

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ТАРАСА ШЕВЧЕНКА  
Факультет інформаційних технологій**

Кафедра технологій управління

Спеціальність 122 «Комп'ютерні науки»

Освітньо-професійна програма «Управління проектами»

**КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА**

На тему:

«Дослідження моделей управління проектом створення та впровадження автоматизованої системи управління кадровими ресурсами «iHR`s»»

**Студентки 2-го курсу групи УПз-21**

Катерини ДЬОГТЯР

**Науковий керівник:**

Кандидат технічних наук, доцент

Богдан ЄРЕМЕНКО

---

*(підпис студента)*

---

*(підпис)*

---

*(дата)*

**Попередній захист:**

---

*(Висновок: “Дозахисту в Екзаменаційній комісії”)*

Завідувач  
кафедри  
технологій  
управління

---

*(підпис)*

**Віктор МОРОЗОВ**

---

*(прізвище, ініціали)*

---

*(дата)*

**Київ 2024**

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

**Факультет інформаційних технологій**

Кафедра технологій управління

Освітній рівень магістр

Спеціальність 122 Комп'ютерні науки

Освітньо-професійна програма Управління проектами

**ЗАТВЕРДЖУЮ**

Завідувач кафедри

Професор Морозов В. В.

“\_\_” \_\_\_\_\_ 20\_\_ року

**З А В Д А Н Н Я**

**НА ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ**

Студентка: Катерина ДЬОГТЯР

Група: УПз-21

**1. Тема кваліфікаційної роботи:**

Дослідження характеристик проекту створення та впровадження системи автоматизації управління кадровими ресурсами «iHR's»

Тема затверджена на засіданні кафедри технологій управління протоколом № 13 від 28 червня 2024 року.

**2. Строк подання студентом готової роботи – “\_\_” \_\_\_\_ 2024 р.**

**3. Цільова установка та вихідні дані до роботи:**

Цільова установка проекту полягає в розробці високоефективної та зручної системи для автоматизації управління кадровими ресурсами в організації. Метою є забезпечення інтеграції основних HR-процесів, таких як управління персоналом, відпустками, заробітною платою та іншими аспектами в єдину автоматизовану систему. Вихідні дані включають аналіз потреб користувачів, вимоги до функціоналу та дослідження існуючих рішень у галузі управління персоналом.

#### 4. Зміст роботи:

Дослідження та обґрунтування доцільності створення системи автоматизації HR.

Опис концепції системи iHR's.

Етапи проектування, розробки, тестування та впровадження.

Оцінка ефективності та продуктивності після впровадження.

#### 5. Календарний план виконання роботи:

№ з/п	Назва частин роботи	План виконання роботи
1	Вивчення літературних джерел з предмету дослідження	18.09.2024 - 25.09.2024
2	Збір і вивчення матеріалів для розробки системи	26.09.2024 - 02.10.2024
3	Складання розгорнутого плану кваліфікаційної роботи	03.10.2024 – 06.10.2024
4	Ознайомлення наукового керівника з розгорнутим планом кваліфікаційної роботи. Внесення змін	09.10.2024
5	Підготовка розділу 1 «Дослідження та обґрунтування доцільності та життєздатності проекту»	10.10.2024 – 23.10.2024
6	Підготовка розділу 2 «Опис концепції системи»	24.10.2024 – 07.11.2024
7	Підготовка розділу 3 «Етап проектування системи»	08.11.2024 – 14.11.2024
8	Підготовка розділу 4 «Розробка програмної реалізації системи»	15.11.2024 – 30.11.2024
9	Оформлення кваліфікаційної роботи	01.12.2024 – 03.12.2024
10	Передача кваліфікаційної роботи рецензенту для рецензування	12.12.2024
12	Попередній захист кваліфікаційної роботи	15.12.2024

№ з/п	Назва частин роботи	План виконання роботи
13	Передача кваліфікаційної роботи на плагіат	16.12.2024
14	Захист кваліфікаційної роботи	23.12.2024

Дата видачі завдання “ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_ р.

Керівник роботи кандидат технічних наук, доцент

*Богдан ЄРЕМЕНКО*

---

*(підпис)*

## ЗМІСТ

<b>ВСТУП.....</b>	<b>9</b>
<b>РОЗДІЛ 1. ДОСЛІДЖЕННЯ ТА ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ ТА ЖИТТЄЗДАТНОСТІ ПРОЄКТУ .....</b>	<b>11</b>
1.1. Аналіз предметної галузі.....	11
1.1.1. Опис поточного стану сфери управління кадровими ресурсами.....	11
1.1.2. Огляд існуючих проблем і трендів.....	12
1.1.3. Приклади сучасних систем автоматизації HR .....	14
1.2 Характеристика індустрії та тенденції розвитку .....	15
1.2.1 Аналіз актуальних змін в HR-технологіях .....	15
1.2.2. Тенденції розвитку ІТ-інструментів для управління персоналом .....	15
1.3 Аналіз конкурентних продуктів .....	16
1.3.1. Огляд існуючих рішень для автоматизації HR .....	16
1.3.2. Аналіз конкурентів .....	18
1.4 Формулювання проблемної області .....	19
1.5 Формулювання наукової новизни та інноваційності проєкту .....	21
1.6 SWOT-аналіз та PEST-аналіз .....	23
1.6.1. SWOT-аналіз.....	23
1.6.2. PEST-аналіз.....	25
<b>РОЗДІЛ 2. ОПИС КОНЦЕПЦІЇ ПРОЄКТУ .....</b>	<b>28</b>
2.1. Визначення цілей та обсягу проєкту .....	28
2.2. Етап збору вимог .....	31
2.2.1. Ідентифікація зацікавлених сторін.....	34
2.2.2. Збір функціональних та нефункціональних вимог.....	35
2.3. Постановка задачі, формулювання технічного завдання на розробку у вигляді паспорту проєкту .....	38
2.4. Розробка концептуальної моделі системи «iHRs».....	43
2.5. Опис математичної моделі проєкту та математична постановка задачі .....	46
2.6. Аналіз та вибір методології управління проєктом .....	49
2.7. Обґрунтування вибору моделі управління для проєкту розробки iHR's .....	51
<b>РОЗДІЛ 3. ЗАСТОСУВАННЯ ТЕХНОЛОГІЙ УПРАВЛІННЯ ІТ-ПРОЄКТОМ .....</b>	<b>54</b>
3.2. Команда та кошторис проєкту .....	55

3.2.1. Організаційна структура команди проекту .....	55
3.2.2. Кошторис проекту.....	59
3.3. Управління змістом проекту. WBS і роботи проекту.....	63
3.4. Управління часом проекту. Діаграма Ганта .....	69
3.5. Управління ризиками проекту .....	71
<b>РОЗДІЛ 4. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ РЕАЛІЗАЦІЇ ІТ ПРОЄКТУ .....</b>	<b>77</b>
4.1. Вибір технологій та архітектури системи.....	78
4.2 Розробка флоу системи.....	81
4.3. Розробка концептуальної моделі бази даних проекту.....	85
4.4. Розробка логічної моделі бази даних .....	86
4.5. Опис архітектури програмного забезпечення .....	91
Загальна архітектура системи.....	92
4.5.1. Взаємодія компонентів.....	94
4.5.2. Безпека та управління доступом .....	94
4.5.3. Масштабованість та продуктивність .....	94
4.6. Розробка алгоритмів та інтерфейсів програмного забезпечення .....	98
<b>ВИСНОВОК .....</b>	<b>104</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....</b>	<b>106</b>
<b>ДОДАТОК А.....</b>	<b>109</b>

## АНОТАЦІЯ

кваліфікаційної роботи магістра на тему:

### **“Дослідження моделей управління проектом створення та впровадження автоматизованої системи управління кадровими ресурсами «iHR`s»”**

Студентка: Дьогтяр Катерина Валеріївна

Науковий керівник: Єременко Богдан Михайлович

Рік захисту — 2024

Темою даної роботи є дослідження моделей управління проектом створення та впровадження автоматизованої системи управління кадровими ресурсами «iHR`s».

*Об'єктом дослідження* є процеси створення та впровадження автоматизованої системи управління кадровими ресурсами в організації.

*Предметом дослідження* виступають моделі управління проектом розробки системи автоматизації HR-процесів, зокрема управління інтеграцією, змістом, термінами, вартістю, якістю, ресурсами, ризиками та зацікавленими сторонами.

