.

Обробка великих даних (Big Data)

Одним із основних викликів для сучасних підприємств є ефективне управління великими обсягами інформації, які генеруються в процесі взаємодії з працівниками, клієнтами та іншими зацікавленими сторонами. Обробка великих даних (Big Data) включає в себе збирання, зберігання, обробку та аналіз величезних масивів структурованих і неструктурованих даних, які постійно оновлюються, зображено на рисунку 6.

Big Data дозволяє підприємствам зібрати і аналізувати дані з різних джерел, таких як CRM-системи, ERP-платформи, соціальні мережі, електронна пошта та інші канали взаємодії з працівниками та клієнтами. Ці дані можуть включати інформацію про виконані завдання, рівень задоволеності клієнтів, час, витрачений на виконання завдань, тощо.

Для ефективної обробки великих даних використовуються спеціалізовані платформи, такі як Hadoop або Apache Spark, які дозволяють масштабувати обробку даних та забезпечити високу швидкість обробки. Вони підтримують паралельну обробку, що дозволяє працювати з великими обсягами даних без значних затримок. Це дозволяє швидко отримувати актуальну інформацію для прийняття рішень, наприклад, у реальному часі.

Для аналізу великих даних застосовуються різноманітні інструменти та методи, зокрема статистичний аналіз, побудова моделей машинного навчання та алгоритми глибинного навчання. Важливими аспектами є вибір правильних інструментів для збору даних, їх обробки та зберігання, а також забезпечення безпеки та конфіденційності даних.

Методи машинного навчання для аналізу продуктивності

Машинне навчання є потужним інструментом для аналізу продуктивності працівників, оскільки дозволяє не лише обробляти великий обсяг даних, але й передбачати майбутні результати. Завдяки алгоритмам машинного навчання система може автоматично виявляти патерни в даних, що дозволяє покращити оцінку ефективності працівників.

Одним з найбільш поширених методів машинного навчання є класифікація, яка дозволяє поділити працівників на різні категорії за рівнем продуктивності (наприклад, високий, середній, низький). Для цього використовуються алгоритми, такі як дерева рішень, логістична регресія, випадкові ліси та нейронні мережі. Ці алгоритми здатні обробляти різноманітні фактори, які впливають на результативність працівника, і на основі цих факторів формувати прогнози.

Іншим важливим методом є регресійний аналіз, який дозволяє оцінити залежність між різними змінними, наприклад, між рівнем ефективності працівника і факторами, такими як час роботи, кількість завдань або рівень мотивації. Методи регресії, такі як лінійна регресія або регресія на основі дерев рішень, можуть бути використані для прогнозування майбутньої продуктивності на основі попередніх даних.

Кластеризація є ще одним методом машинного навчання, який допомагає групувати працівників за схожими характеристиками продуктивності. Алгоритми, такі як K-means або Hierarchical clustering, дозволяють автоматично визначати групи працівників, які мають схожі результати, що дає змогу більш ефективно управляти командами або підрозділами.

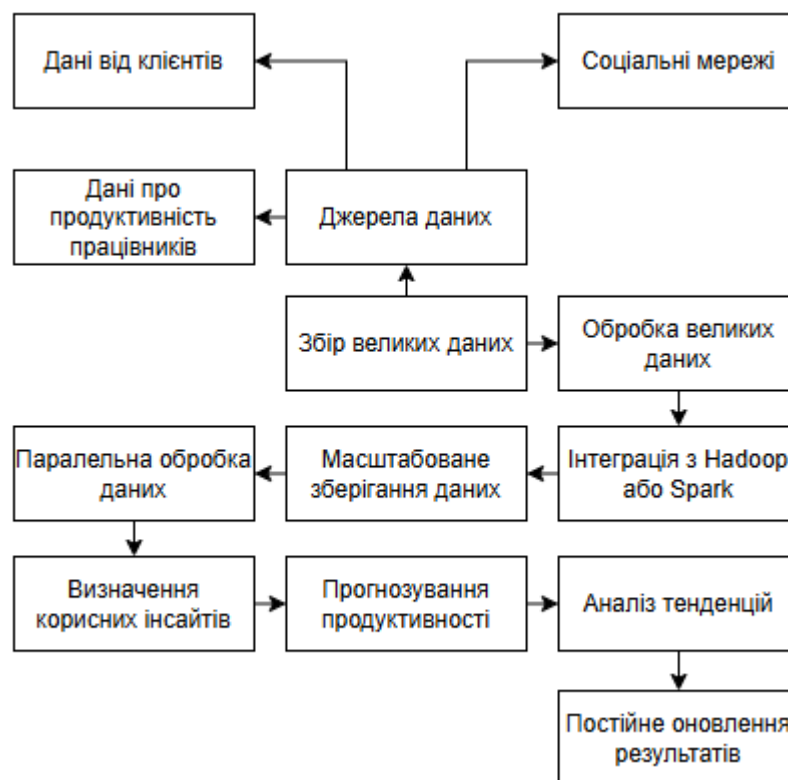


Рисунок 6. Діаграма для обробки великих даних

Ця діаграма ілюструє використання платформ для обробки великих даних (Hadoop або Spark), які дозволяють здійснювати масштабовану обробку даних, зберігати їх у великих масивах та здійснювати паралельну обробку. Це дозволяє ефективно працювати з великими обсягами даних у реальному часі, що є критично важливим для системи оцінки продуктивності працівників.

1.8. Методи збору обробки та аналізу великих даних

Збір, обробка та аналіз даних є критично важливими етапами у створенні системи оцінки якості та ефективності роботи працівників, оскільки на основі зібраних даних приймаються управлінські рішення, формуються звіти та рекомендації для підвищення продуктивності. Враховуючи великі обсяги даних, які генеруються в процесі діяльності підприємства, застосування передових методів збору та аналізу даних є необхідним для отримання точної та своєчасної інформації.

Одним із основних викликів для сучасних підприємств є ефективне управління великими обсягами інформації, які генеруються в процесі взаємодії з працівниками, клієнтами та іншими зацікавленими сторонами. Обробка великих даних (Big Data) включає в себе збирання, зберігання, обробку та аналіз величезних масивів структурованих і неструктурованих даних, які постійно оновлюються.

Big Data дозволяє підприємствам зібрати і аналізувати дані з різних джерел, таких як CRM-системи, ERP-платформи, соціальні мережі, електронна пошта та інші канали взаємодії з працівниками та клієнтами. Ці дані можуть включати інформацію про виконані завдання, рівень задоволеності клієнтів, час, витрачений на виконання завдань, тощо.

Для ефективної обробки великих даних використовуються спеціалізовані платформи, такі як Hadoop або Apache Spark, які дозволяють масштабувати обробку даних та забезпечити високу швидкість обробки. Вони підтримують паралельну обробку, що дозволяє працювати з великими обсягами даних без значних затримок. Це дозволяє швидко отримувати актуальну інформацію для прийняття рішень, наприклад, у реальному часі.

Для аналізу великих даних застосовуються різноманітні інструменти та методи, зокрема статистичний аналіз, побудова моделей машинного навчання та алгоритми глибинного навчання. Важливими аспектами є вибір правильних інструментів для збору даних, їх обробки та зберігання, а також забезпечення безпеки та конфіденційності даних.

Методи машинного навчання для аналізу продуктивності

Машинне навчання є потужним інструментом для аналізу продуктивності працівників, оскільки дозволяє не лише обробляти великий обсяг даних, але й передбачати майбутні результати. Завдяки алгоритмам машинного навчання система може автоматично виявляти патерни в даних, що дозволяє покращити оцінку ефективності працівників (рис. 7).

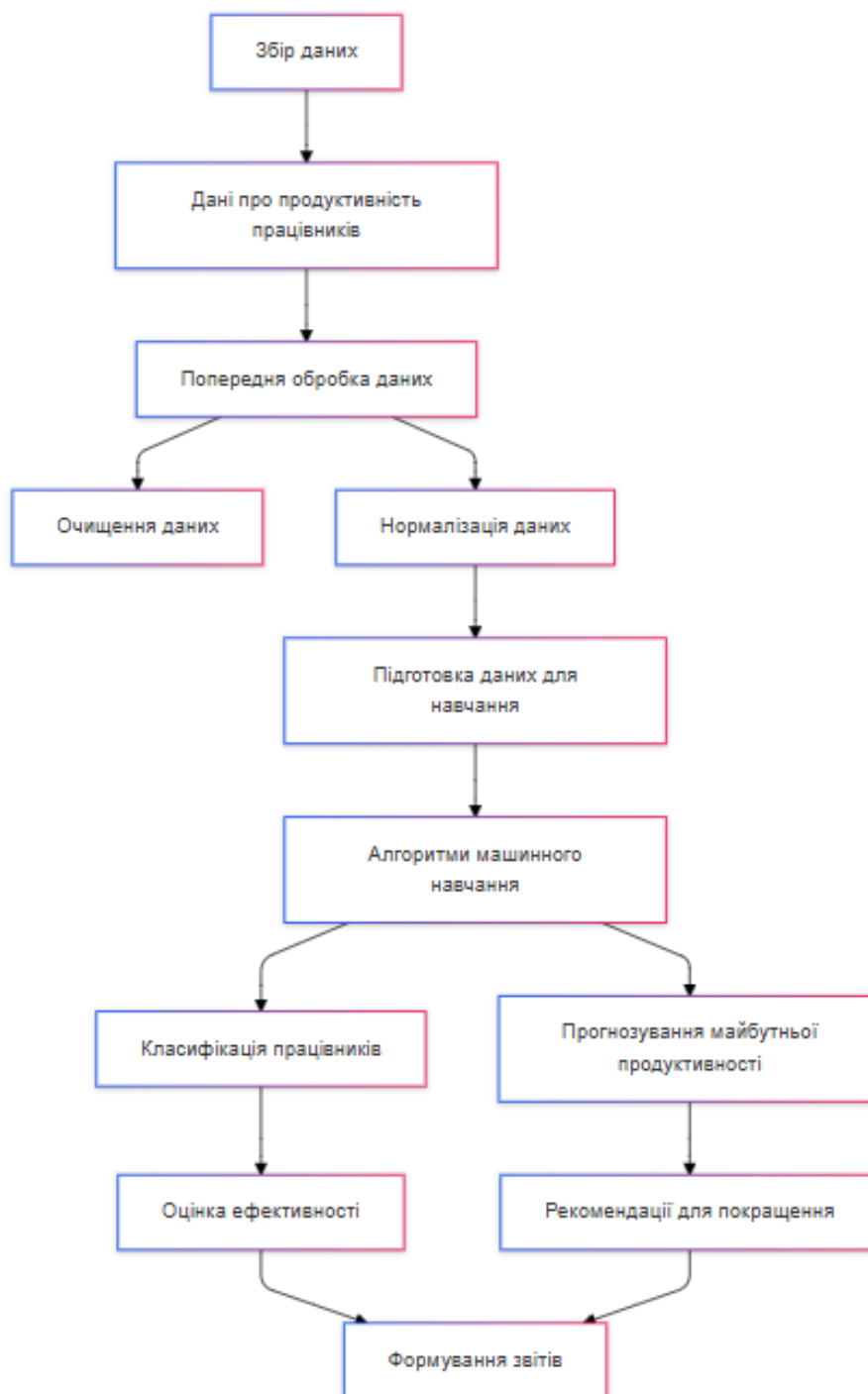


Рисунок 7. Діаграма для аналізу та обробки даних з використанням алгоритмів машинного навчання

Одним з найбільш поширених методів машинного навчання є класифікація, яка дозволяє поділити працівників на різні категорії за рівнем продуктивності (наприклад, високий, середній, низький). Для цього використовуються алгоритми, такі як дерева рішень, логістична регресія, випадкові ліси та нейронні мережі. Ці алгоритми здатні обробляти різноманітні фактори, які впливають на результативність працівника, і на основі цих факторів формувати прогнози.

Іншим важливим методом є регресійний аналіз, який дозволяє оцінити залежність між різними змінними, наприклад, між рівнем ефективності працівника і факторами, такими як час роботи, кількість завдань або рівень мотивації. Методи регресії, такі як лінійна регресія або регресія на основі дерев рішень, можуть бути використані для прогнозування майбутньої продуктивності на основі попередніх даних.

Кластеризація є ще одним методом машинного навчання, який допомагає групувати працівників за схожими характеристиками продуктивності. Алгоритми, такі як K-means або Hierarchical clustering, дозволяють автоматично визначати групи працівників, які мають схожі результати, що дає змогу більш ефективно управляти командами або підрозділами.

Діаграма, зображена на рисунку 7, описує процес застосування алгоритмів машинного навчання для аналізу продуктивності працівників. Зібрані дані проходять попередню обробку, після чого використовуються для навчання моделей машинного навчання, які класифікують працівників за рівнем продуктивності і прогнозують їх майбутні результати. На основі цього генеруються звіти та рекомендації для покращення.

1.9. Стандарти управління якістю та ефективністю

Управління якістю та ефективністю є невід'ємною частиною сучасного бізнесу. Визначення стандартів управління є необхідним для забезпечення високого рівня продуктивності працівників та досягнення стратегічних цілей організації.

Міжнародні стандарти ISO (Міжнародна організація по стандартизації) є одним із основних інструментів управління якістю на підприємствах. Стандарт ISO 9001, зокрема, є стандартом для систем управління якістю, який визначає вимоги до процесів, що забезпечують постійне вдосконалення якості послуг і продукції. Впровадження стандартів ISO дозволяє підприємствам забезпечити високий рівень відповідності своїх продуктів або послуг міжнародним вимогам, що є особливо важливим для конкурентоспроможності на глобальному ринку.

ISO 9001 зосереджений на таких аспектах як задоволеність клієнтів, ефективність процесів і постійне вдосконалення. Для підприємств, що використовують стандарти ISO, ефективність працівників вимірюється через здатність виконувати завдання відповідно до встановлених вимог і стандартів, що забезпечує стабільну та високу якість роботи.

Lean і Six Sigma є двома популярними методологіями, що фокусуються на поліпшенні ефективності підприємств шляхом мінімізації витрат і підвищення якості. Обидві методології з'явилися у виробничих середовищах, але згодом були адаптовані для широкого спектра бізнес-процесів.

Lean — це методологія, яка зосереджена на усуненні всіх видів втрат, що не додають цінності продукту чи послугі. У контексті оцінки працівників, Lean фокусується на зменшенні неефективних процесів, скороченні часу, витраченого на виконання завдань, та підвищенні цінності кожного етапу робочого процесу.

Six Sigma — це методологія, яка націлена на зниження варіацій в процесах, що веде до підвищення якості результатів і зменшення помилок. Використання Six Sigma в управлінні ефективністю працівників допомагає зосередитися на точності виконання завдань і забезпечує постійне вдосконалення результатів, зокрема за допомогою методів статистичного контролю якості (SQC).

Обидві методології використовують принципи, які дозволяють оптимізувати процеси, виявляти причини неефективності та знижувати витрати часу і ресурсів. Вони можуть бути застосовані для оцінки не тільки якості, але й продуктивності працівників на різних етапах їх роботи, зображено на рисунку 8.



Рисунок 8. Модель управління якістю та ефективністю

У результаті застосування стандартів ISO, Lean та Six Sigma підприємства можуть значно підвищити ефективність роботи своїх працівників, знизити витрати та покращити загальну якість на всіх етапах виробничих або бізнес-процесів.

1.10. Інтеграція автоматизованих систем у бізнес-процеси

Інтеграція комп'ютерної системи оцінки працівників із бізнес-процесами підприємства є ключовим фактором її ефективного впровадження (рис. 9). Сучасні підприємства використовують різні інформаційні системи, такі як CRM (Customer Relationship Management), ERP (Enterprise Resource Planning), системи управління завданнями (Task Management Systems) тощо. Кожна з цих систем містить цінні дані, які можуть бути використані для оцінки роботи працівників.

Інтеграція передбачає створення єдиного інформаційного середовища, в якому дані з різних джерел зводяться до єдиної бази. Наприклад, інформація з CRM-систем може бути використана для оцінки працівників відділу продажів, ERP — для аналізу ефективності виробничих процесів, а системи управління

завданнями — для оцінки продуктивності команд розробників або проектних менеджерів.

Процес інтеграції вимагає використання API (Application Programming Interface) для взаємодії між системами. Також важливо враховувати питання захисту даних і забезпечення відповідності міжнародним стандартам, таким як GDPR, що стосуються конфіденційності персональної інформації працівників.



Рисунок 9. Діаграма інтеграції автоматизованих систем

1.11. Економічний і соціальний вплив автоматизації оцінки роботи працівників

Автоматизація процесу оцінки працівників не лише підвищує ефективність управління, але й має значний економічний і соціальний вплив на підприємство. Економічний вплив виражається у скороченні витрат, пов'язаних із ручною обробкою даних, зменшенні часу, необхідного для підготовки звітів, і підвищенні загальної продуктивності працівників. Крім того, автоматизація дозволяє точніше визначати пріоритети у витратах на навчання персоналу, що знижує витрати на неефективні тренінги та інші заходи з розвитку.

Соціальний вплив автоматизації пов'язаний із покращенням умов праці для співробітників. Об'єктивні дані, зібрані автоматизованою системою, сприяють прозорості процесу оцінювання, що зменшує ймовірність конфліктів між

працівниками та керівництвом. Співробітники отримують зрозумілі результати оцінювання та рекомендації для особистісного й професійного зростання. Водночас, автоматизація може створювати ризики, такі як зростання технічної залежності або опір працівників до впровадження нових технологій. Тому важливо забезпечити належну підготовку персоналу до роботи з новою системою та інтегрувати її у корпоративну культуру підприємства, які зображені на рисунку 10.



Рисунок 10. Діаграма економічного та соціального впливу

1.12. Кількісний аналіз характеристик стандартів оцінки ефективності

У поєднанні з систематичним моделюванням ознак та ієрархічним аналізом стандартів оцінки ефективності виконується кількісний аналіз характеристик стандартів оцінки ефективності. Параметри обмежень стандартів оцінки діяльності аналізуються за допомогою методу аналізу індексу нечітких обмежень. Результати ієрархічного аналізу стандартів оцінки ефективності такі:

$$\text{MSD}_{a \rightarrow b} = 1 - \frac{\sum_{i=1}^{I_{a,b}} \sqrt{d_{a,i}^2 + d_{b,i}^2}}{I_{a,b} \times \sum_{i=1}^{I_{a,b}} \left[\sqrt{(d_{a,i} - \bar{d}_a)^2} + \sqrt{(d_{b,i} - \bar{d}_b)^2} \right]} \quad (2.1.)$$

де $\text{MSD}_{a \rightarrow b}$ — функція узгодженості розподіленої оцінки оцінювання ефективності за багаторівневою схемою оцінювання, а $d_{a,i}$ — екзаменаційний бал; \bar{d}_a вказує вид і кількість методів оцінки; $d_{b,i}$ вказує на раціональність оцінки роботи (значення від розумного до необґрунтованого поділяються на 1, 2 і 3); \bar{d}_b являє собою суму інвестованого підприємством капіталу; $I_{a,b}$ вказує на ступінь розміщення підприємства. Рівень розподілу факторів $X(0)$ показників оцінки діяльності поділяється на 8 оцінок, що становить $X(1), X(2), \dots, X(N)$, тобто $x^{(0)} = U_{i=1}^N x$. З використанням моделі лінійного програмування зв'язок кореляційного відображення показників атестації має такий вигляд:

$$P(R^N = r,) = p\left(\frac{X^N = x, |x_i| = |r_i|, \text{angle}(x_i)}{(\text{angle}(r_i) = \varphi_g), \text{mod}(2\pi)}\right) \quad (2.2.)$$

Метод декомпозиції емпіричного режиму використовується для декомпозиції характеристик системи оцінки ефективності, а функція нечіткого набору будується як . Отримано, що характерна величина аналізу індексу параметра обмеження відповідає $I(R^N; \varphi_g | Z^N) = V_2$, яке підставляється у наведену вище формулу:

$$R_2^T = \frac{(R^N; \varphi_g | Z^N) \sum_2 V_2^T}{R_2} \quad (2.3.)$$

Встановить параметр статусу набір факторів впливу на атестацію як

$$F = \{f_1, f_2, \dots, f_n\} \quad (2.4)$$

При кількісному оцінюванні факторів системи оцінки отримано, що набір планування нечіткої множини розподілу факторів оцінки в режимі лінійного програмування

$$f(t) = \frac{1}{R_2^T F} \frac{d}{dt} [\text{arg}z(t)] \quad (2.5.)$$

Характерною величиною будівельного рішення є

$$S_x = Ef(t) + \sqrt{sbu}[s(t - T_0)] \quad (2.6.)$$

$$K_x = Ef(t) - 3E^2 f(t)b, \quad (2.7.)$$

де $E[x^3(t)]$ є очікуваним значенням, b є коефіцієнтом прийняття рішень, а набір нечіткого планування для планування індексу факторів оцінки є

$$\begin{cases} H_0: x'(t) = w(t), \\ H_1: \sqrt{Es}(t) + w(t), \\ 0 \leq t \leq T. \end{cases} \quad (2.8.)$$

У наведеній вище формулі $x'(t)$ і $s'(t)$ є відповідно

$$\begin{aligned} x'(t) &= x(t) * h_w(t) \\ s'(t) &= s(t) * h_w(t) \end{aligned} \quad (2.9.)$$

Підводячи підсумок, витягуються характеристики правил асоціації та будується архітектура індексу параметра обмеження.

Побудовано багатопорядкову модель аналізу власних мод факторів оцінки.

2.10. Дизайн бази даних

В даний час існує багато продуктів баз даних. Продукти кожної бази даних орієнтовані на різні групи користувачів. Їх можна розділити на ієрархічну базу даних, мережеву базу даних і реляційну базу даних. Різні типи баз даних використовують різні методи зберігання, і внутрішня структура даних також відрізняється. Дані різних типів баз даних використовують різні методи організації. З розвитком технологій і зміною потреб користувачів перші два типи баз даних не можуть адаптуватися до розвитку часу і були ліквідовані. Зараз більшість баз даних на ринку є реляційними та нереляційними базами даних. Продукти даних на ринку включають Oracle, DB2, MySQL, SQL Server та інші продукти баз даних. Ці бази даних мають свої переваги. Вибираючи базу даних, не варто вибирати велику та високопродуктивну базу даних. Дуже важливо під час вибору бази даних вибрати більш відповідну базу даних відповідно до конкретного прикладного середовища.

База даних SQL Server — це реляційна база даних, розроблена Microsoft. Він має дуже відмінну продуктивність. Через ринок операційної системи

Windows її базу даних також віддає перевагу більшість користувачів. Microsoft SQL Server постійно оновлює версії, включаючи Microsoft SQL Server 2000, Microsoft SQL Server 2005 і Microsoft SQL Server 2008. Бази даних займають великий ринок. На даний момент останньою версією є SQL Server 2012, запущений у березні 2012 року. У цьому документі в якості даних системи вибрано базу даних SQL Server, яка може забезпечити ефективну підтримку бази даних для системи управління оцінкою ефективності співробітників підприємства.

РОЗДІЛ 3.

ПРОЕКТУВАННЯ СИСТЕМИ ОЦІНКИ ЕФЕКТИВНОСТІ РОБОТИ ПРАЦІВНИКІВ

Проектування комп'ютерної системи є ключовим етапом у процесі її розробки, оскільки саме на цьому етапі визначаються основні функціональні можливості, структура і компоненти майбутньої системи. У контексті завдання автоматизації оцінки якості та ефективності роботи працівників проектування повинно забезпечувати створення комплексного інструменту, який буде відповідати потребам користувачів, а також інтегруватися в існуючу інфраструктуру підприємства.

Сучасні інформаційні системи, зокрема системи оцінки працівників, базуються на багаторівневій архітектурі, що дозволяє розділяти відповідальність між різними компонентами системи. Таке розділення сприяє спрощенню розробки, обслуговування та масштабування системи, що особливо важливо для великих організацій із численними підрозділами.

Проектування починається з визначення вимог до системи, які формуються на основі аналізу бізнес-процесів підприємства. Вимоги визначають функціонал системи, її продуктивність, масштабованість, безпеку та зручність використання. Особливу увагу слід приділити потребам кінцевих користувачів, адже саме вони будуть взаємодіяти із системою щодня.

3.1. Постановка задачі для програмної реалізації

Задача проектування полягає у створенні програмного продукту, який дозволить підприємству автоматизувати оцінку продуктивності працівників. Система повинна вирішувати низку завдань, серед яких: збір, обробка та аналіз даних, відображення результатів у зрозумілому форматі, а також забезпечення інтерактивної взаємодії з користувачами.

Особливе значення на цьому етапі має визначення ключових функцій, які буде реалізовано у фронтенд-частині системи. До них належать:

- створення інтуїтивного інтерфейсу користувача, що дозволить легко вводити дані, переглядати результати та взаємодіяти з іншими компонентами системи;
- реалізація функціоналу для відображення графіків і таблиць, які візуалізують результати оцінювання;
- інтеграція з бекендом через API для передачі та отримання даних;
- забезпечення адаптивності інтерфейсу для роботи на різних пристроях, таких як комп'ютери, планшети та смартфони.

Для програмної реалізації необхідно врахувати сучасні тенденції у веб-розробці. Зокрема, слід використовувати гнучкі фреймворки та бібліотеки для створення адаптивного і динамічного інтерфейсу, що забезпечить високу продуктивність системи навіть за умови роботи з великими обсягами даних.

3.2. Архітектура системи

Архітектура системи є важливим компонентом проектування, оскільки вона визначає, як саме будуть взаємодіяти між собою різні компоненти системи. Для системи оцінки якості та ефективності роботи працівників пропонується багаторівнева архітектура, що включає три основні рівні: фронтенд, бекенд і базу даних.

Фронтенд є точкою взаємодії користувача із системою. Його основна функція — надання зручного та зрозумілого інтерфейсу, через який можна вводити дані, переглядати звіти та аналізувати результати. Для розробки фронтенду використовуються HTML, CSS і JavaScript, а також сучасні фреймворки, такі як React або Vue.js, які забезпечують динамічність і адаптивність інтерфейсу.

Бекенд відповідає за обробку запитів від фронтенду, виконання бізнес-логіки та взаємодію з базою даних. Він також забезпечує роботу API, через які здійснюється обмін даними між клієнтським і серверним рівнями. Для реалізації бекенду часто використовуються Node.js або Python (наприклад, фреймворк Django), які забезпечують високу продуктивність і масштабованість системи.

База даних слугує сховищем для всіх даних, що використовуються системою. Вона зберігає інформацію про працівників, їхні KPI, результати оцінки, графіки продуктивності та інші показники. Для реалізації бази даних використовуються реляційні системи управління базами даних, такі як MySQL або PostgreSQL, які забезпечують надійність і швидкість доступу до даних.

3.3. Загальна структура

Загальна структура системи (рис. 11) складається з трьох основних рівнів:

- Клієнтський рівень (фронтенд) — інтерфейс користувача. Забезпечує введення та відображення даних.
- Серверний рівень (бекенд) — обробка запитів і виконання бізнес-логіки.
- Рівень даних (база даних) — зберігання всієї інформації, необхідної для роботи системи.



Рисунок 11. Загальна структура системи

3.4. Компоненти та їх взаємодія

Основними компонентами фронтенду є форми для введення даних, таблиці для їх відображення та модулі для візуалізації. Наприклад, система повинна дозволяти користувачу додавати інформацію про працівників, вводити КРІ і переглядати графіки їхньої продуктивності.

Форми для введення даних повинні бути зручними й інтуїтивними. Для відображення даних використовуються таблиці з можливістю сортування та фільтрації, а для графіків — інтерактивні бібліотеки, такі як Chart.js.