*Мета дослідження* полягає у розробці та обґрунтуванні моделі управління проектом створення автоматизованої системи управління кадровими ресурсами, яка сприятиме підвищенню ефективності HR-процесів, забезпеченню їхньої інтеграції та відповідності сучасним вимогам бізнесу.

*Наукова новизна* роботи полягає в адаптації сучасних методологій управління проектами до специфіки автоматизації HR-процесів, розробці концептуальної моделі системи «iHR`s» та її математичній постановці, що дозволяє оптимізувати ключові HR-задачі.

Кваліфікаційна робота складається з анотації, вступу, основної частини, яка включає чотири розділи, висновків, списку використаних джерел та додатків.

*Перший розділ* присвячено аналізу предметної галузі, дослідженню сучасних проблем автоматизації HR-процесів, визначенню основних вимог до системи та аналізу методологій управління проектами.

*Другий розділ* включає розробку концептуальної моделі системи, математичне формулювання задачі, логічну модель бази даних і загальну архітектуру проекту.

*Третій розділ* зосереджений на реалізації проекту: розробці ключових модулів системи, виборі інструментів та технологій, управлінні ресурсами, ризиками, термінами та вартістю.

*Четвертий розділ* описує результати тестування системи, аналіз її ефективності, перспективи впровадження в реальних умовах та рекомендації щодо подальшого вдосконалення.

*Практична цінність* роботи полягає у створенні автоматизованої системи управління кадровими ресурсами «iHRs», яка знижує адміністративне навантаження на HR-відділ і підвищує ефективність управління персоналом. Система автоматизує облік даних співробітників, управління відпустками, розрахунок заробітної плати та генерацію звітів, що дозволяє економити час і мінімізувати помилки. Завдяки централізованій базі даних забезпечується висока точність обробки інформації та доступ до актуальних даних у режимі реального часу. Інтуїтивно зрозумілий інтерфейс спрощує взаємодію з системою для HR-фахівців і співробітників, а інтеграція з корпоративними платформами, такими як ERP і бухгалтерські системи, створює єдине інформаційне середовище компанії. Впровадження «iHRs» забезпечує відповідність сучасним вимогам захисту даних, знижує юридичні ризики, підвищує якість управлінських рішень і сприяє оптимізації бізнес-процесів.

Робота містить 108 сторінок без додатків, 31 рисунок та 5 таблиць.

**Ключові слова:** *автоматизація HR-процесів; управління персоналом; автоматизовані системи; управління проектами; інтеграція корпоративних платформ; ефективність HR-систем; управління даними співробітників; тестування системи; математична модель; аналітика персоналу.*

## ВСТУП

Управління персоналом в ІТ-компаніях має свої особливості, обумовлені специфікою цієї галузі. Сфера інформаційних технологій відрізняється динамічністю, швидкими змінами та високою конкуренцією за кваліфіковані кадри. Успішне управління персоналом потребує глибокого розуміння технічної експертизи працівників, постійного стимулювання їх професійного розвитку, створення умов для творчості та забезпечення ефективних комунікацій, зокрема в умовах віддаленої роботи.

Зважаючи на високий рівень плинності кадрів у галузі, особливого значення набуває автоматизація HR-процесів. Це дозволяє зменшити адміністративне навантаження, підвищити точність обробки даних та зосередити увагу на стратегічних аспектах управління персоналом.[27]

**Об'єктом дослідження** є процеси створення та впровадження автоматизованої системи управління кадровими ресурсами в організації. **Предметом дослідження** виступають моделі управління проектом розробки системи автоматизації HR-процесів, зокрема управління інтеграцією, змістом, термінами, вартістю, якістю, ресурсами, ризиками та зацікавленими сторонами.

**Мета дослідження** полягає у розробці та обґрунтуванні моделі управління проектом створення автоматизованої системи управління кадровими ресурсами, яка сприятиме підвищенню ефективності HR-процесів, забезпеченню їхньої інтеграції та відповідності сучасним вимогам бізнесу.

**Наукова новизна** роботи полягає в адаптації сучасних методологій управління проектами до специфіки автоматизації HR-процесів, розробці концептуальної моделі системи «iHR`s» та її математичній постановці, що дозволяє оптимізувати ключові HR-задачі.

**Практична цінність** дослідження полягає у можливості використання запропонованої системи в реальних умовах роботи компаній, що дозволить автоматизувати рутинні процеси, зменшити витрати часу на адміністративні завдання, підвищити точність аналізу даних та покращити прийняття рішень.

Для досягнення мети роботи були поставлені такі **задачі**:

1. Провести аналіз предметної галузі, сучасних тенденцій розвитку HR-технологій та існуючих рішень для автоматизації HR-процесів.
2. Розробити концептуальну та математичну моделі системи автоматизації управління кадровими ресурсами.
3. Обґрунтувати вибір методології управління проектом для створення системи «iHR`s».
4. Розробити організаційну структуру команди проекту, WBS, а також визначити ключові ризики і стратегії їхнього управління.
5. Створити концептуальну модель бази даних та інтерфейси системи.
6. Оцінити ефективність запропонованого рішення та розробити рекомендації щодо його впровадження.
7. Забезпечити відповідність системи сучасним вимогам безпеки даних

# РОЗДІЛ 1. ДОСЛІДЖЕННЯ ТА ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ ТА ЖИТТЄЗДАТНОСТІ ПРОЕКТУ

## 1.1. Аналіз предметної галузі

### 1.1.1. Опис поточного стану сфери управління кадровими ресурсами

Сфера управління кадровими ресурсами (HR) є однією з ключових функцій у діяльності будь-якої організації, незалежно від її масштабу чи сфери діяльності. У сучасному світі ефективне управління персоналом стає все важливішим, оскільки впливає на продуктивність, корпоративну культуру та конкурентоспроможність компаній.

Ключові завдання HR включають:

- Рекрутинг та адаптацію нових співробітників.
- Управління персональними даними працівників.
- Моніторинг ефективності праці та планування кар'єрного зростання.
- Організацію навчання, підвищення кваліфікації.
- Кадрове адміністрування (управління відпустками, лікарняними та іншими адміністративними процесами).
- Компенсації і пільги(премії, бонуси, заробітна плата та нематеріальна мотивація).
- Робота з корпоративною культурою.
- Оптимізація та автоматизація HR процесів [31].

Технологічний розвиток значно змінив підхід до виконання цих завдань. Сьогодні HR активно інтегрує інформаційні системи для автоматизації та підвищення ефективності своїх процесів. Це дозволяє зменшити рутинне навантаження на працівників HR-відділів, впорядковувати зберігання даних і генерування звітів, підвищити доступність інформації про співробітників і команду, зосередитися на стратегічних завданнях, таких як розвиток талантів і побудова корпоративної культури.

### 1.1.2. Огляд існуючих проблем і трендів

Незважаючи на розвиток технологій, HR-сектор ще стикається з такими проблемами:

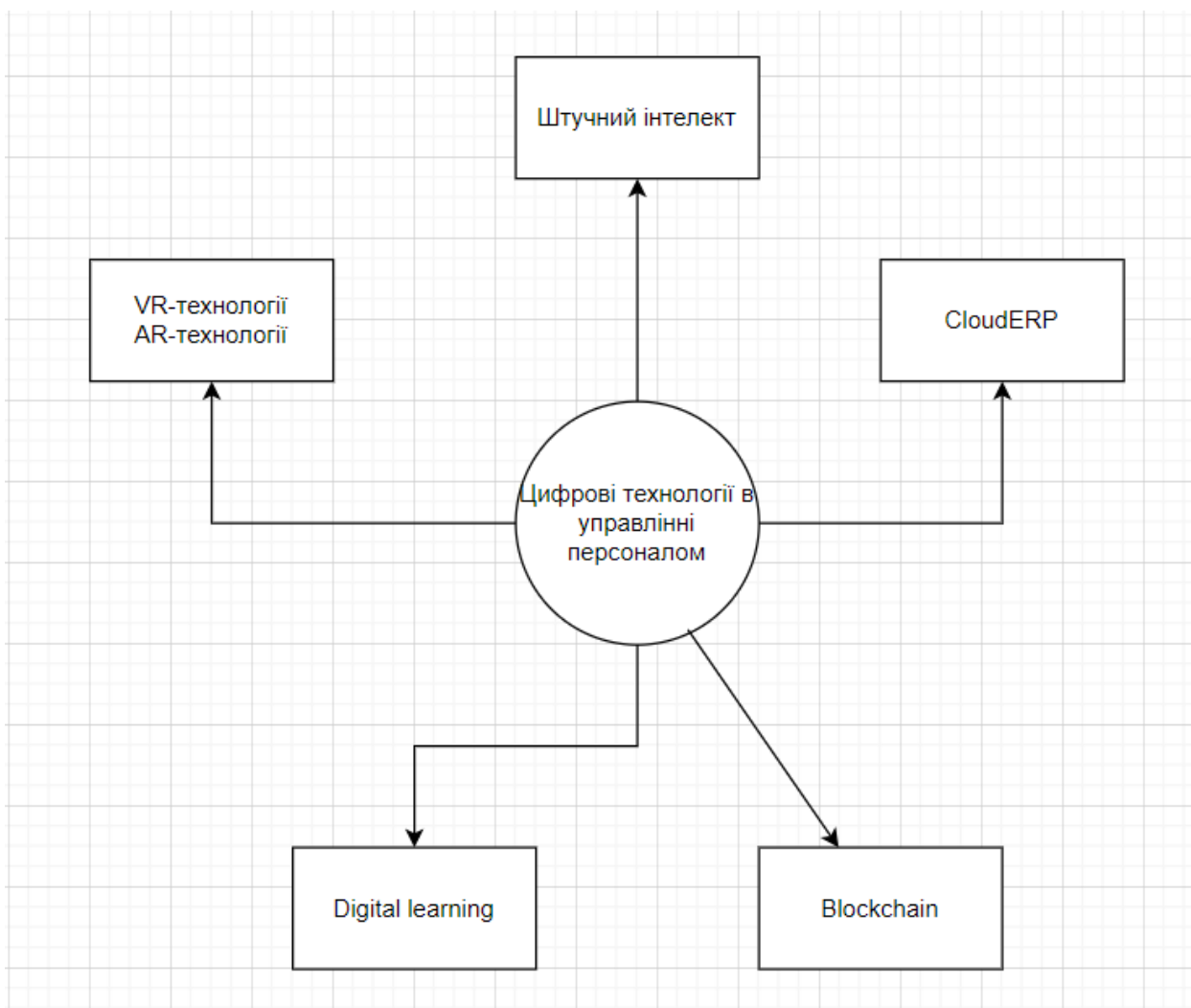
- Ручна обробка даних. У багатьох компаніях досі використовуються паперові документи чи прості таблиці Excel, що ускладнює процеси аналізу та управління даними, негативно впливає на оперативність вирішення проблем.
- Довгий шлях доступу до інформації. Відсутність централізованих баз даних призводить до затримок у прийнятті рішень, заважає економити витрати часу на підготовку, пошук необхідних документів, послаблює контроль за створенням, веденням та зберіганням документації з персоналу компанії.
- Високі адміністративні витрати. Значна частина часу HR-спеціалістів витрачається на рутинні завдання, надмірний бюрократизм під час оформлення кадрової документації.
- Складність інтеграції систем. Недосконала сумісність різних HR-інструментів ускладнює автоматизацію процесів.
- Недостатній рівень аналітики. Без відповідних інструментів складно проводити аналіз показників продуктивності та прогнозування.

*Сучасні тренди у сфері HR:*

1. Цифрова трансформація. Масштабне впровадження автоматизованих HRM-систем (HR Management Systems) для обробки даних, аналізу та управління процесами.
2. Інтеграція штучного інтелекту (ШІ). Використання AI для аналізу резюме, підбору кандидатів, автоматичного оцінювання продуктивності тощо.
3. Хмарні технології. Перехід до хмарних платформ для забезпечення доступу до даних з будь-якої точки світу.
4. Аналітика даних. Використання HR-аналітики для прогнозування плинності кадрів, оцінки ефективності навчальних програм, визначення потреб персоналу.

5. Підхід до співробітників як до клієнтів. Створення персоналізованих HR-стратегій для підвищення задоволеності працівників.
6. Мобільність. Створення мобільних додатків та чат-ботів для доступу до HR-сервісів, таких як перегляд графіку роботи, подання заявок на відпустки тощо.
7. Використання Big Data та Data mining в сфері HR [29].
8. HR-бренд. Створення привабливого HR-бренду компанії та його постійне підтримання.
9. Захист інформації. Підвищення загального рівня конфіденційності інформації та постійне посилення ступеня її захист [29].

На рис.1.1 представлена графічна схема цифрових технологій управління персоналом[32].



**Рис. 1.1. Цифрові технології управління персоналом**

### 1.1.3. Приклади сучасних систем автоматизації HR

На ринку IT представлено багато систем для автоматизації HR-процесів.

Найпопулярніші серед них:

#### 1. BambooHR

- Спеціалізується на малих і середніх компаніях.
- Пропонує інструменти для управління персоналом, відстеження часу, проведення опитувань співробітників.
- Інтегрується з іншими системами, такими як платіжні платформи.

#### 2. Workday

- Використовується великими корпораціями.
- Включає функціонал для управління персоналом, фінансами, плануванням ресурсів.
- Орієнтована на аналітику та прогнозування.

#### 3. SAP SuccessFactors

- Потужна система для управління талантами та виконання стратегічних HR-завдань.
- Пропонує модулі для рекрутингу, навчання, управління ефективністю та формування звітності.

#### 4. Zoho People

- Орієнтована на автоматизацію повсякденних HR-процесів, таких як облік робочого часу, управління документами.
- Має доступну ціну, що робить її популярною серед малих компаній.

#### 5. Oracle HCM Cloud

- Забезпечує комплексне управління кадровими процесами, включаючи аналітику, планування та управління винагородами.
- Підтримує мобільність і інтеграцію з іншими корпоративними системами.

Ці системи демонструють широкий спектр можливостей, але кожна з них потребує адаптації під специфіку компанії. Вибір потрібного інструменту залежить від розміру організації, функціоналу системи, цінової спроможності, а також стратегічних цілей.

## **1.2 Характеристика індустрії та тенденції розвитку**

### **1.2.1 Аналіз актуальних змін в HR-технологіях**

Технології впливають на сферу управління людськими ресурсами (HR) і вона зазнає швидких змін під їх впливом, що робить її більш орієнтованою на дані, автоматизацію та персоналізацію. Серед основних змін можна виділити:

Активне застосування штучного інтелекту (AI): AI використовується в різних аспектах діяльності HR від первинного відбору резюме, опрацювання заявок, розсилання листів до аналізу даних, автоматизації рекрутингових процесів, створення індивідуальних планів розвитку співробітників.

- Хмарні рішення (Cloud ERP). Використання хмарних платформ забезпечує доступ до HR-даних з будь-якого пристрою та місця, що підвищує мобільність співробітників.
- HR-аналітика. Широке застосування великих даних для оцінки продуктивності, прогнозування плинності кадрів та оптимізації процесів.
- Мобільні додатки. Зростає популярність мобільних рішень, які дозволяють співробітникам подати заявку на відпустку, перевірити графік роботи чи отримати доступ до навчальних матеріалів.
- Гейміфікація. Впровадження елементів гри для мотивації співробітників, навчання та оцінки їхньої ефективності [17, 19].
- Технології доповненої та віртуальної реальності — ці технології все більш стають популярними в програмах навчання та розвитку персоналу [27].

### **1.2.2. Тенденції розвитку IT-інструментів для управління персоналом**

IT- інструменти для управління HR процесами швидко адаптуються під ринкові зміни задля оптимізації бізнесу. Основні запити щодо їх розвитку складаються з таких аспектів:

1. Консолідація функцій в одній платформі. HR-системи стають все більш універсальними, включаючи в себе рекрутинг, управління ефективністю, оплату праці та звітність.

2. Підтримка гібридного формату роботи. Рішення для моніторингу продуктивності віддалених працівників та управління проектами стають необхідністю.
3. Автоматизація рутинних процесів. Зменшується потреба в ручній обробці даних завдяки автоматизації процесів, таких як розрахунок заробітної плати, управління відпустками тощо.
4. Фокус на добробуті співробітників. Зростає кількість інструментів для оцінки задоволеності співробітників, організації психологічної підтримки та моніторингу професійного вигорання.
5. Інтеграція Big Data з HR-Digital [30].