Взаємодія компонентів фронтенду з бекендом здійснюється через API, яку зображено на рисунку 12. Наприклад, при додаванні нового працівника дані надсилаються на сервер, де вони зберігаються у базі даних. Після цього система може автоматично оновити таблицю на фронтенді, щоб користувач одразу бачив внесені зміни.

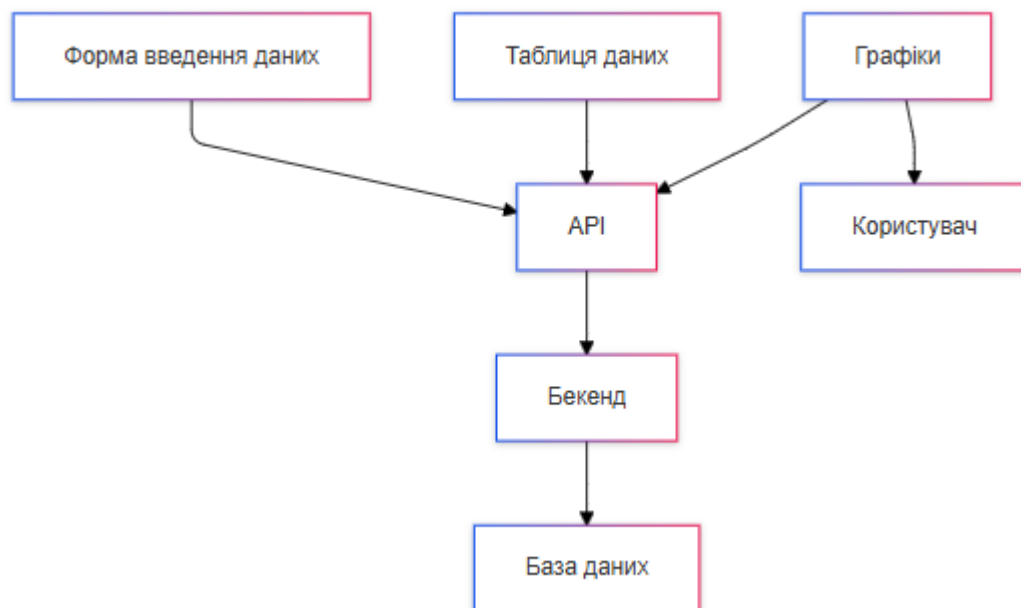


Рисунок 12. Взаємодія компонентів фронтенду з бекендом

3.5. Вибір технологій для реалізації

Вибір технологій для розробки комп'ютерної системи є важливим етапом, який визначає не лише продуктивність і масштабованість системи, але й зручність її обслуговування, адаптації до змін і інтеграції з іншими

інструментами. При виборі технологій враховуються такі фактори, як специфіка завдань, обсяги даних, особливості користувацького інтерфейсу та вимоги до безпеки.

3.5.1. Аналіз мов програмування

Python є одним із найпоширеніших виборів для розробки бекенд-частини систем, що працюють із великими обсягами даних. Його основними перевагами є велика кількість бібліотек для обробки даних (NumPy, Pandas), роботи з базами даних (SQLAlchemy), машинного навчання (TensorFlow, Scikit-learn) і створення REST API (Flask, Django). Python забезпечує високу продуктивність розробки завдяки простому синтаксису та широкому співтовариству розробників.

Java підходить для створення великих і складних систем, які потребують високої продуктивності та безпеки. Вона використовується в банківській сфері, корпоративних додатках та системах управління даними. Для створення бекенду можна використовувати такі фреймворки, як Spring і Hibernate, які забезпечують високу масштабованість і надійність.

JavaScript є основною мовою для розробки фронтенду. Завдяки фреймворкам, таким як React, Angular і Vue.js, JavaScript дозволяє створювати динамічні та інтерактивні інтерфейси. Крім того, JavaScript можна використовувати на сервері через Node.js, що дозволяє створювати універсальні додатки, де одна мова використовується як для фронтенду, так і для бекенду.

Інші мови:

- **C#** широко використовується для корпоративних додатків на платформі .NET, особливо в компаніях, які працюють з екосистемою Microsoft.
- **PHP** залишається популярним для веб-додатків завдяки простоті та великій кількості готових рішень, хоча його рідко використовують для високонавантажених систем.

3.5.2. Середовище розробки

Середовище розробки визначає зручність і ефективність роботи розробників. Для створення системи оцінки якості роботи працівників можуть використовуватися такі інструменти:

Visual Studio Code — багатофункціональний текстовий редактор, який підтримує численні мови програмування та має розширення для інтеграції з системами контролю версій, такими як Git.

PyCharm — інтегроване середовище розробки (IDE) для Python, яке забезпечує зручне автозаповнення коду, відлагодження і тестування.

IntelliJ IDEA — популярне середовище розробки для Java. Завдяки підтримці різних фреймворків і баз даних IntelliJ IDEA забезпечує швидку і якісну розробку.

WebStorm — спеціалізоване середовище для роботи з JavaScript і фреймворками, яке підтримує фронтенд- і бекенд-розробку.

Jupyter Notebook — інтерактивне середовище для роботи з Python, яке часто використовується для обробки даних і створення прототипів алгоритмів.

3.5.3. Спеціалізовані бібліотеки та фреймворки

Для прискорення розробки використовуються спеціалізовані бібліотеки та фреймворки. Їхній вибір залежить від задачі, яку потрібно вирішити.

Для бекенду:

- **Flask** — легкий фреймворк Python для створення REST API. Простий у використанні, ідеально підходить для невеликих проєктів.
- **Django** — потужний фреймворк Python із вбудованим ORM (Object-Relational Mapping), який дозволяє швидко створювати додатки.
- **Spring** — фреймворк для Java, який забезпечує високий рівень безпеки та масштабованості.

Для фронтенду:

- **React** — популярна бібліотека для створення динамічних інтерфейсів.

- **Vue.js** — гнучкий фреймворк, що підходить для невеликих і середніх проєктів.
- **Chart.js** — бібліотека для створення інтерактивних графіків і візуалізацій.
Для роботи з даними:
- **NumPy** і **Pandas** — бібліотеки Python для обробки великих масивів даних.
- **Scikit-learn** — бібліотека для машинного навчання.
- **Matplotlib** і **Seaborn** — для створення графіків.

3.6. Проектування бази даних

Проектування бази даних є важливим етапом у розробці системи, оскільки саме вона слугує центральним сховищем даних, які використовуються для роботи всіх компонентів системи. Для комп'ютерної системи оцінки якості та ефективності роботи працівників база даних повинна бути добре структурованою, забезпечувати швидкий доступ до даних і підтримувати всі необхідні функції для аналізу й обробки інформації.

База даних проєктується з урахуванням потреб системи в масштабованості, швидкодії та інтеграції з іншими модулями. Вона повинна забезпечувати зберігання як базових даних про працівників, так і результатів оцінки їхньої ефективності, а також даних для звітності.

3.6.1. Структура бази даних

Структура бази даних передбачає розділення інформації на логічно пов'язані таблиці. Це забезпечує зручність управління даними, скорочує дублювання інформації та підвищує ефективність запитів (рис. 13).

Основні таблиці для системи:

- **Employees (Працівники):** містить базову інформацію про працівників, включаючи ідентифікатор, ім'я, відділ та посаду.
- **KPI (Ключові показники ефективності):** зберігає інформацію про метрики оцінки ефективності, включаючи назву показника, його опис і допустимі значення.

- **Performance (Результати оцінки):** містить дані про результати оцінювання працівників за конкретними KPI, включаючи дату, оцінку та коментарі.
- **Departments (Відділи):** таблиця з інформацією про підрозділи компанії, які пов'язані з працівниками.

```
CREATE TABLE Employees (  
    EmployeeID INT AUTO_INCREMENT PRIMARY KEY,  
    Name VARCHAR(100) NOT NULL,  
    Position VARCHAR(50),  
    DepartmentID INT,  
    FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)  
);  
  
CREATE TABLE Departments (  
    DepartmentID INT AUTO_INCREMENT PRIMARY KEY,  
    Name VARCHAR(100) NOT NULL  
);  
  
CREATE TABLE KPI (  
    KpiID INT AUTO_INCREMENT PRIMARY KEY,  
    Name VARCHAR(100) NOT NULL,  
    Description TEXT  
);  
  
CREATE TABLE Performance (  
    PerformanceID INT AUTO_INCREMENT PRIMARY KEY,  
    EmployeeID INT,  
    KpiID INT,  
    Score DECIMAL(5, 2),  
    Date DATE NOT NULL,  
    Comments TEXT,  
    FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID),  
    FOREIGN KEY (KpiID) REFERENCES KPI(KpiID)  
);
```

Рисунок 13. SQL для створення бази даних

Ці таблиці забезпечують модульність і дозволяють легко змінювати структуру або додавати нові поля без впливу на інші таблиці.

3.6.2. Моделі даних (ER-діаграми)

Для кращого розуміння взаємозв'язків між таблицями у базі даних використовується модель "Сутність-Зв'язок" (ER). Вона дозволяє наочно продемонструвати, як саме дані пов'язані між собою, зображено на рисунку 14.

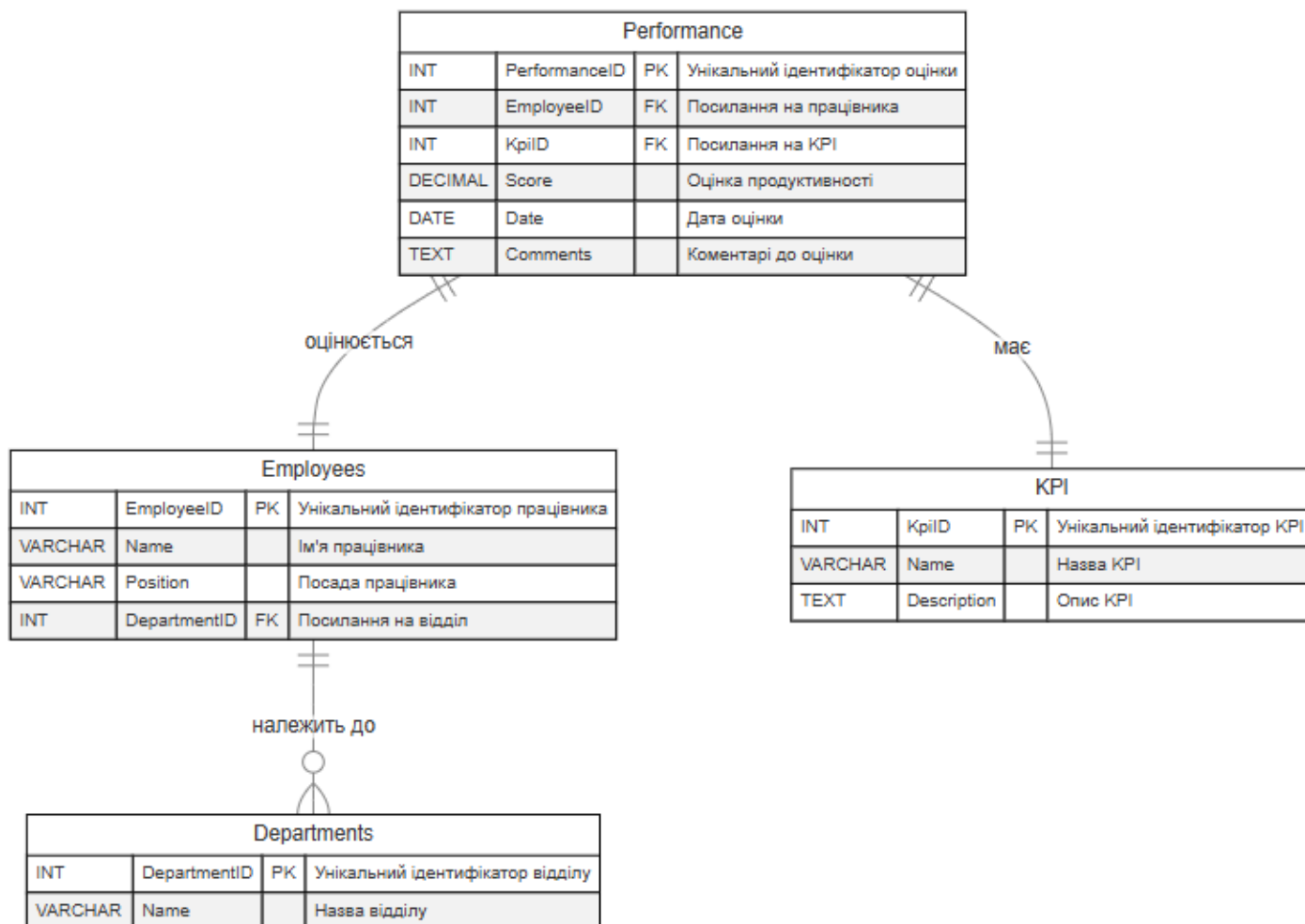


Рисунок 14. ER-діаграма

1. Таблиця Employees (Працівники):

- Кожен працівник має унікальний ідентифікатор EmployeeID.
- Працівники пов'язані з відділами через поле DepartmentID.
- Таблиця містить інформацію про ім'я, посаду та відділ.

2. Таблиця Departments (Відділи):

- Містить перелік усіх відділів, кожен із яких має унікальний ідентифікатор.

- Ця таблиця дозволяє уникнути дублювання назв відділів у таблиці працівників.

3. Таблиця KPI:

- Зберігає ключові показники ефективності (KPI), які використовуються для оцінки працівників.
- Опис поля Description дозволяє деталізувати критерії оцінювання.

4. Таблиця Performance (Результати оцінювання):

- Зберігає результати оцінки працівників за кожним KPI.
- Має зв'язки із таблицями Employees і KPI, що дозволяє відстежувати, як кожен працівник відповідає встановленим критеріям.