### **1.3 Аналіз конкурентних продуктів**

#### **1.3.1. Огляд існуючих рішень для автоматизації HR**

На сучасному ринку HRM-систем представлено велику кількість рішень, які автоматизують та оптимізують управління кадровими ресурсами. Ці платформи надають різноманітні інструменти для ефективнішої роботи HR-відділів, забезпечують надійне зберігання даних, полегшують прийняття управлінських рішень. Розглянемо кілька популярних систем:

##### **1. Workday**

Опис. Workday є однією з провідних HRM-систем, орієнтованих на великі корпорації. Основний акцент робиться на управлінні талантами, автоматизації процесів рекрутингу, нарахування зарплати та інтеграції з фінансовими даними. Система пропонує комплексні інструменти аналітики, які допомагають виявляти ключові показники ефективності.

Ключові можливості:

- Управління талантами та планування кар'єрного зростання.
- Моніторинг продуктивності співробітників.
- Мобільний доступ до HR-даних.
- Інтеграція з ERP-системами та бізнес-додатками.

Цільова аудиторія — великі корпорації та міжнародні організації, які потребують масштабованих та потужних рішень.

## **2. SAP SuccessFactors**

Опис. SAP SuccessFactors є одним із найбільш комплексних рішень для автоматизації HR-процесів. Ця система орієнтована на глобальні компанії, пропонуючи широкий спектр модулів для управління персоналом, рекрутингу, навчання та розвитку співробітників. Для кожного працівника в системі створюється акаунт, в якому він бачить свою мету і відмічає виконані завдання[28].

Ключові можливості:

- Комплексне управління рекрутингом.
- Оцінка ефективності роботи співробітників.
- Підтримка стратегічного управління персоналом через планування навчання та розвитку.
- Інструменти аналітики для прогнозування та прийняття рішень.

Цільова аудиторія — великі та середні компанії, які мають потребу у глобальних HR-рішеннях з акцентом на управління талантами.

## **3. BambooHR**

Опис. Програмне забезпечення, орієнтоване на малий та середній бізнес. Система зосереджена на простоті використання, що робить її доступною для компаній з обмеженими ресурсами. Включає в себе великий набір послуг з управління персоналом та може бути доповнене інструментарієм з обліку заробітної плати [28].

Ключові можливості:

- Облік даних працівників.
- Управління відпустками та лікарняними.
- Нарахування заробітної плати.
- Інтеграція з популярними бухгалтерськими програмами.

Цільова аудиторія — малі та середні компанії, які потребують простих і доступних HRM-рішень.

#### **4. Zoho People**

Опис. Це інформаційна система, де менеджер з персоналу може записувати важливі справи відразу в інтерфейсі, без зайвих деталей. Система надає портал співробітників, портал самообслуговування, організаційну структуру, контрольні списки для бізнес-процесів. Розроблена у веб та мобільній версії [28].

Ключові можливості:

- Ведення кадрових даних.
- Управління робочим часом.
- Відстеження ефективності співробітників.
- Інтуїтивно зрозумілий інтерфейс.

Цільова аудиторія: Малі компанії та стартапи, які потребують базового рішення для управління персоналом.

#### **1.3.2. Аналіз конкурентів**

*Сильні сторони:*

1. Універсальність і модульність. Більшість HRM-систем, таких як Workday і SAP SuccessFactors, пропонують модульну структуру, що дозволяє організаціям обирати лише ті функції, які їм потрібні. Це забезпечує адаптивність до потреб компаній різного розміру та сфери діяльності.
2. Потужна аналітика. Всі провідні системи мають вбудовані інструменти аналітики, які дозволяють HR-менеджерам приймати обґрунтовані рішення. Наприклад, Workday пропонує аналітику для оцінки продуктивності працівників, планування талантів і аналізу плинності кадрів.
3. Високий рівень безпеки даних. Усі платформи дотримуються міжнародних стандартів безпеки, таких як GDPR, для захисту персональних даних. Це забезпечує конфіденційність і безпеку інформації співробітників.

4. Інтеграція з іншими бізнес-додатками. Провідні системи, як-от SAP SuccessFactors і Workday, забезпечують інтеграцію з ERP, бухгалтерськими системами та іншими бізнес-додатками, що значно полегшує обмін даними та забезпечує зручність у використанні.

*Слабкі сторони:*

1. Висока вартість впровадження та обслуговування. Рішення, як-от SAP SuccessFactors і Workday, можуть мати значну вартість впровадження та ліцензування, що обмежує їх доступність для малих і середніх компаній. Крім того, обслуговування цих систем потребує спеціалізованої технічної підтримки.
2. Складність налаштування та тривалий період адаптації. Потужні функції та гнучкість налаштування можуть ускладнити впровадження таких систем. Процес навчання персоналу та адаптації до нових інструментів може зайняти багато часу.
3. Недостатня локалізація для окремих ринків. Багато систем розроблені для глобального ринку, але не завжди враховують специфічні вимоги локальних законодавств або культурні особливості. Це може стати проблемою для компаній, що працюють у специфічних умовах.

#### **1.4 Формулювання проблемної області**

Проблемна область, яку розглядає дана робота, полягає в потребі розробки нової автоматизованої системи для управління кадровими ресурсами «iHRs», яка забезпечить ефективне та безпечне управління даними про працівників, а також оптимізує рутинні процеси у компаніях. Незважаючи на наявність інших систем автоматизації HR-процесів, існують певні обмеження та недоліки, які можуть ускладнювати роботу HR-менеджерів та керівників. *Огляд існуючих систем виявив такі ключові проблеми:*

1. Технічні проблеми. Багато існуючих HR-систем можуть мати проблеми зі швидкодією, відсутністю підтримки актуальних версій програмного забезпечення, а також відсутністю інтеграцій з іншими корпоративними системами (наприклад, бухгалтерією або ERP). Ці обмеження можуть

призводити до затримок у обробці даних, труднощів із оновленням інформації та збоїв в роботі системи, що створює нестабільні умови для HR-менеджерів, знижуючи ефективність управління персоналом.

2. Проблеми функціональності. Існуючі системи можуть бути обмежені в своїй функціональності. Вони не завжди надають всі необхідні інструменти для комплексного управління кадровими даними, автоматизації нарахувань заробітної плати, аналізу ефективності співробітників або оптимізації процесів відпусток і лікарняних. Обмежена функціональність може призводити до того, що HR-менеджери змушені використовувати кілька систем для виконання різних завдань, що є неефективним і збільшує ймовірність помилок.
3. Проблеми зручності використання. Інтерфейси деяких HR-систем можуть бути складними для користувачів або мати незрозумілу навігацію. Недостатня зручність використання може затруднити доступ до необхідної інформації, зробити процес обробки даних занадто громіздким і призвести до помилок. Це може зайняти багато часу на навчання співробітників, а також зменшити ефективність роботи в компанії. Важливою проблемою є також відсутність персоналізації інтерфейсу під конкретні потреби користувача.
4. Проблеми безпеки. Безпека даних є критично важливим аспектом для систем управління персоналом. Існуючі HR-системи можуть не мати достатнього рівня захисту персональних даних співробітників або бути уразливими до несанкціонованого доступу. Проблеми з безпекою можуть включати недостатнє шифрування даних, незахищене зберігання паролів, слабкий контроль доступу до персональних даних або відсутність системи резервного копіювання. Це ставить під загрозу конфіденційність інформації та знижує рівень довіри до системи з боку користувачів.
5. Проблеми адаптивності до змін у законодавстві. Часті зміни в законодавстві, що стосуються трудових відносин, податків або обробки персональних даних, можуть призводити до того, що старі системи не встигають адаптуватися до нових вимог. Це створює додаткове

навантаження на HR-менеджерів і юридичний відділ компанії, а також може призвести до порушень законодавчих норм. Система «iHRs» повинна бути адаптивною до змін у законодавстві, щоб забезпечити відповідність актуальним стандартам.

#### *Вирішення проблем*

Проект «iHRs» передбачає розробку нової системи управління кадровими ресурсами, яка:

1. Покращить технічну стабільність. Система буде побудована на сучасних технологіях, що забезпечать високу швидкість, регулярне оновлення та інтеграцію з іншими корпоративними системами, такими як бухгалтерія та ERP. Це дозволить знизити технічні збої та підвищити ефективність роботи HR-менеджерів.
2. Розширить функціональність. «iHRs» забезпечить широкий спектр можливостей, включаючи управління даними співробітників, автоматичне нарахування заробітної плати, управління відпустками, аналіз продуктивності працівників та інтеграцію з іншими системами для більш ефективного керування кадровими процесами.
3. Покращить зручність використання. Система буде мати інтуїтивно зрозумілий інтерфейс з можливістю персоналізації. Всі функції будуть доступні через єдиний інтерфейс, що дозволить HR-менеджерам швидко знаходити необхідну інформацію та ефективно виконувати свої завдання.
4. Забезпечить високий рівень безпеки. «iHRs» буде мати сучасні механізми захисту даних, включаючи шифрування, багаторівневу авторизацію та регулярне оновлення системи безпеки. Це забезпечить конфіденційність і захист персональних даних співробітників[23].

### **1.5 Формулювання наукової новизни та інноваційності проекту**

Наукова новизна проекту «iHR`s» полягає у розробці інноваційної моделі управління проектом створення автоматизованої системи управління кадровими ресурсами, яка ґрунтується на сучасних підходах та вирішує актуальні проблеми автоматизації HR-процесів. Основні аспекти новизни включають:

1. **Розробка концептуальної та математичної моделі автоматизованої HR-системи.** У проєкті вперше запропоновано формалізований підхід до моделювання HR-процесів із використанням математичних функцій для оптимізації обробки даних, управління ресурсами та прогнозування ефективності кадрових рішень.
2. **Інтеграція принципів адаптивного управління.** Проєкт пропонує інноваційний підхід до управління проєктом із застосуванням гнучких методологій (Agile, Scrum), адаптованих до специфіки розробки автоматизованих систем для HR. Це дозволяє забезпечити гнучкість у реагуванні на зміни вимог та підвищити ефективність виконання проєкту.
3. **Використання технологій аналізу великих даних (Big Data) у HR-процесах.** Система «iHR`s» передбачає впровадження алгоритмів збору та аналізу великих масивів даних для оцінки продуктивності персоналу, прогнозування плинності кадрів і моделювання стратегій розвитку співробітників.
4. **Запровадження інтелектуальних механізмів прийняття рішень.** У системі передбачено використання штучного інтелекту для автоматичного аналізу даних співробітників, оцінки ефективності, формування персоналізованих рекомендацій і планів розвитку на основі зворотного зв'язку та корпоративних цілей.
5. **Інноваційна система інтеграції із зовнішніми інформаційними середовищами.** Наукова новизна проявляється у розробці механізмів інтеграції HR-системи із зовнішніми платформами навчання, сертифікації, а також корпоративними ERP- і CRM-системами, що дозволяє забезпечити єдиний інформаційний простір для управління всіма аспектами кадрових ресурсів [21].

## 1.6 SWOT-аналіз та PEST-аналіз

### 1.6.1. SWOT-аналіз

Це стратегічний інструмент для оцінки внутрішніх і зовнішніх факторів, які можуть впливати на успішність проекту, організації або конкретного рішення. Назва аналізу є аббревіатурою, яка розшифровується:

- **S (Strengths)** – Сильні сторони: внутрішні переваги проекту чи організації, які допомагають досягати цілей.
- **W (Weaknesses)** – Слабкі сторони: внутрішні недоліки, які можуть обмежувати ефективність чи результативність.
- **O (Opportunities)** – Можливості: зовнішні фактори, які організація чи проект можуть використати для свого розвитку.
- **T (Threats)** – Загрози: зовнішні ризики або обмеження, які можуть перешкодити досягненню цілей.

Таблиця 1.1

#### SWOT аналіз

Сильні сторони	Слабкі сторони
1	2
Інноваційний підхід Універсальність Масштабованість Високий рівень безпеки	Високі початкові інвестиції Потреба в навчанні Тривалість впровадження
1	2
Можливості	Загрози
Інтеграція Зростання ринку Цифровізація бізнесу Запити на локалізацію	Конкуренція Зміни в законодавстві Економічна нестабільність Технологічні виклики

#### Опис аспектів:

##### Сильні сторони

Інноваційний підхід: використання сучасних технологій (штучний інтелект, аналітика даних) для автоматизації кадрових процесів, що підвищує ефективність і точність обробки даних.

Універсальність: система адаптована для використання в організаціях різного розміру та галузей, що робить її привабливою для широкого ринку.

Масштабованість: можливість легко додавати нові модулі, інтегрувати з іншими системами та адаптувати під зростання компанії або зміну бізнес-процесів.

Високий рівень безпеки: використання сучасних механізмів захисту даних (шифрування, багаторівнева автентифікація) для забезпечення конфіденційності кадрової інформації.

### **Слабкі сторони**

Високі початкові інвестиції: розробка та впровадження системи потребує значних фінансових витрат, що може обмежити її доступність для невеликих компаній.

Потреба в навчанні: необхідність навчання персоналу для ефективного використання системи, що може збільшити витрати часу та ресурсів на впровадження.

Тривалість впровадження: процес інтеграції та налаштування системи може зайняти значний час, особливо для великих організацій з комплексними кадровими процесами.

### **Можливості**

Інтеграція: можливість інтеграції з іншими корпоративними системами (ERP, CRM, бухгалтерія), що забезпечує комплексний підхід до управління бізнес-процесами.

Зростання ринку: ринок автоматизації HR-процесів активно зростає, що створює сприятливі умови для впровадження нових рішень.

Цифровізація бізнесу: зростання популярності цифрових інструментів для управління бізнесом підвищує попит на такі рішення, як «iHRs».

Запити на локалізацію: нестача локалізованих рішень на ринку створює можливість для «iHRs» зайняти нішу, враховуючи специфіку локальних законодавств і культурних особливостей.

## Загрози

Конкуренція: наявність великих гравців (Workday, SAP SuccessFactors) з потужними ресурсами та впізнаваним брендом може створити значний конкурентний тиск.

Зміни в законодавстві: постійне оновлення трудового законодавства та регулювання обробки персональних даних може вимагати регулярних змін у системі.

Економічна нестабільність: економічні кризи можуть знизити готовність компаній інвестувати в автоматизацію, особливо серед малого та середнього бізнесу.

Технологічні виклики: швидкий розвиток технологій вимагає постійного вдосконалення системи, щоб залишатися конкурентоспроможною.

### 1.6.2. PEST-аналіз

PEST-аналіз (аналіз політичних, економічних, соціокультурних та технологічних чинників) допомагає оцінити зовнішнє середовище, в якому буде реалізовуватися проект «iHRs» (система автоматизації управління кадровими ресурсами). Основна мета аналізу — визначити фактори, які можуть вплинути на успішність і стабільність проекту, а також забезпечити коректне стратегічне планування[42]. Ось детальний огляд PEST-аналізу для проекту «iHRs»:

#### *Політичні фактори:*

- Законодавчі вимоги до обробки персональних даних: Зміни в законодавстві щодо захисту персональних даних (наприклад, вимоги GDPR або локальні закони) можуть значно вплинути на спосіб зберігання та обробки кадрової інформації в системі «iHRs». Потрібно враховувати всі юридичні аспекти, щоб забезпечити відповідність законодавству, особливо в частині обробки чутливих даних.
- Державні програми підтримки цифровізації та автоматизації бізнес-процесів: Політика урядів щодо підтримки цифрових технологій і автоматизації в бізнесі може створити сприятливі умови для розвитку проекту. Наприклад, державні ініціативи, що фінансують або стимулюють

впровадження інноваційних технологій, можуть полегшити впровадження системи у великих організаціях.

*Економічні фактори:*

- Фінансова стабільність та бюджети на автоматизацію: Зміни в економічній ситуації можуть вплинути на готовність компаній інвестувати в автоматизацію бізнес-процесів. В умовах економічних криз або нестабільності, компанії можуть скорочувати витрати, що може затримати впровадження проекту. Однак в разі стабільного економічного зростання, компанії будуть готові вкладати більше коштів у вдосконалення своїх кадрових процесів.
- Ринок праці та необхідність оптимізації кадрових ресурсів: В умовах високої конкуренції на ринку праці компанії шукають шляхи оптимізації своїх HR-процесів, щоб забезпечити ефективне управління персоналом при мінімальних витратах. Проект «iHRs» може стати конкурентною перевагою для компаній, які прагнуть автоматизувати процеси відбору, найму та управління кадрами.

*Соціокультурні фактори:*

- Зміна ставлення до цифровізації та автоматизації в HR: У багатьох компаніях HR-менеджери та працівники вже активно використовують цифрові інструменти для управління персоналом. Однак є й такі організації, які ще не готові до повної автоматизації. Це може бути викликано соціокультурними факторами, такими як недостатня довіра до технологій або бажання зберегти існуючі процеси. Проект «iHRs» повинен враховувати ці фактори і бути адаптованим до змінюваних соціокультурних умов.
- Розвиток корпоративної культури та гнучкість робочих процесів: В умовах сучасного бізнесу зростає важливість гнучкості в управлінні персоналом, в тому числі у вигляді дистанційної роботи, гнучких графіків, та адаптації під потреби співробітників. Це може стимулювати попит на інструменти, які дозволяють оперативно керувати кадровими процесами та забезпечити зручність для персоналу.