3.6.3. Переваги такої структури бази даних

1. Модульність: Таблиці є незалежними одна від одної, але мають чіткі зв'язки.
2. Масштабованість: Легко додавати нові таблиці або поля, наприклад, для оцінювання команд або проєктів.
3. Простота запитів: Реляційна структура дозволяє швидко формувати запити для отримання потрібних даних.
4. Захист від дублювання: Унікальні ключі забезпечують унікальність записів, а зовнішні ключі зберігають цілісність даних.

Ця структура бази даних є основою для побудови системи, яка ефективно зберігає, обробляє та аналізує інформацію про продуктивність працівників, забезпечуючи її зручність для використання та інтеграції.

3.7.Інтерфейс користувача

Інтерфейс користувача є критично важливим компонентом комп'ютерної системи, оскільки саме він забезпечує взаємодію між користувачем і функціональністю системи. У контексті системи для оцінки якості та ефективності роботи працівників інтерфейс повинен бути інтуїтивно зрозумілим, естетично привабливим і зручним для роботи. Для створення

ефективного інтерфейсу враховуються принципи UX/UI-дизайну, а також використовуються прототипи для попереднього тестування.

3.7.1. UX/UI-дизайн

UX (User Experience) і UI (User Interface) дизайн мають різні, але взаємопов'язані завдання. UX-дизайн фокусується на забезпеченні зручності взаємодії з системою, а UI-дизайн — на візуальному оформленні інтерфейсу.

Принципи UX-дизайну:

- **Зручність навігації:** Усі ключові функції, такі як додавання працівників, перегляд KPI і генерація звітів, повинні бути легко доступними через головне меню або панель управління.
- **Простота використання:** Форма введення даних, таблиці й графіки мають бути зрозумілими для користувачів із різним рівнем технічної підготовки.
- **Фокус на задачах:** Інтерфейс має допомагати користувачеві швидко виконувати основні задачі, наприклад, переглядати продуктивність команди або генерувати звіт.
- **Реакція системи:** Важливо, щоб користувачі отримували зворотний зв'язок на свої дії, наприклад, підтвердження успішного додавання працівника або сповіщення про помилку.

Принципи UI-дизайну:

- **Гармонійна палітра кольорів:** Використання корпоративних кольорів або нейтральної палітри, яка не перевантажує зір.
- **Зрозуміла типографіка:** Використання читабельних шрифтів, таких як Arial або Roboto, з достатньою контрастністю тексту.
- **Єдність стилю:** Усі елементи інтерфейсу (кнопки, таблиці, графіки) повинні мати однаковий стиль і гармонійно поєднуватися.
- **Адаптивність:** Інтерфейс має коректно відображатися на різних пристроях (десктопах, планшетах, смартфонах).

3.7.2. Створення прототипів

Прототипування є важливим етапом у розробці інтерфейсу, оскільки дозволяє оцінити функціональність і зручність інтерфейсу ще до його реалізації. Прототипи можуть бути як низької точності (наприклад, у вигляді ескізів), так і високої (створені за допомогою спеціалізованих інструментів).

Етапи створення прототипів:

1. **Визначення вимог:** Формуються завдання, які користувачі виконуватимуть через інтерфейс (наприклад, введення даних про працівників, перегляд графіків).
2. **Ескізи:** Створюються перші начерки інтерфейсу, які визначають розташування елементів (кнопок, таблиць, графіків).
3. **Створення інтерактивного прототипу:** За допомогою інструментів, таких як Figma або Adobe XD, створюється макет із можливістю взаємодії.
4. **Тестування:** Прототип тестується на цільовій аудиторії для виявлення проблем і отримання зворотного зв'язку.

Прототип у Figma (приклад концепції):

1. **Головний екран:**
 - **Ліве меню:** Містить розділи "Працівники", "KPI", "Звіти".
 - **Дашборд:** Відображає основні показники (кількість працівників, середній KPI).
 - **Кнопка додавання:** Дозволяє швидко додати нового працівника.
2. **Форма додавання:**
 - Поля для введення імені, KPI, дати та коментарів.
 - Кнопка "Зберегти".
3. **Таблиця даних:**
 - Список працівників із KPI.
 - Фільтри та сортування.

3.7.3. UX/UI-прототипи

UX/UI-прототип слугує як технічне й візуальне керівництво для розробників. Він допомагає:

- Зрозуміти, як користувачі будуть взаємодіяти із системою.
- Виявити можливі недоліки або труднощі у використанні інтерфейсу на етапі проектування.
- Отримати зворотний зв'язок від цільової аудиторії, що дозволяє внести покращення до того, як інтерфейс буде реалізовано у код.

Таким чином, цей прототип є базою для створення інтерактивної та функціональної системи, яка буде відповідати всім потребам користувачів, рисунок 15.

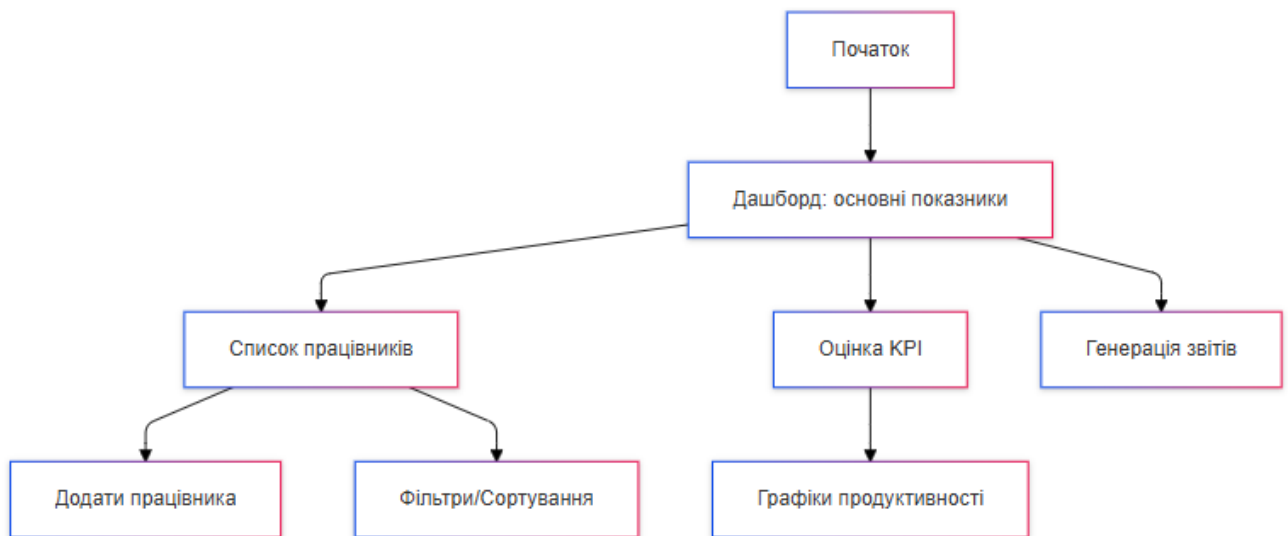


Рисунок 15. діаграма UX/UI

- **Початок роботи (Dashboard):** Дашборд є центральним елементом системи, який відображає основні показники продуктивності підприємства. Він забезпечує користувача швидким доступом до найважливішої інформації, такої як кількість працівників і середній KPI.
- **Список працівників:** Через цей розділ користувач може переглядати всі дані про працівників, редагувати їх або додавати нових.
- **Оцінка KPI:** Діаграма показує, що дані KPI доступні для перегляду як у табличному вигляді, так і через графіки, що дозволяє зручно аналізувати динаміку продуктивності.
- **Фільтри та сортування:** Функція фільтрування даних допомагає швидко знаходити інформацію про конкретних працівників чи певні показники.

- **Генерація звітів:** Цей розділ дозволяє створювати аналітичні звіти на основі зібраних даних, які можуть бути представлені у вигляді PDF чи Excel.

3.7.4. Використання HTML-прототипу в системі

HTML-прототип є початковою точкою для розробки кінцевого інтерфейсу (рис. 16). У реальній системі прототип буде доповнений:

- **Інтерактивністю:** Використання JavaScript для динамічного оновлення даних без перезавантаження сторінки. Наприклад, завантаження інформації про KPI через API.
- **Стилями:** Додавання більш складного оформлення через CSS або CSS-фреймворки, такі як Bootstrap або Tailwind CSS, щоб забезпечити сучасний і привабливий вигляд.
- **Інтеграцією із сервером:** Підключення до бекенд-системи (наприклад, на Flask чи Django), щоб отримувати й відправляти дані.

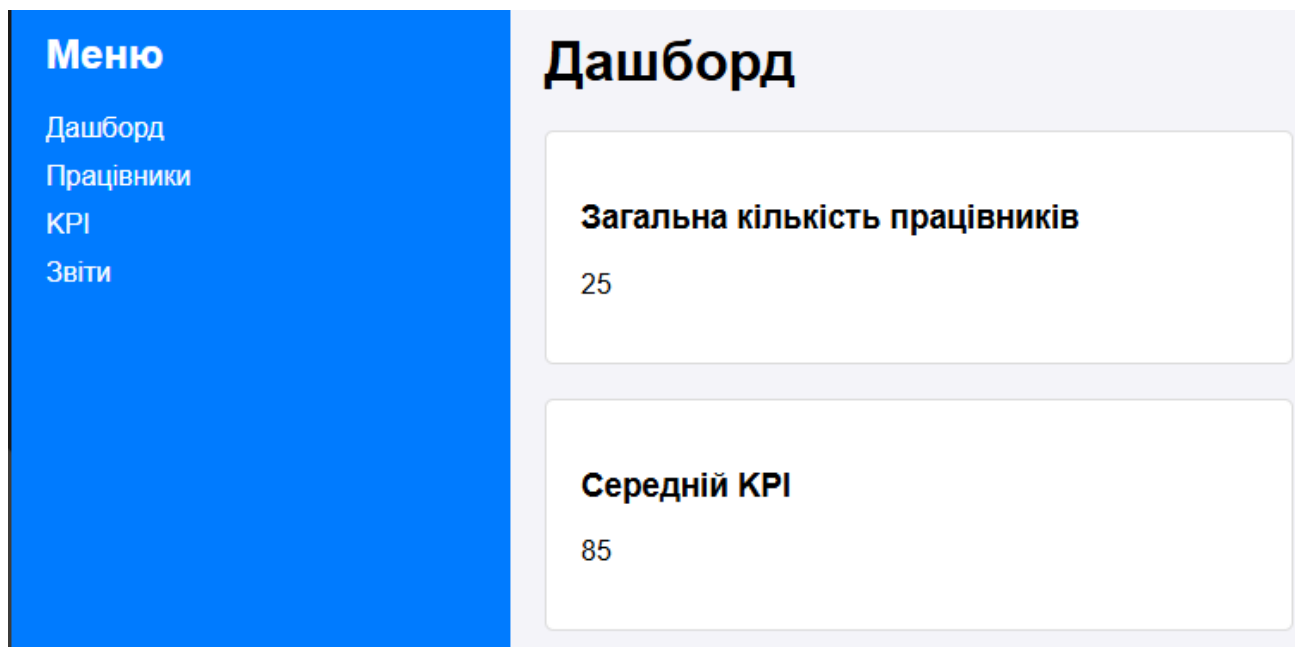


Рисунок 16. Прототип інтерфейсу користувача

На додатку С представлено базову структуру інтерфейсу користувача .

1. Меню навігації (sidebar):

- Включає основні розділи: "Дашборд", "Працівники", "КРІ" і "Звіти".
- Це статична панель, яка дозволяє швидко перемикатися між функціональними модулями системи.
- У реальній реалізації посилання будуть інтерактивними, спрямовуючи користувача до відповідних сторінок.

2. Дашборд:

- Відображає ключові показники: загальна кількість працівників і середній КРІ.
- Дані, показані в цьому розділі, можуть оновлюватися в реальному часі за допомогою JavaScript, що дозволить відображати актуальну інформацію.

3. Карти даних (card):

- Використовуються для компактного відображення окремих показників. Наприклад, карта "Загальна кількість працівників" демонструє числову інформацію, яка є зручною для перегляду.
- Карти забезпечують структуроване уявлення даних, що полегшує їх сприйняття.

3.8. Забезпечення безпеки даних

Захист даних працівників і результатів їхньої оцінки є ключовим аспектом у розробці комп'ютерної системи. Оскільки система оперує конфіденційною інформацією, важливо врахувати сучасні стандарти безпеки на всіх етапах: від збору й зберігання даних до їхньої передачі й аналізу (рис. 17).

Механізми безпеки:

1. Аутентифікація та авторизація:

- Використання системи облікових записів для забезпечення доступу лише авторизованим користувачам.
- Реалізація різних рівнів доступу (наприклад, адміністратор, керівник відділу, працівник).

2. Шифрування даних:

- Шифрування з'єднань між клієнтським і серверним рівнями за допомогою SSL/TLS.
- Шифрування даних у базі даних для забезпечення їхньої конфіденційності навіть у разі несанкціонованого доступу.

3. Контроль безпеки API:

- Використання токенів доступу (OAuth 2.0 або JWT) для верифікації запитів до API.
- Впровадження обмежень на частоту запитів для захисту від атак типу DDoS.

4. Регулярне резервне копіювання:

- Автоматизація резервного копіювання бази даних, щоб уникнути втрати даних у разі збоїв або атак.