*Технологічні фактори:*

- Розвиток технологій в області HR: Постійний розвиток технологій, зокрема в області штучного інтелекту, аналітики даних та автоматизації, створює великі можливості для впровадження нових рішень. Використання сучасних технологій у «iHRs» дозволить не лише автоматизувати рутинні процеси, але й прогнозувати потреби компанії в персоналі, аналізувати ефективність роботи співробітників та покращити прийняття управлінських рішень.
- Мобільні технології та хмарні сервіси: Впровадження хмарних технологій і мобільних рішень значно підвищує доступність і зручність використання системи «iHRs». Мобільний доступ до HR-даних дозволяє співробітникам і менеджерам зручно керувати процесами з будь-якого місця і в будь-який час, що є важливим для зростання популярності та конкурентоспроможності системи на ринку.

## РОЗДІЛ 2. ОПИС КОНЦЕПЦІЇ ПРОЄКТУ

### 2.1. Визначення цілей та обсягу проєкту

Ціль проєкту є основною метою, яку необхідно досягти за допомогою реалізації проєкту. Це конкретне, чітке твердження про те, що ми плануємо досягти після завершення всіх робіт. Ціль проєкту «iHRs» полягає в розробці та впровадженні автоматизованої системи для управління кадровими ресурсами, яка дозволить автоматизувати обробку даних працівників, управління відпустками та лікарняними, нарахування заробітної плати, а також генерувати звіти для керівництва організації. Це дозволить HR-фахівцям значно зменшити час на рутинні операції та підвищити точність обробки кадрової інформації.

Обсяг проєкту визначає всі роботи, які необхідно виконати для досягнення цілі. Він включає в себе всі завдання, функції та функціональності, необхідні для успішного завершення проєкту. Визначення обсягу проєкту є важливим етапом, оскільки воно допомагає чітко зрозуміти, які роботи входять до проєкту, а які знаходяться поза його межами. Також на цьому етапі необхідно визначити ролі учасників проєкту, розподілити ресурси, скласти графік виконання робіт і розробити систему контролю якості та оцінки результатів.

*Цільові завдання проєкту «iHRs»:*

1. Розробка функціональної системи для управління кадровими ресурсами. Створення ефективної автоматизованої системи для обліку співробітників, управління відпустками та нарахуванням заробітної плати. Забезпечення системи інтуїтивно зрозумілим інтерфейсом для HR-менеджерів та співробітників.
2. Інтеграція ключових функцій для оптимізації HR-процесів. Розробка модулів для управління кадровими даними, включаючи створення профілів працівників, їх переміщення між підрозділами, зміни статусів і нарахувань. Інтеграція з іншими корпоративними системами, такими як бухгалтерія для автоматизації нарахування зарплати.
3. Розробка механізмів для збору та генерації звітів. Створення можливості автоматичної генерації звітів про працівників, витрати на персонал,

- плинність кадрів. Налаштування функцій для надання звітів в різних форматах (Excel, PDF).
4. Аналіз та впровадження вимог до безпеки та конфіденційності даних. Розробка заходів для захисту персональних даних співробітників та забезпечення їх збереження відповідно до чинних стандартів безпеки. Впровадження шифрування та системи контролю доступу до чутливих даних.
  5. Оцінка ризиків та розробка стратегій управління для забезпечення стабільності та надійності системи. Проведення аналізу ризиків щодо стабільності роботи системи та безпеки даних. Розробка планів на випадок відмови або збоїв у системі, забезпечення її стійкості.
  6. Розробка стратегії підтримки користувачів та створення каналів комунікації для вирішення їхніх питань.
  7. Налагодження ефективної служби підтримки для HR-менеджерів та співробітників. Забезпечення зручних каналів комунікації (електронна пошта, чат, телефон) для отримання технічної підтримки.
  8. Вивчення та врахування законодавчого середовища для забезпечення відповідності вимогам законодавства. Аналіз та інтеграція змін у законодавстві, що стосуються обробки персональних даних, нарахування заробітної плати та інших кадрових процесів. Адаптація системи до нових норм і стандартів, що можуть з'являтися в межах правового поля.
  9. Забезпечення масштабованості системи для підтримки зростання користувачів та організацій. Визначення архітектури системи, яка дозволяє масштабувати її під потреби великих компаній. Впровадження ефективної роботи системи на великій кількості користувачів та даних.
  10. Розробка маркетингової стратегії для просування системи на ринку. Створення стратегії для просування системи серед потенційних клієнтів, включаючи HR-агентства, консалтингові компанії, великі корпорації. Розробка матеріалів для презентації та демонстрації можливостей системи.
  11. Впровадження моніторингових засобів для стеження за ефективністю системи. Розробка інструментів для моніторингу стану системи, що

дозволяють оперативно реагувати на збої та покращувати її продуктивність.

*Обсяг проекту:*

Обсяг проекту «iHRs» включає в себе розробку та впровадження повноцінної автоматизованої системи для управління кадровими ресурсами в компанії. Ключові етапи проекту:

1. Планування проекту:

- Збір вимог і визначення основних функцій системи.
- Ідентифікація зацікавлених сторін і визначення їхніх очікувань.
- Формування команди проекту та вибір керівника.
- Установлення комунікаційних процедур та ресурсів для реалізації проекту.

2. Проектування та розробка системи:

- Визначення технічних рішень, включаючи вибір мов програмування, бази даних та архітектури системи.
- Розробка інтерфейсу користувача, проектування функціональних модулів для управління кадровими даними та нарахуванням зарплати.

3. Реалізація ключових модулів:

- Розробка модуля для автоматизації обліку та обробки відпусток.
- Створення системи для генерації звітів і аналітики.
- Впровадження механізмів захисту даних і безпеки.

4. Тестування та виправлення помилок:

- Проведення модульних, інтеграційних та системних тестів для перевірки всіх функціональних можливостей.
- Оптимізація продуктивності та усунення виявлених помилок.

5. Впровадження та підтримка:

- Розгортання системи на платформі організації.
- Організація постійної технічної підтримки, оновлень і вдосконалень системи на основі зворотного зв'язку від користувачів.

На рис.2.1 показано дерево цілей проєкту.

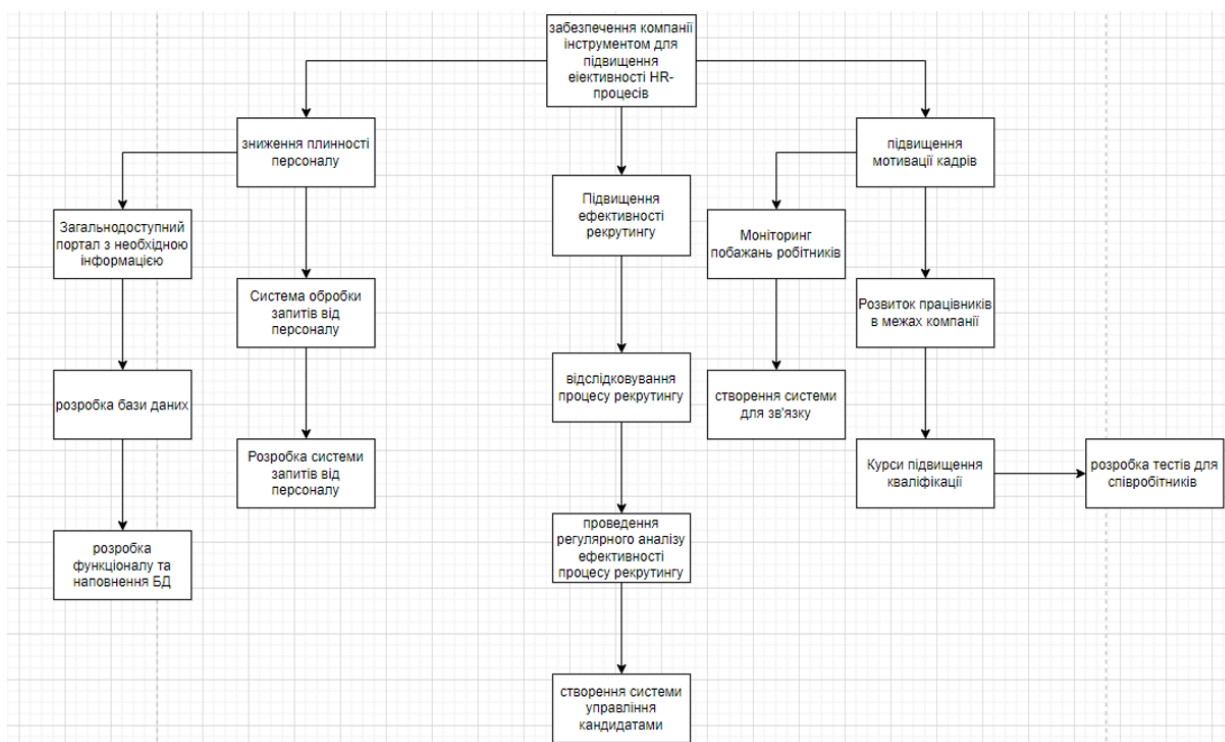


Рис. 2.1. Дерево цілей проєкту

## 2.2. Етап збору вимог

Етап збору вимог є критично важливим для проєкту «iHRs» (система автоматизації управління кадровими ресурсами). На цьому етапі здійснюється детальний аналіз потреб користувачів і визначення їхніх очікувань щодо функціональності та характеристик системи. Збір вимог дозволяє створити чітке уявлення про те, як система має функціонувати та яких результатів повинна досягти. Це забезпечить розробникам необхідні дані для створення системи, яка повністю відповідає вимогам замовника і кінцевих користувачів.

Збір вимог включає в себе консультації з усіма зацікавленими сторонами, аналіз наявних процесів і документації, а також проведення інтерв'ю та опитувань. Цей етап має важливе значення для уникнення непорозумінь у подальшому етапі розробки та забезпечення ефективності майбутньої системи.

### 1. Функціональні вимоги

Описують конкретні функції та операції, які система повинна виконувати. Вони визначають, які задачі система має виконувати для задоволення потреб користувачів.

Включають у себе такі функції, як реєстрація співробітників, управління відпустками, нарахування заробітної плати, створення звітів, а також інтеграція з іншими корпоративними системами.

Описують, як система повинна взаємодіяти з користувачем, включаючи інтерфейс, а також взаємодіяти з іншими системами в межах організації (наприклад, бухгалтерія, система управління проектами).

*Приклад:*

Реалізація функції автоматичного нарахування заробітної плати на основі оброблених даних про робочий час і премії.

Створення модуля для обробки заявок на відпустки, автоматичне розрахування залишку відпусток співробітників.

## **2. Нефункціональні вимоги**

Визначають характеристики системи, що не пов'язані з її основною функціональністю, але є важливими для її ефективної роботи.

Включають вимоги до продуктивності, безпеки, доступності, надійності та сумісності.

Можуть визначати обмеження щодо технічних характеристик системи, таких як час відгуку, максимальне навантаження, потреби в ресурсах тощо.

*Приклад:*

Вимоги до часу обробки запитів (наприклад, нарахування зарплати має бути завершено за 10 хвилин).

Вимоги до безпеки даних: всі персональні дані співробітників повинні бути зашифровані і зберігатися відповідно до стандартів конфіденційності.

## **3. Бізнес-вимоги**

Визначають, як система має допомогти організації досягти стратегічних цілей, покращити ефективність управління персоналом і знизити витрати.

Включають економічні аспекти, такі як витрати на впровадження системи, економія часу та ресурсів завдяки автоматизації процесів.

Визначають очікувану прибутковість від використання системи та її вплив на зростання продуктивності працівників і керівників.

*Приклад:*

Зменшення часу на обробку документації та звітності.

Скорочення витрат на адміністративні послуги та HR-процеси завдяки автоматизації.

### **1. Зовнішні вимоги**

Пов'язані з зовнішніми факторами, такими як законодавчі норми, регулювання персональних даних, стандарти галузі та екологічні обмеження.

Враховують вимоги до сумісності системи з іншими зовнішніми програмними продуктами або стандартами, які використовуються у галузі HR.

*Приклад:*

Дотримання вимог GDPR для захисту персональних даних співробітників.

Інтеграція з державними реєстрами або іншими зовнішніми базами даних для автоматизації процесів обліку співробітників.

### **2. Операційні вимоги**

Операційні вимоги визначають, як система повинна функціонувати в реальних умовах. Включають вимоги до продуктивності, відновлюваності, обслуговування та інших аспектів, які забезпечують успішну експлуатацію системи. Визначають, як система повинна реагувати на непередбачувані ситуації, такі як відмова компонентів або потреба в оновленнях.

*Приклад:*

Система повинна мати механізми для відновлення даних після збоїв (наприклад, відновлення інформації про відпустки співробітників після відмови бази даних).

Підтримка регулярних оновлень і обслуговування для збереження сумісності з новими версіями операційних систем та програмного забезпечення.

Процес збору вимог. Збір вимог в проєкті «iHRs» включає кілька етапів:

1. Консультації з зацікавленими сторонами — зустрічі з HR-менеджерами, співробітниками IT-відділу, керівниками відділів, щоб зрозуміти їхні потреби та очікування.
2. Аналіз існуючих бізнес-процесів — оцінка поточних HR-процесів та їх автоматизації.
3. Інтерв'ю та опитування — збір додаткових даних за допомогою інтерв'ю з кінцевими користувачами для виявлення їхніх вимог до системи.
4. Документування вимог — усі вимоги фіксуються в документації, яка потім узгоджується з усіма учасниками проекту.

### **2.2.1. Ідентифікація зацікавлених сторін**

У проекті «iHRs» (система автоматизації управління кадровими ресурсами) зацікавлені сторони — це особи, групи або організації, які мають прямий інтерес у реалізації проекту або можуть вплинути на його результати. Розуміння потреб і очікувань цих сторін є важливим для успішного виконання проекту, оскільки дозволяє врахувати різноманітні інтереси та максимізувати підтримку всіх учасників. Нижче наведені основні зацікавлені сторони для цього проекту:

#### **1. Замовники (Клієнти)**

Організації або особи, які ініціюють і фінансують проект. Очікування: висока функціональність системи, швидка інтеграція, відповідність бюджету й термінів, легкість навчання користувачів.

#### **2. Кінцеві користувачі (HR-менеджери та співробітники)**

Основні користувачі системи для управління персоналом та взаємодії з даними. Очікування: інтуїтивний інтерфейс, швидкий доступ до функцій, відсутність помилок у роботі системи.

#### **3. Проектний менеджер**

Відповідає за координацію виконання проекту в межах бюджету і термінів. Очікування: організація роботи команди, дотримання графіка, якісне виконання завдань.

#### **4. Розробники**

Інженери, які реалізують технічні аспекти проекту.  
Очікування: чіткі вимоги, технічні ресурси, зворотний зв'язок користувачів.

#### **5. Тестувальники**

Спеціалісти із забезпечення якості програмного продукту.  
Очікування: виявлення помилок, тестування функцій, стабільна робота системи після запуску.

#### **6. Фінансові аналітики**

Відповідають за фінансову оцінку проекту та контроль витрат.  
Очікування: дотримання бюджету, аналіз економічної ефективності.

#### **7. Керівництво організації**

Високий рівень управління, зацікавлений у підвищенні ефективності персоналу.  
Очікування: зниження витрат, автоматизація процесів, дотримання стандартів обробки даних.

### **2.2.2. Збір функціональних та нефункціональних вимог**

Під час зустрічей з зацікавленими сторонами проекту «iHRs» (автоматизована система управління кадровими ресурсами) проводився ретельний аналіз та обговорення функціональних і нефункціональних вимог до майбутньої системи. Обговорення на цих зустрічах допомогло виявити ключові аспекти, які необхідно врахувати під час розробки, щоб система задовольняла потреби користувачів і була ефективною в управлінні кадровими процесами.

#### **Функціональні вимоги:**

##### *1. Реєстрація та авторизація:*

Забезпечення реєстрації нових користувачів, що включає введення базових даних (ім'я, прізвище, посада, підрозділ).

Впровадження системи авторизації з двофакторною аутентифікацією для забезпечення безпеки доступу до системи.

##### *2. Управління кадровими даними:*

Можливість зберігання та редагування особистих даних співробітників, включаючи історію працевлаштування, статуси відпусток, нарахування заробітної плати та бонусів.

Функція додавання нових працівників в систему через інтерфейс HR-менеджера.

### *3. Управління відпустками та лікарняними:*

Автоматизація процесу обробки заявок на відпустки та лікарняні для співробітників.