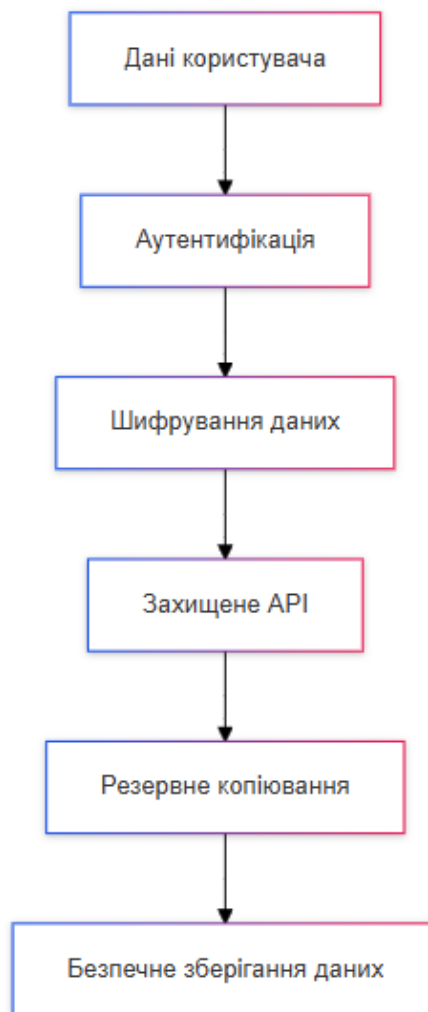


Рисунок 17. Діаграма для механізмів безпеки

3.9. Масштабованість і продуктивність системи

Система повинна бути здатною обробляти збільшені обсяги даних і кількість користувачів у міру зростання підприємства. Масштабованість забезпечує гнучкість у використанні системи для малих, середніх і великих організацій (рис. 18).



Рисунок 18. Діаграма масштабованості системи

Підходи до забезпечення масштабованості:

1. **Горизонтальне масштабування:** Додавання нових серверів для розподілу навантаження. Використання балансувальників навантаження (наприклад, NGINX) для рівномірного розподілу запитів між серверами.

2. **Вертикальне масштабування:** Збільшення потужності існуючих серверів (процесори, оперативна пам'ять).
3. **Кешування:** Використання кеш-систем, таких як Redis або Memcached, для зберігання часто запитуваних даних.
4. **Оптимізація запитів до бази даних:**
 - Впровадження індексів для прискорення пошуку.
 - Агрегування даних у запитах для зменшення кількості звернень до бази даних.

3.10. Тестування системи

Тестування є обов'язковим етапом перед впровадженням системи, оскільки дозволяє виявити помилки та недоліки, перевірити відповідність вимогам і забезпечити стабільну роботу (рис. 19-20).

Типи тестування:

1. **Функціональне тестування:**
 - Перевірка всіх функцій системи, таких як додавання працівників, перегляд КРІ, генерація звітів.
 - Тестування відповідності результатів очікуваним значенням.
2. **Навантажувальне тестування:**
 - Перевірка роботи системи за високих навантажень (велика кількість одночасних користувачів або обробка великих обсягів даних).
 - Використання інструментів, таких як JMeter або Locust.
3. **Юзабіліті-тестування:**
 - Оцінка зручності використання інтерфейсу.
 - Залучення тестових користувачів для отримання зворотного зв'язку.
4. **Безпекове тестування:**
 - Перевірка вразливостей, таких як SQL-ін'єкції або міжсайтовий скриптинг (XSS).
 - Використання інструментів, наприклад, OWASP ZAP.

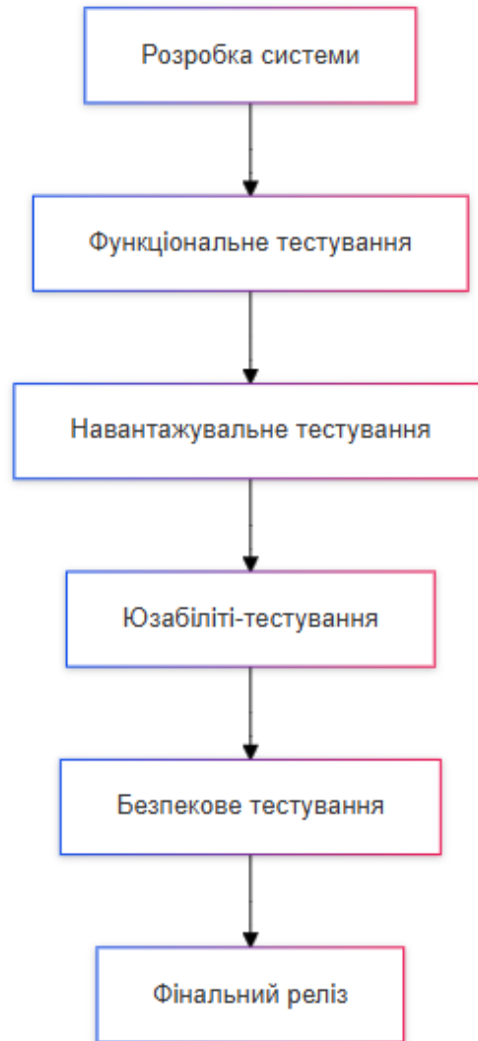


Рисунок 19. Діаграма процесу тестування

```
class TestEmployeeSystem(unittest.TestCase):  
    def test_add_employee(self):  
        # Тест на додавання працівника  
        employees = []  
        new_employee = {"name": "Employee A", "kpi": 85}  
        employees.append(new_employee)  
        self.assertEqual(len(employees), 1)  
        self.assertEqual(employees[0]["name"], "Employee A")  
        self.assertEqual(employees[0]["kpi"], 85)  
  
    def test_average_kpi(self):  
        # Тест на обчислення середнього КРІ  
        employees = [{"name": "A", "kpi": 80}, {"name": "B", "kpi": 90}]  
        avg_kpi = sum(emp["kpi"] for emp in employees) / len(employees)  
        self.assertEqual(avg_kpi, 85)  
  
if __name__ == "__main__":  
    unittest.main()
```

Рисунок 20. Приклад коду для автоматизованого тестування на Python

Переваги доданих пунктів:

1. **Безпека даних:** Забезпечує довіру користувачів до системи.
2. **Масштабованість:** Дозволяє системі працювати з більшими обсягами даних і користувачів.
3. **Якість системи:** Тестування забезпечує стабільність роботи та високу якість кінцевого продукту.

РОЗДІЛ 4.

ПРАКТИЧНЕ ВПРОВАДЖЕННЯ ТА ТЕСТУВАННЯ СИСТЕМИ

Розробка програмного забезпечення є ключовим етапом у створенні комп'ютерної системи для оцінки якості та ефективності роботи працівників. Цей процес включає реалізацію компонентів системи, опис основного коду та логіки функціонування. Основна увага приділяється інтеграції раніше спроектованих архітектурних рішень, бази даних і інтерфейсу користувача в єдину робочу систему.

4.1. Вибір технологій для реалізації

Вибір технологій для реалізації комп'ютерної системи є важливим етапом у розробці. Це визначає ефективність роботи системи, її масштабованість, а також зручність для розробників під час створення і подальшої підтримки. У контексті системи для оцінки якості та ефективності роботи працівників, необхідно врахувати різні аспекти: від вибору мов програмування до інструментів для роботи з даними та створення інтерфейсу користувача.

4.1.1. Аналіз мов програмування (Python, Java, etc.)

При виборі мови програмування для розробки комп'ютерної системи необхідно враховувати кілька факторів, таких як простота розробки, продуктивність, підтримка бібліотек і фреймворків, а також можливість масштабування.

1. Python:

- **Переваги:**
 - Легкість у навчанні та використанні. Python має простий і зрозумілий синтаксис, що робить його ідеальним для швидкої розробки прототипів і MVP.
 - Велика кількість бібліотек для роботи з даними, машинного навчання, веб-розробки та автоматизації (наприклад, NumPy, Pandas, Flask, Django, Scikit-learn).

- Підтримка як серверної, так і клієнтської частини. Зокрема, можна використовувати Flask або Django для створення REST API, що легко інтегрується з фронтенд-частиною.
 - **Недоліки:**
 - Порівняно низька швидкість роботи порівняно з мовами компільованими, такими як C++ або Java. Це може бути важливо в разі обробки великих обсягів даних або високих навантажень.
2. **Java:**
- **Переваги:**
 - Висока продуктивність і масштабованість. Java є одним із найпопулярніших варіантів для створення великих корпоративних додатків і веб-систем, де важлива висока продуктивність.
 - Підтримка багатьох фреймворків, таких як Spring і Hibernate, що робить її ідеальним вибором для побудови складних та надійних серверних рішень.
 - Кросплатформеність: Java працює на будь-якій платформі, де встановлений JVM.
 - **Недоліки:**
 - Більш складний синтаксис порівняно з Python.
 - Великі вимоги до пам'яті та ресурсоемність при роботі з великими обсягами даних.
3. **JavaScript (Node.js):**
- **Переваги:**
 - Одночасна розробка фронтенду і бекенду за допомогою однієї мови програмування.
 - Висока продуктивність завдяки асинхронному виконанню коду в Node.js.
 - Можливість створювати масштабовані, високопродуктивні API.
 - **Недоліки:**

- Node.js не найкращий вибір для ресурсомістких обчислень, де Python або Java можуть дати кращі результати.

4. C# (.NET):

○ **Переваги:**

- Підтримка об'єктно-орієнтованого підходу і потужні інструменти для розробки серверних додатків.
- Добре інтегрується з продуктами Microsoft, що може бути важливим для підприємств, які використовують екосистему Microsoft.

○ **Недоліки:**

- Дещо більший поріг входу для нових розробників.
- Не така велика кількість бібліотек і фреймворків, як у Python або Java.

4.1.2. Середовище розробки.

Вибір середовища розробки (IDE) і інструментів для кодування важливий для швидкості і якості розробки. Ось деякі з популярних варіантів:

1. **Visual Studio Code (VSCode):**

- Легкий, зручний, і багатофункціональний текстовий редактор, який підтримує безліч мов програмування, включаючи Python, JavaScript і C#.
- Велика кількість плагінів для інтеграції з базами даних, системами контролю версій (Git), а також для роботи з фреймворками, такими як Flask або Django.

2. **PyCharm:**

- Спеціалізоване середовище для розробки на Python. Містить безліч функцій для роботи з Django, Flask, а також підтримує тестування і інтеграцію з базами даних.

3. **IntelliJ IDEA:**

- Ідеальний вибір для розробки на Java. Пропонує безліч інструментів для роботи з фреймворками (Spring, Hibernate) і системами баз даних.

4. WebStorm:

- IDE для розробки на JavaScript. Відмінно підходить для фронтенд- і бекенд-розробки за допомогою Node.js.

5. Eclipse:

- Один з найстаріших IDE для Java-розробки, який має велику кількість плагінів для інтеграції з різними інструментами та фреймворками.

4.1.3. Спеціалізовані бібліотеки та фреймворки

Для ефективної розробки комп'ютерної системи необхідно використовувати бібліотеки та фреймворки, які дозволяють значно знизити час розробки та забезпечити високу продуктивність і зручність у роботі.

1. Для бекенду:

- **Flask** (для Python): Легкий мікрофреймворк для розробки RESTful API. Підходить для невеликих проєктів і швидкої розробки.
- **Django** (для Python): Потужний фреймворк для створення веб-додатків із вбудованими функціями, такими як аутентифікація, робота з базами даних та шаблонами.
- **Spring** (для Java): Платформа для створення складних корпоративних додатків. Підтримує розподілені системи, масштабованість та безпеку.
- **Express** (для Node.js): Легкий і гнучкий фреймворк для створення веб-додатків та API на Node.js.

2. Для фронтенду:

- **React**: Бібліотека для створення інтерактивних інтерфейсів користувача. Швидка і зручна для розробки компонентів.
- **Vue.js**: Легкий фреймворк для розробки динамічних інтерфейсів. Має низький поріг входу і чудову документацію.
- **Angular**: Платформа для створення складних веб-додатків. Має все необхідне для масштабних проєктів.

3. Для роботи з даними:

- **Pandas** (для Python): Бібліотека для маніпулювання і аналізу даних. Ідеальна для обробки таблиць, CSV, баз даних.
- **NumPy** (для Python): Бібліотека для роботи з багатовимірними масивами і матрицями, що корисно для статистичних обчислень.
- **Chart.js**: Легкий у використанні інструмент для побудови графіків та діаграм у веб-додатках.

4. Для машинного навчання:

- **Scikit-learn** (для Python): Бібліотека для машинного навчання, що містить моделі для класифікації, регресії, кластеризації.
- **TensorFlow** (для Python): Потужний фреймворк для глибинного навчання, що використовується для створення нейронних мереж.

5. Для роботи з базами даних:

- **SQLAlchemy** (для Python): ORM для роботи з реляційними базами даних, який дозволяє використовувати Python для роботи з SQL-запитами.
- **Hibernate** (для Java): ORM для роботи з реляційними базами даних у Java, що дозволяє працювати з об'єктами, не пишучи SQL-код.

4.1.4. Підсумки вибору технологій

Python: Є чудовим вибором для бекенд-розробки завдяки своїй простоті, величезній кількості бібліотек і підтримці наукових обчислень і машинного навчання.

Java: Використовується для створення масштабованих корпоративних рішень, де важлива висока продуктивність і надійність.

JavaScript/Node.js: Ідеальний вибір для інтеграції фронтенду та бекенду в єдину систему за допомогою однієї мови програмування.

React, Vue.js, Angular: Це фреймворки для фронтенду, які дозволяють створювати інтерактивні й масштабовані веб-додатки.

Flask/Django (Python) і Spring (Java): Бекенд-фреймворки для створення REST API та обробки запитів.

4.2. Реалізація компонентів системи

Цей підрозділ детально описує компоненти, які складають систему, включаючи основний код для їхньої реалізації.

4.2.1. Опис основного коду

Основний код системи охоплює функціонал бекенду, бази даних і фронтенду. Основним завданням є інтеграція цих частин для забезпечення їхньої узгодженої роботи.

Бекенд: Реалізація REST API на Flask

Приклад реалізації маршруту для додавання працівників у базу даних (рис. 21)

```

from flask import Flask, request, jsonify
from flask_sqlalchemy import SQLAlchemy

app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///employees.db'
db = SQLAlchemy(app)

# Модель працівника
class Employee(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(100), nullable=False)
    kpi = db.Column(db.Float, nullable=False)

# Ініціалізація бази даних
@app.before_first_request
def create_tables():
    db.create_all()

# Додавання працівника
@app.route('/add_employee', methods=['POST'])
def add_employee():
    data = request.json
    new_employee = Employee(name=data['name'], kpi=data['kpi'])
    db.session.add(new_employee)
    db.session.commit()
    return jsonify({'message': 'Employee added successfully'})

# Отримання працівників
@app.route('/employees', methods=['GET'])
def get_employees():
    employees = Employee.query.all()
    return jsonify([{'id': emp.id, 'name': emp.name, 'kpi': emp.kpi} for emp in employees])

if __name__ == '__main__':
    app.run(debug=True)

```

Рисунок 21. Приклад реалізації маршруту для додавання працівників у базу даних

Цей код реалізує базовий API-сервер для управління даними працівників. Основні функції коду:

1. Ініціалізація програми Flask(рис. 22):

```
app = Flask(__name__)
```

Рисунок 22. Ініціалізація програми Flask

Це створює сервер додатку. Flask забезпечує просту та ефективну основу для розробки веб-додатків.

2. Налаштування бази даних (рис. 23):

```
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///employees.db'  
db = SQLAlchemy(app)
```

Рисунок 23. Налаштування БД.

1. SQLALCHEMY_DATABASE_URI: вказує шлях до бази даних. У цьому випадку використовується SQLite, файлова реляційна база даних.

2. SQLAlchemy: це ORM (Object-Relational Mapping), який спрощує роботу з базами даних через Python.

3. Опис моделі бази даних (рис. 24)

```
class Employee(db.Model):  
    id = db.Column(db.Integer, primary_key=True)  
    name = db.Column(db.String(100), nullable=False)  
    kpi = db.Column(db.Float, nullable=False)
```

Рисунок 24. Модель БД.

1. Employee: клас, який описує структуру таблиці "Employees" у базі даних.

2. id: унікальний ідентифікатор кожного запису.

3. name: поле для зберігання імені працівника.

4. kpi: поле для оцінки KPI.

4. Створення таблиць (рис. 25)

```
@app.before_first_request  
def create_tables():  
    db.create_all()
```

Рисунок 25. Створення таблиці

Ця функція автоматично створює таблиці в базі даних під час першого запуску сервера.

5. Реалізація маршруту для додавання працівників (рис. 26)

```
new_employee = Employee(name=data['name'], kpi=data['kpi'])
db.session.add(new_employee)
db.session.commit()
```

Рисунок 26. Реалізація маршруту додавання працівників

- Цей маршрут приймає POST-запити для додавання нового працівника.
- Отримані дані додаються до бази даних через ORM SQLAlchemy:

6. Реалізація маршруту для отримання працівників (рис. 27)

```
 jsonify([{'id': emp.id, 'name': emp.name, 'kpi': emp.kpi} for emp in employees])
```

Рисунок 27. Реалізація маршруту для отримання працівників

- Цей маршрут повертає список усіх працівників у форматі JSON.
- Відповідь формується шляхом перетворення кожного об'єкта Employee у словник:

Цей код демонструє, як створити API для додавання та отримання даних про працівників. Він використовує Flask як основний фреймворк і SQLite як базу даних.

Фронтенд: Взаємодія з API за допомогою JavaScript

Приклад завантаження працівників із сервера та відображення їх у таблиці (рис. 28):

```

<script>
  async function loadEmployees() {
    const response = await fetch('http://127.0.0.1:5000/employees');
    const employees = await response.json();
    const tableBody = document.querySelector('#employeeTable tbody');
    tableBody.innerHTML = '';
    employees.forEach(emp => {
      const row = `<tr>
        <td>${emp.id}</td>
        <td>${emp.name}</td>
        <td>${emp.kpi}</td>
      </tr>`;
      tableBody.innerHTML += row;
    });
  }

  // Завантаження працівників при завантаженні сторінки
  document.addEventListener('DOMContentLoaded', loadEmployees);
</script>

```

Рисунок 28. Взаємодія з API за допомогою JavaScript

Цей код демонструє, як завантажувати дані про працівників із бекенду та відображати їх у таблиці на вебсторінці (рис. 29-31).

```

async function loadEmployees() {
  const response = await fetch('http://127.0.0.1:5000/employees');
  const employees = await response.json();
}

```

Рисунок 29. Асинхронне завантаження працівників

- Використовується метод `fetch` для надсилання запиту на API `/employees`.
- Результати запиту (список працівників) отримуються у форматі JSON.

```

const tableBody = document.querySelector('#employeeTable tbody');
tableBody.innerHTML = '';
employees.forEach(emp => {
  const row = `<tr>
    <td>${emp.id}</td>
    <td>${emp.name}</td>
    <td>${emp.kpi}</td>
  </tr>`;
  tableBody.innerHTML += row;
});

```

Рисунок 30 Додавання рядків до таблиці

- HTML-код для кожного рядка таблиці створюється динамічно за допомогою шаблонних рядків.
- Дані працівника додаються до відповідних комірок таблиці (id, name, kpi).
- innerHTML використовується для оновлення вмісту таблиці.

```
document.addEventListener('DOMContentLoaded', loadEmployees);
```

Рисунок 31 Автоматичне завантаження даних

Код забезпечує, що функція loadEmployees виконується автоматично після завантаження сторінки.

Цей код дозволяє отримувати дані з бекенду та динамічно відображати їх на сторінці у вигляді таблиці.

4.2.2. Логіка функціонування системи

Логіка функціонування системи описує, як саме користувач взаємодіє із системою, і як вона забезпечує виконання основних задач (рис. 32).

1. Додавання працівників:

- Користувач вводить дані про нового працівника через форму на веб-сторінці.
- Дані передаються на сервер через API.
- Сервер зберігає ці дані в базі даних і підтверджує успішність операції.

2. Отримання списку працівників:

- Коли користувач відкриває список працівників, фронтенд надсилає запит до API.
- Сервер повертає список працівників у форматі JSON.
- Фронтенд відображає отримані дані у вигляді таблиці.

3. Обчислення середнього KPI:

- Сервер обробляє запит, виконуючи обчислення середнього значення KPI в базі даних.

- Результат повертається фронтенду, який відображає його на дашборді.

4. Генерація звітів:

- Користувач натискає кнопку "Згенерувати звіт".
- Сервер формує звіт (наприклад, у форматі PDF) на основі даних у базі та надсилає його користувачеві.

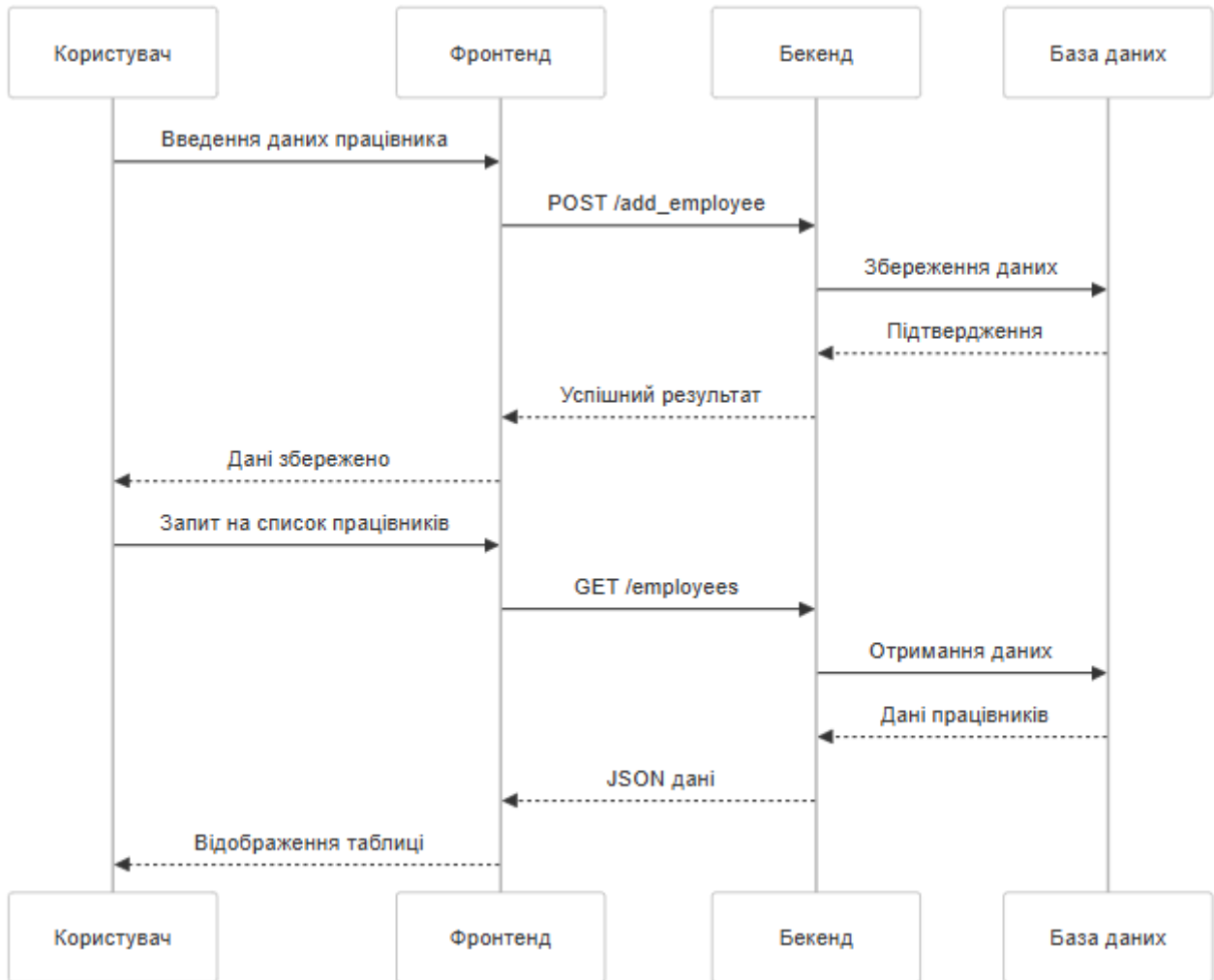


Рисунок 32. Діаграма логіки функціонування системи

- **Взаємодія:** Система забезпечує прямий обмін даними між користувачем, фронтендом і бекендом.
- **Обробка:** Логіка бекенду відповідає за валідацію, обробку та збереження даних, а також за виконання аналітичних операцій.

- **Візуалізація:** Фронтенд гарантує зручне відображення даних та інтерактивність інтерфейсу.

4.3. Інтеграція з іншими системами

Інтеграція з іншими системами є важливим етапом у розробці комп'ютерної системи, оскільки це дозволяє обмінюватися даними з зовнішніми додатками, аналітичними платформами, ERP- та CRM-системами. Правильно реалізована інтеграція підвищує функціональність системи, розширює її можливості і спрощує взаємодію з іншими програмними рішеннями.

4.3.1. API та веб-сервіси

API (Application Programming Interface) — це набір протоколів і інструментів, який дозволяє різним програмам взаємодіяти між собою. У сучасних системах використовуються REST API або GraphQL для обміну даними між клієнтською та серверною частинами або іншими системами.

1. Особливості:

REST (Representational State Transfer) — це стандарт для створення веб-сервісів. Він використовує HTTP-запити (GET, POST, PUT, DELETE) для взаємодії з ресурсами (рис. 33).

2. Приклад інтеграції:

- **API зовнішньої системи:** Система може отримувати дані про KPI працівників із зовнішньої HR-платформи.
- **Реалізація:** Фронтенд надсилає запит на бекенд, а бекенд звертається до зовнішнього API.

```
import requests

def get_crm_data():
    url = "https://api.crmexample.com/employees"
    headers = {"Authorization": "Bearer YOUR_API_TOKEN"}
    response = requests.get(url, headers=headers)
    if response.status_code == 200:
        return response.json()
    else:
        raise Exception(f"Failed to fetch data: {response.status_code}")
```

Рисунок 33. Реалізації REST API для інтеграції з CRM

Python-код виконує запит до API зовнішньої CRM-системи, отримуючи список працівників.

GraphQL:

1. Особливості:

GraphQL дозволяє клієнтам запитувати лише ті дані, які потрібні, у зручному форматі. Це зменшує кількість запитів до сервера.

2. Приклад запиту зображено на рисунку 34:

```
query {  
  employees {  
    id  
    name  
    department  
    kpi  
  }  
}
```

Рисунок 34. Приклад запиту на GraphQL

GraphQL є ідеальним вибором для складних інтеграцій, де потрібно працювати з великою кількістю ресурсів.

4.3.2. Методи обміну даними

Обмін даними між системами є основою інтеграції (рис. 35). Основні методи обміну даними включають:

1. JSON (JavaScript Object Notation):

- Найпоширеніший формат обміну даними між API та системами.
- Легко читається, підтримується практично всіма мовами програмування.

```
{
  "employees": [
    {
      "id": 1,
      "name": "John Doe",
      "department": "Sales",
      "kpi": 85.5
    },
    {
      "id": 2,
      "name": "Jane Smith",
      "department": "Marketing",
      "kpi": 92.3
    }
  ]
}
```

Рисунок 35. JSON-відповіді

Цей формат може бути переданий між системами через HTTP-запити або вебсокет.

2. XML (eXtensible Markup Language):

- Використовується в старих або специфічних корпоративних системах (рис. 36).
- Хоча XML є більш громіздким, ніж JSON, він все ще залишається актуальним у деяких галузях.

```
<employees>
  <employee>
    <id>1</id>
    <name>John Doe</name>
    <department>Sales</department>
    <kpi>85.5</kpi>
  </employee>
  <employee>
    <id>2</id>
    <name>Jane Smith</name>
    <department>Marketing</department>
    <kpi>92.3</kpi>
  </employee>
</employees>
```

Рисунок 36. XML-відповіді

3. WebSockets:

- Використовуються для обміну даними в реальному часі.
- Наприклад, WebSockets можуть передавати оновлення КРІ працівників у режимі реального часу, що дозволяє менеджерам миттєво реагувати на зміни (рис. 37).

```
const socket = new WebSocket('ws://example.com/kpi-updates');

socket.onmessage = function(event) {
  const data = JSON.parse(event.data);
  console.log("Оновлений КРІ:", data);
};
```

Рисунок 37. Обмін даними через WebSockets

4. Інтеграція через файли (CSV, Excel):

- Використовується для експорту або імпорту даних із систем, які не підтримують API.
- Формати CSV та Excel дозволяють обмінюватися великими обсягами даних у табличному вигляді (рис. 38).

```
import csv

def read_csv(file_path):
    with open(file_path, mode='r') as file:
        csv_reader = csv.DictReader(file)
        for row in csv_reader:
            print(row)
```

Рисунок 38. обробка CSV на Python

5. Message Queue (черги повідомлень):

- Використовуються для асинхронної передачі даних між компонентами системи.
- Наприклад, RabbitMQ або Apache Kafka можна використовувати для обробки великих обсягів даних у багатокористувацьких системах (рис. 39).

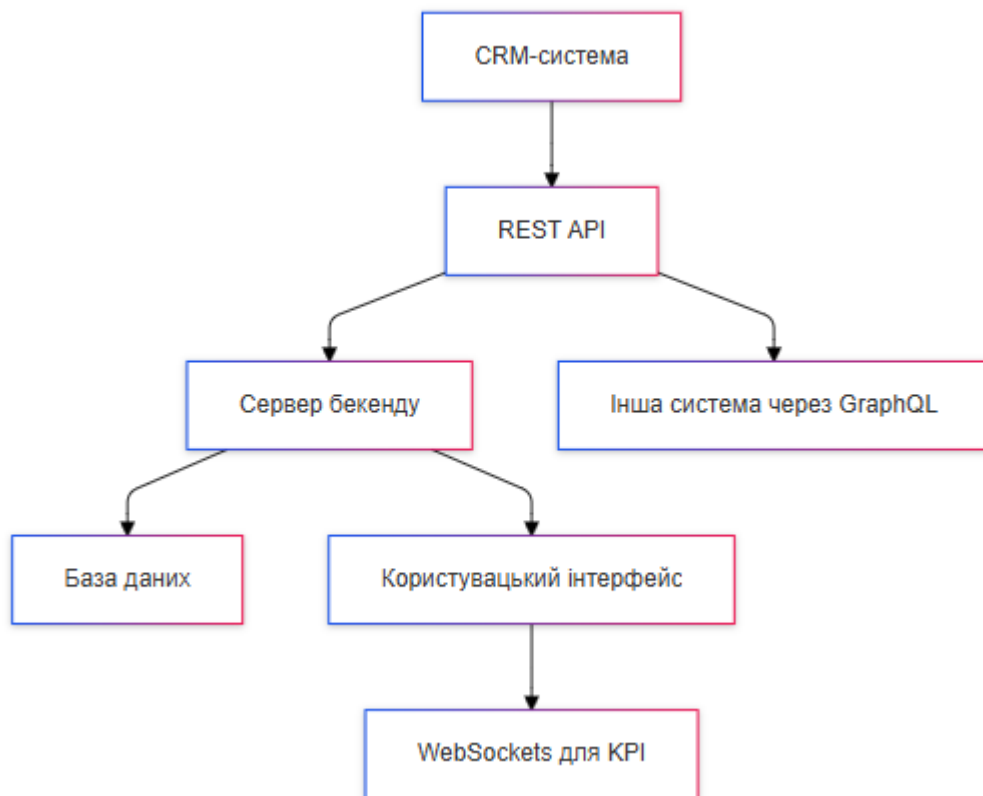


Рисунок 39. Діаграма обміну даними між системами

Переваги інтеграції через API та обмін даними:

1. **Швидкість та ефективність:** API забезпечують швидкий і безпечний обмін даними, особливо через REST або GraphQL.
2. **Гнучкість:** Можна налаштувати обмін даними залежно від потреб, використовуючи JSON, XML, або CSV.
3. **Масштабованість:** Інструменти, такі як WebSockets і Message Queue, дозволяють працювати з великими обсягами даних і забезпечувати обробку в реальному часі.
4. **Сумісність:** Веб-сервіси та API дозволяють легко підключати нові системи без необхідності вносити значні зміни в існуючу інфраструктуру.

Інтеграція через API та продумані методи обміну даними забезпечують ефективну взаємодію між системами, що розширює можливості комп'ютерної системи.

4.4. Тестування системи

Тестування є обов'язковим етапом розробки, що забезпечує високу якість програмного забезпечення. Воно дозволяє виявити помилки, перевірити коректність роботи компонентів системи, а також оцінити продуктивність і безпеку.

4.4.1. Методи тестування

У процесі тестування системи використовуються різні методи, залежно від рівня перевірки та цілей. Основними методами є юніт-тестування і інтеграційне тестування.

Юніт-тестування (Unit Testing) (рис. 40):

- **Опис:**

Юніт-тестування перевіряє окремі модулі або функції системи, щоб переконатися, що вони працюють належним чином у ізольованому середовищі.

- **Інструменти:**

- Для Python: unittest, pytest.
- Для JavaScript: Jest, Mocha.
- Для Java: JUnit.

- **Приклад юніт-тесту на Python:**

```
import unittest

def calculate_average_kpi(employees):
    if not employees:
        return 0
    return sum(emp['kpi'] for emp in employees) / len(employees)

class TestEmployeeSystem(unittest.TestCase):
    def test_calculate_average_kpi(self):
        employees = [{'name': 'A', 'kpi': 80}, {'name': 'B', 'kpi': 90}]
        result = calculate_average_kpi(employees)
        self.assertEqual(result, 85)

        empty_employees = []
        result = calculate_average_kpi(empty_employees)
        self.assertEqual(result, 0)

if __name__ == '__main__':
    unittest.main()
```

Рисунок 40. Перевірка функції обчислення середнього КРІ працівників

Інтеграційне тестування (Integration Testing):

- **Опис:**

Інтеграційне тестування перевіряє взаємодію між компонентами системи, такими як база даних, сервер і фронтенд (рис. 41).

- **Інструменти:**

- Для API: Postman, RestAssured.
- Для комплексного тестування: Selenium, Cypress.

- **Приклад інтеграційного тесту API на Python:**

```
import requests

def test_add_employee_api():
    url = "http://127.0.0.1:5000/add_employee"
    data = {"name": "John Doe", "kpi": 95}
    response = requests.post(url, json=data)
    assert response.status_code == 200
    assert response.json().get("message") == "Employee added successfully"
```

Рисунок 41. Перевірка API чи додавання працівників працює належним чином

Інші види тестування:

1. **Навантажувальне тестування:** Перевіряє, як система працює за високого навантаження (велика кількість користувачів або запитів). Наприклад, використовується JMeter або Locust.
2. **Юзабіліті-тестування:** Перевіряє, наскільки зручний інтерфейс системи. Тестувальники виконують типові завдання, щоб оцінити інтуїтивність і простоту взаємодії.
3. **Безпекове тестування:** Перевіряє захист даних від атак, таких як SQL-ін'єкції або XSS. Для цього використовуються інструменти, такі як OWASP ZAP.

4.4.2. Аналіз результатів тестування

Після проведення тестів результати аналізуються для виявлення проблем і оцінки відповідності системи вимогам.

Аналіз результатів юніт-тестування:

- Якщо окремі функції не проходять тести, це вказує на проблеми в їхній логіці.
- Наприклад, якщо тест функції обчислення середнього КРІ повертає неправильний результат, слід перевірити обробку порожніх списків або формат даних.

Аналіз результатів інтеграційного тестування:

- Тести можуть виявити проблеми з узгодженістю між компонентами, наприклад, некоректну обробку запитів у API або помилки взаємодії з базою даних.
- Наприклад, якщо API повертає помилку при додаванні працівника, слід перевірити валідацію вхідних даних і роботу з базою даних.

Формування звіту:

- Усі результати тестування документуються. Звіт включає:
 - Кількість виконаних тестів.
 - Кількість успішних і невдалих тестів.
 - Опис знайдених помилок із вказівкою місця їх виникнення.

Використання автоматизованих звітів:

- Інструменти, такі як pytest, автоматично генерують звіти у зручному форматі, наприклад, HTML.
- Звіт може виглядати так:
 - **Пройдено:** 95 тестів.
 - **Помилки:** 5 тестів.
 - **Причини:** некоректна обробка вхідних даних у API, проблеми з підключенням до бази даних.



Рисунок 42. Діаграма процесу тестування

1. **Юніт-тестування:** Дозволяє переконатися в правильній роботі окремих функцій.
2. **Інтеграційне тестування:** Забезпечує перевірку взаємодії між компонентами системи.
3. **Навантажувальне, юзабіліті- та безпекове тестування:** Перевіряють роботу системи в умовах реального використання.

Аналіз результатів тестування допомагає вдосконалити систему, усунути недоліки та гарантувати її надійну роботу у продакшн-середовищі.

РОЗДІЛ 5.

ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ПРОЕКТУ

5. Економічне обґрунтування проекту

Економічне обґрунтування проекту є невід'ємною частиною розробки програмного забезпечення, оскільки дозволяє оцінити його рентабельність. Воно включає аналіз витрат на розробку, впровадження та підтримку системи, а також оцінку економічного ефекту від її використання, такого як підвищення продуктивності та зниження витрат.

5.1. Вартість розробки та впровадження системи

Оцінка вартості включає прямі й непрямі витрати, пов'язані з розробкою, тестуванням, впровадженням і підтримкою системи.

Складові витрат:

1. Заробітна плата розробників:

- **Команда:** 1 бекенд-розробник, 1 фронтенд-розробник, 1 QA-інженер.
- **Час роботи:** 4 місяці.
- **Середня ставка:** \$2,500/місяць на одного розробника.
- **Загальна сума:** \$30,000 (3 спеціалісти × 4 місяці × \$2,500).

2. Програмне забезпечення та інструменти:

- Використання відкритих бібліотек і фреймворків (безкоштовно).
- Купівля IDE (якщо необхідно, наприклад, IntelliJ IDEA): \$300.

3. Інфраструктура:

- Сервер для хостингу системи: \$50/місяць.
- Хмарні сервіси (AWS, Azure): \$100/місяць.
- Загальна сума за 4 місяці: \$600.

4. Тестування та впровадження:

- Витрати на юзабіліті-тестування (залучення тестувальників): \$1,000.
- Навчання співробітників для роботи із системою: \$2,000.

Загальні витрати:

Заробітна плата:	\$30,000
Програмне забезпечення:	\$300
Інфраструктура:	\$600
Тестування і навчання:	\$3,000

Підсумок:

Загальна вартість розробки та впровадження системи оцінюється в **\$33,900**.

5.2. Оцінка економічного ефекту

Ефект від впровадження системи оцінюється через підвищення продуктивності роботи працівників і зниження витрат на управління процесами.

5.2.1. Підвищення продуктивності

1. **Автоматизація процесів:** Система дозволяє зменшити час, витрачений на ручну обробку даних і складання звітів.
 - **Приклад:** Менеджер, який витрачав 10 годин на підготовку звіту щомісяця, скорочує цей час до 1 години.
 - **Економія:** 9 годин × \$20/година = \$180 на одного менеджера щомісяця.
2. **Швидкість прийняття рішень:** Завдяки аналітичним інструментам і автоматизованим звітам керівники можуть приймати рішення швидше. Це підвищує ефективність управління.
3. **Оцінка ROI (Return on Investment):**
 - **Загальна економія часу:** 100 годин/місяць для всієї команди.
 - **Загальна економія коштів:** \$2,000/місяць (при середній ставці \$20/година).
 - **Річний ефект:** \$24,000.

5.2.2. Зниження витрат

Оптимізація ресурсів:

Система дозволяє точно оцінювати KPI працівників, що сприяє більш ефективному розподілу завдань і ресурсів.

Приклад: Зменшення витрат на зайві тренінги для персоналу, які не впливають на підвищення продуктивності.

Економія: \$5,000/рік.

Зниження витрат на адміністративну роботу:

Завдяки автоматизації оцінювання продуктивності зменшується потреба в адміністративному персоналі.

Приклад: Скорочення 1 посади адміністратора (\$15,000/рік).

Загальний економічний ефект:

Підвищення продуктивності: \$24,000/рік.

Зниження витрат: \$20,000/рік (оптимізація ресурсів і скорочення адміністративної роботи).

Річний економічний ефект: \$44,000.

Таблиця 5.1. Порівняння витрат і вигод

Показник	Сума (\$)
Вартість розробки системи	\$33,900
Річний економічний ефект	\$44,000
Чистий прибуток за рік	\$10,100

Висновок до розділу

Впровадження комп'ютерної системи для оцінки якості та ефективності роботи працівників є економічно вигідним.

- Окупність інвестицій:** Система окупається менш ніж за рік, оскільки економічний ефект (\$44,000/рік) перевищує витрати на її розробку та впровадження (\$33,900).
- Підвищення продуктивності:** Завдяки автоматизації процесів скорочується час, необхідний для обробки даних і прийняття рішень.
- Зниження витрат:** Оптимізація ресурсів та автоматизація дозволяють суттєво зменшити операційні витрати.

Це свідчить про високий рівень ефективності розробленої системи як з точки зору підвищення продуктивності, так і зниження витрат.

Висновки

Розробка автоматизованої системи базується на інтеграції різноманітних джерел даних та аналітичних підходів. Впровадження таких систем дозволяє уникнути суб'єктивності, властивої традиційним методам оцінки, шляхом використання алгоритмів машинного навчання, які аналізують великі обсяги даних у реальному часі. Це створює передумови для швидкого реагування на проблеми в управлінні персоналом та оптимізації використання людських ресурсів.

Система використовує багатокритеріальні моделі аналізу, що дозволяють враховувати кількісні та якісні аспекти роботи працівників. Зокрема, впровадження KPI за методологією SMART забезпечує конкретність, вимірюваність і адаптивність оцінок. Використання методів класифікації та прогнозування, таких як дерева рішень і аналіз ієрархій, дозволяє виявляти тенденції та формувати рекомендації для підвищення ефективності як окремих працівників, так і команд.

Система проектується з урахуванням можливості інтеграції з існуючими корпоративними платформами, такими як ERP, CRM, та інші інструменти управління завданнями. Це забезпечує збір і аналіз даних із різних джерел, формування звітності та створення прозорих процесів управління. Особлива увага приділяється збереженню конфіденційності даних відповідно до міжнародних стандартів, таких як GDPR.

Система є універсальним рішенням, яке може бути адаптоване до потреб підприємств різного масштабу і галузей. Її функціонал включає не лише оцінку продуктивності, але й створення персоналізованих планів розвитку працівників. Вона сприяє зростанню мотивації, підвищенню ефективності прийняття рішень і адаптації до швидкозмінних умов бізнес-середовища, забезпечуючи конкурентоспроможність підприємств у довгостроковій перспективі.

Список використаних джерел

1. Трихлєб А. С., Демішонкова С. А. Інформаційна комп'ютерна система контролю та управління : thesis. 2020.
URL: <https://er.knutd.edu.ua/handle/123456789/16605>
2. Advanced Security Features in Oracle HCM Cloud / Md Abul Khair, et al. *Universal Research Reports*. 2023. Vol. 10, no. 4. P. 493–511.
URL: <https://doi.org/10.36676/urr.v10.i4.1360>
3. Automating Employee Appeals Using Data-Driven Systems / Priyank Mohan et al. *International Journal for Research Publication and Seminar*. 2020. Vol. 11, no. 4. P. 390–405. URL: <https://doi.org/10.36676/jrps.v11.i4.1588>
4. Ступак В. В. Інформаційна комп'ютерна система постачальної організації : thesis. 2020. URL: <http://ir.stu.cn.ua/123456789/23534>
5. Efficiency assessment of banking systems' performance / M. Korneyev et al. *Banks and Bank Systems*. 2022. Vol. 17, no. 3. P. 72–88.
URL: [https://doi.org/10.21511/bbs.17\(3\).2022.07](https://doi.org/10.21511/bbs.17(3).2022.07)
6. Employee performance evaluation. *Effective Security Management*. 2021. P. 355–361. URL: <https://doi.org/10.1016/b978-0-12-814794-8.00036-6>
7. Баландін П. О. Комп'ютерна система потокового аудіо стрімінгу : thesis. 2021. URL: <http://ir.stu.cn.ua/123456789/23000>
8. Склярєва Д. Ю. Інформаційна комп'ютерна система контролю успішності в навчальних закладах : thesis. 2021.
URL: <http://ir.stu.cn.ua/123456789/25014>
9. Khokhlov D. O. ASSESSMENT OF WORK EFFICIENCY MANAGER. *EurasianUnionScientists*. 2019. Vol. 9, no. 63. P. 37–40.
URL: <https://doi.org/10.31618/esu.2413-9335.2019.9.63.204>
10. Ващенко О. В. Логістична інформаційно-комп'ютерна система вантажоперевізника : thesis. 2020.
URL: <http://ir.stu.cn.ua/123456789/23441>

11. Proka N. I. ASSESSMENT OF DECENT WORK AND ITS EFFICIENCY. *Bulletin of Agrarian Science*. 2023. Vol. 2, no. 101. P. 161–166. URL: <https://doi.org/10.17238/issn2587-666x.2023.2.161>
12. Ольховецька Х. А., Olkhovetska K. Комп'ютерна система контролю параметрів мікроклімату приміщення : bachelor's thesis. 2021. URL: <http://elartu.tntu.edu.ua/handle/lib/35570>
13. Software Quality. *Computer System Reliability*. 2016. P. 120–133. URL: <https://doi.org/10.1201/b14852-9>
14. Verordnung über die Berufsausbildung zum IT-System-Elektroniker und zur IT-System-Elektronikerin (IT-System-Elektroniker-Ausbildungsverordnung – ITSEAusV) : of 28.02.2020. *BGBI. I*. 2020. 5 March. URL: https://media.offenegesetze.de/bgb11/2020/bgb11_2020_9.pdf#page=32
15. Advanced Security Features in Oracle HCM Cloud / Md Abul Khair, et al. *Universal Research Reports*. 2023. Vol. 10, no. 4. P. 493–511. URL: <https://doi.org/10.36676/urr.v10.i4.1360>
16. Thangarasa T. Workday. *The COVID Journals*. 2023. P. 47–49. URL: <https://doi.org/10.1515/9781772126914-007>
17. Гасюк І., Баюк М. МОЖЛИВОСТІ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ ДЛЯ АВТОМАТИЗАЦІЇ РУТИННИХ ЗАДАЧ ТА ПРОЦЕСІВ ПУБЛІЧНОГО УПРАВЛІННЯ. *Наукові інновації та передові технології*. 2023. № 11(25). URL: [https://doi.org/10.52058/2786-5274-2023-11\(25\)-65-81](https://doi.org/10.52058/2786-5274-2023-11(25)-65-81)
18. Compare HRMS Software to Zoho People. Harrison Ford, 2022.
19. Camilleri E. Application of the Performance Management Tools. *Key Performance Indicators*. London, 2024. P. 480–524. URL: <https://doi.org/10.4324/9781032685465-14>
20. Nikitina L., Dzheniuk N., Borysova L. ЕКСПЕРТНА СИСТЕМА ДЛЯ ОЦІНКИ РИЗИКІВ ХМАРНИХ СЕРВІСІВ. *Системи управління, навігації та зв'язку. Збірник наукових праць*. 2024. Т. 1, № 75. С. 146–151. URL: <https://doi.org/10.26906/sunz.2024.1.146>

21. Key performance indicators (KPIs) / E. Sanz et al. Scipedia, 2021.
URL: <https://doi.org/10.23967/prodphd.2021.9.002>
22. Abertura de concurso documental para recrutamento de 1 posto de trabalho de Professor Auxiliar na área disciplinar de Informática, subáreas de Human-centered computing - Human computer interaction (HCI), Computing methodologies - Computer vision, Computing methodologies - Computer graphics (definidas no ACM Computing Classification System 2012) : Edital no. 445/2019. *Diário da República II Série*. 2019. 28 March.
URL: <https://dre.pt/application/conteudo/121699147>
23. Wang P. System Functional Hazard Assessment. *Civil Aircraft Electrical Power System Safety Assessment*. 2017. P. 69–99.
URL: <https://doi.org/10.1016/b978-0-08-100721-1.00004-2>
24. Camilleri E. Performance Management Concepts and Related Tools. *Key Performance Indicators*. London, 2024. P. 26–69.
URL: <https://doi.org/10.4324/9781032685465-3>
25. Panayides P. M. Key performance indicators. *Shipping Performance Management*. London, 2023. P. 43–52.
URL: <https://doi.org/10.4324/9781315717845-5>
26. Шарко А. І. Ефективність системи стимулювання працівників. *Держава та регіони. Економіка та підприємництво*. 2011. № 3. С. 166–170.
27. Designing a smart system for monitoring employees' work efficiency / D. Mehta et al. *International Journal of Student Project Reporting*. 2023. Vol. 1, no. 3. P. 283–300. URL: <https://doi.org/10.1504/ijspr.2023.132513>
28. Похиленко І. ПРАВОВЕ РЕГУЛЮВАННЯ ЗАХИСТУ ПЕРСОНАЛЬНИХ ДАНИХ. *Scientific works of National Aviation University. Series: Law Journal "Air and Space Law"*. 2023. Т. 4, № 69. С. 94–99. URL: <https://doi.org/10.18372/2307-9061.69.18322>
29. Vitlinskyi V., Slabko M. EVALUATION OF EMPLOYEES WORK EFFICIENCY AT THE ENTERPRISE. *Scientific Notes of Ostroh Academy*

- National University, "Economics" Series*. 2021. Vol. 1, no. 20(48). P. 134–140. URL: [https://doi.org/10.25264/2311-5149-2021-20\(48\)-134-140](https://doi.org/10.25264/2311-5149-2021-20(48)-134-140)
30. How Digitalization, Work-Family Balance, and Work Efficiency Can Influence Employees' Preferences for Teleworking in the Future / G. C. Dimian et al. *Engineering Economics*. 2023. Vol. 34, no. 2. P. 139–157. URL: <https://doi.org/10.5755/j01.ee.34.2.30090>
31. НОВАК М. В. ВПЛИВ ПРОФЕСІЙНОГО СТРЕСУ НА ЕФЕКТИВНІСТЬ РЕАБІЛІТАЦІЙНОЇ РОБОТИ СОЦІАЛЬНИХ ПРАЦІВНИКІВ. *Інклюзія і суспільство*. 2024. № 2. С. 95–101. URL: <https://doi.org/10.32782/2787-5137-2024-2-14>
32. Han Z. Research on the relationship between employees' core competence and work efficiency. *Journal of Education, Humanities and Social Sciences*. 2024. Vol. 29. P. 144–148. URL: <https://doi.org/10.54097/b4c4h594>
33. Camilleri E. Root Cause Analysis and Its Impact on Performance Management. *Key Performance Indicators*. London, 2024. P. 70–103. URL: <https://doi.org/10.4324/9781032685465-4>
34. IMPLEMENTATION OF THE TRAINING PROGRAM «PROFESSIONAL EDUCATION»: EXPERIENCE, PROBLEMS, PROSPECTS FOR DEVELOPMENT / L. V. Vakhidova et al. *Pedagogicheskiy Zhurnal Bashkortostana*. 2020. Vol. 88, no. 3. P. 55–67. URL: <https://doi.org/10.21510/1817-3292-2020-88-3-55-67>
35. Lueck M. GDPR in the new remote-working normal. *Computer Fraud & Security*. 2020. Vol. 2020, no. 8. P. 14–16. URL: [https://doi.org/10.1016/s1361-3723\(20\)30086-5](https://doi.org/10.1016/s1361-3723(20)30086-5)

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ
ФАКУЛЬТЕТ АВТОМАТИЗАЦІЇ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ**

**Комп'ютерна система для оцінки якості та
ефективності виконання роботи працівниками
підприємства**

Виконав: Галацан Д.М. КНм-23

Наукові керівники: д.т.н. доц. Гончаренко Т. А.
асис. Мацієвський О. О.

Київ- 2024

Актуальність дослідження

У сучасному світі управління якістю та ефективністю роботи працівників стає одним із ключових викликів для підприємств. Збільшення обсягів інформації та складності бізнес-процесів ускладнює застосування традиційних підходів до оцінки. Відсутність систематичного підходу призводить до неефективного використання ресурсів та суб'єктивності в оцінюванні.



НАУКОВА НОВИЗНА

Наукова новизна розробки комп'ютерної системи для оцінки якості та ефективності виконання роботи працівників підприємства полягає в:

1. Інтеграція сучасних технологій аналізу даних
2. Оптимізація процесу управління
3. Впровадження моделей багатокритеріального аналізу
4. Універсальність системи
5. Інноваційний підхід до візуалізації даних
6. Впровадження стандартів управління якістю

Наукове завдання

Наукове завдання цього дослідження полягає у створенні інтегрованої комп'ютерної системи для автоматизації процесів оцінки якості та ефективності виконання роботи працівників підприємства.



Мета дослідження

Метою дослідження є розробка та впровадження комп'ютерної системи для автоматизованої оцінки якості та ефективності роботи працівників підприємства.



ПРОБЛЕМАТИКА

Сучасні підприємства стикаються з низкою викликів в оцінці роботи працівників, серед яких суб'єктивність традиційних методів, складність обробки великих обсягів даних, відсутність інтеграції з корпоративними системами та недостатній рівень автоматизації. Це ускладнює об'єктивне оцінювання, уповільнює прийняття рішень і знижує ефективність управління. Необхідна розробка автоматизованої системи, яка забезпечить об'єктивність, швидкість і точність оцінки з можливістю інтеграції у бізнес-процеси.



Аналіз існуючих систем

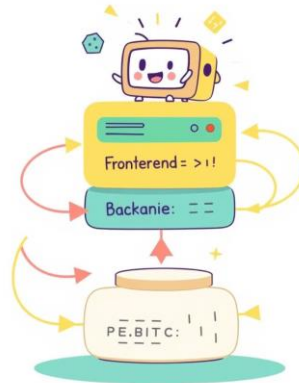
SAP SuccessFactors

Oracle HCM Cloud



Проектування комп'ютерної системи

1. Фронтенд
2. Бекенд
3. База даних



Програмне забезпечення

Реєстрація

Ім'я користувача:
Галацян Дмитро

Електронна пошта:
Galacyan.dmytro@gmail.com

Пароль:

Підтвердження пароля:

[Зареєструватися](#)

Вже маєте акаунт? [Увійти тут](#)

Авторизація

Ім'я користувача:
Галацян Дмитро

Пароль:

[Увійти](#)

Ще не маєте акаунту? [Зареєструйтесь тут](#)

Система оцінки працівників

Ім'я працівника:
Введіть ім'я

КРІ (Оцінка):
Введіть КРІ

Дата:
ДД - ММ - РРРР

[Додати працівника](#)

Ім'я	КРІ	Дата
------	-----	------

Ім'я	KPI	Дата
Галацан Д. М.	60.00	2020-08-20
Галацан Дмитро Миколайович	50.00	2021-10-21
Г. Д. М.	31.00	1996-06-06
Галацан	78.00	2121-07-07

Статистика
 Загальна кількість працівників: 4
 Середній KPI: 54.75

Меню

Дашборд

Працівники

KPI

Звіти

Дашборд

Загальна кількість працівників

4

Середній KPI

54.75

Висновок

- Ця система демонструє, як сучасні технології можуть полегшити управління персоналом та підвищити ефективність роботи. Вона підходить для компаній будь-якого масштабу та може легко адаптуватися до конкретних потреб.
- Система проектується з урахуванням можливості інтеграції з існуючими корпоративними платформами, такими як ERP, CRM, та інші інструменти управління завданнями. Це забезпечує збір і аналіз даних із різних джерел, формування звітності та створення прозорих процесів управління. Особлива увага приділяється збереженню конфіденційності даних відповідно до міжнародних стандартів, таких як GDPR.