Перегляд залишків відпусток для кожного працівника з можливістю їх коригування.

### *4. Нарахування заробітної плати та бонусів:*

Автоматизований розрахунок заробітної плати, бонусів та премій на основі визначених критеріїв (відпрацьований час, продуктивність).

Генерація платіжних відомостей та інтеграція з бухгалтерськими системами для виплати заробітної плати.

### *5. Збір та генерація звітів:*

Забезпечення системи для формування різних звітів: звіт про кількість працівників, витрати на персонал, ефективність працівників.

Можливість експорту звітів у формати PDF та Excel.

### *6. Інтерфейс для співробітників:*

Співробітники повинні мати доступ до своїх особистих даних, залишку відпусток, нарахованої заробітної плати та іншої інформації.

Можливість подачі заявок на відпустку або лікарняний через особистий кабінет.

## **Нефункціональні вимоги:**

### *1. Безпека та конфіденційність:*

Всі особисті дані співробітників повинні бути захищені за допомогою шифрування.

Впровадження багаторівневої авторизації для забезпечення захисту даних і доступу до різних функцій системи.

Дотримання стандартів безпеки даних (GDPR, локальні закони щодо захисту персональних даних).

## *2. Швидкодія та висока доступність:*

Система повинна мати високу швидкодію для обробки великих обсягів даних (дані про тисячі співробітників).

Забезпечення безперебійної роботи системи з 99,9% доступністю, що дозволяє користувачам працювати без збоїв.

## *3. Масштабованість:*

Система повинна бути здатною обробляти значний обсяг даних і масштабуватись у разі зростання кількості користувачів або обсягів даних.

Можливість додавання нових підрозділів або філій до вже існуючої структури.

## *4. Крос-платформенність:*

Система повинна підтримувати доступ як через веб-інтерфейс, так і через мобільні пристрої для співробітників і HR-менеджерів.

Забезпечення роботи на основних операційних системах (Windows, macOS, Android, iOS).

## *5. Інтуїтивний інтерфейс:*

Розробка зручного і зрозумілого інтерфейсу для користувачів різного рівня кваліфікації (HR-менеджери, співробітники).

Можливість налаштування інтерфейсу для різних ролей користувачів (адміністратор, HR-менеджер, працівник).

## *6. Інтеграція з іншими системами:*

Інтеграція з іншими корпоративними системами, такими як бухгалтерія та ERP, для забезпечення безперебійного обміну даними.

Можливість інтеграції з платіжними системами для автоматизації процесу виплат заробітної плати.

### **2.3. Постановка задачі, формулювання технічного завдання на розробку у вигляді паспорту проєкту**

Задача проєкту полягає в розробці автоматизованої системи управління кадровими ресурсами «iHRs», яка надасть компаніям ефективний інструмент для автоматизації обліку персоналу, управління відпустками, нарахуванням заробітної плати, а також для збору і генерації звітності. Система повинна забезпечити користувачів зручним інтерфейсом для роботи з кадровими даними, а також забезпечувати безпеку та конфіденційність інформації.

#### **Основні функціональні блоки системи:**

1. Реєстрація та авторизація користувачів
  - Механізм реєстрації нових користувачів, що включає заповнення обов'язкових полів (ім'я, прізвище, контактні дані, роль в компанії).
  - Перевірка правильності введених даних та автоматичне створення профілю в базі даних.
  - Система авторизації, яка передбачатиме перевірку логіну та пароля з використанням захисту даних (шифрування паролів і двофакторна аутентифікація).
2. Управління кадровими даними
  - Збір та зберігання даних про працівників (особисті дані, трудова історія, відпустки, підвищення, оцінки ефективності).
  - Механізм редагування даних, який дозволяє змінювати інформацію про працівника за потребою, в тому числі через інтерфейс адміністратора.
3. Облік відпусток та лікарняних
  - Інтерфейс для HR-менеджерів для управління заявками на відпустки і лікарняні.
  - Автоматизація розрахунку залишку відпусток на основі даних про працівника.
  - Система моніторингу відпусток, що дозволяє HR-менеджерам легко відслідковувати статус запитів на відпустки.
4. Зарплата та нарахування

- Автоматичне нарахування заробітної плати на основі відпрацьованого часу, бонусів, премій, вирахувань (податки, страхування).
- Генерація платіжних відомостей, які можна експортувати в різні формати (PDF, Excel) для подальшого використання або передачі.

#### 5. Збір статистики та звітність

- Інтеграція з іншими системами (наприклад, для збору статистики з бухгалтерії або фінансів).
- Генерація звітів для аналізу плинності кадрів, витрат на персонал, ефективності працівників тощо.
- Функція експорту звітів в форматах Excel, PDF для подальшого використання або надання керівництву.

#### 6. Безпека та захист даних

- Шифрування та захист персональних даних працівників, застосування механізмів автентифікації.
- Контроль доступу до різних функцій системи в залежності від ролі користувача (адміністратор, HR-фахівець, звичайний працівник).
- Заходи з протидії шахрайству, витоку даних та іншим видам кіберзлочинності.

#### 7. Інтерфейс користувача та взаємодія

- Зручний інтерфейс для користувачів (HR-спеціалістів та адміністративного персоналу), що дозволяє ефективно орієнтуватися в системі, переглядати та редагувати інформацію.
- Інтуїтивно зрозумілий інтерфейс для співробітників для перегляду особистих даних, відпусток, заробітної плати.

Технічне завдання для проекту iHRs включає в себе наступні вимоги до системи:

#### **1. Функціональні вимоги:**

- Реалізація всіх зазначених вище функцій з можливістю їх масштабування в залежності від розміру компанії.
- Інтерфейс, адаптований для роботи на різних пристроях (ПК, мобільних телефонах).

- Використання сучасних баз даних для зберігання і обробки даних про співробітників (PostgreSQL або MySQL).
- Інтеграція з іншими корпоративними системами для обміну даними (наприклад, бухгалтерія, управління фінансами).

## **2. Нефункціональні вимоги:**

- Безпека — використання сучасних методів шифрування для зберігання паролів та персональних даних.
- Продуктивність — швидкість обробки запитів до бази даних не повинна перевищувати 1 секунду.
- Масштабованість — система повинна підтримувати одночасну роботу понад 1000 користувачів без зниження продуктивності.
- Кросплатформність — система повинна працювати на основних платформах (Windows, Linux, MacOS).

## **3. Технічні обмеження:**

- Система повинна бути сумісною з популярними веб-браузерами (Google Chrome, Mozilla Firefox, Microsoft Edge).
- Інтерфейс повинен бути адаптований для роботи на мобільних пристроях з операційною системою Android та iOS.
- Інтеграція з іншими корпоративними системами (бухгалтерія, фінансова система).

## **4. Інтеграція з іншими системами:**

- Інтеграція з платіжними системами для автоматичного нарахування заробітної плати.
- Можливість обміну даними з іншими корпоративними програмами для автоматизації процесів.

У таблиці 2.1. показано ключові аспекти проекту у вигляді паспорту проекту. Вона є узагальненням і деталізацією попереднього тексту, що описує постановку задачі та технічне завдання.

## Паспорт проєкту

Пункт	Опис
Назва проєкту	Дослідження моделей управління проектом створення і впровадження автоматизованої системи управління кадровими ресурсами «iHRs».
Мета проєкту	Розробка та впровадження автоматизованої системи управління кадровими ресурсами «iHRs», яка оптимізує процеси обліку, моніторингу та аналізу кадрових даних у компанії.
Основні задачі проєкту	<ul style="list-style-type: none"> <li>- Аналіз потреб користувачів і визначення вимог до системи.</li> <li>- Розробка концептуальної моделі системи.</li> <li>- Вибір технологій для розробки програмного забезпечення.</li> <li>- Розробка інтерфейсу користувача та основних функцій.</li> <li>- Інтеграція з іншими корпоративними системами.</li> <li>- Розробка механізмів безпеки і захисту даних.</li> <li>- Тестування і налагодження системи.</li> <li>- Впровадження і супровід системи після запуску.</li> </ul>
Очікувані результати	<ul style="list-style-type: none"> <li>- Створення стабільної і безпечної системи управління кадровими ресурсами для компаній.</li> <li>- Впровадження автоматизації в обробці кадрових даних.</li> <li>- Зручний інтерфейс для HR-фахівців.</li> <li>- Інтеграція з існуючими корпоративними системами.</li> <li>- Документація для користувачів і технічний опис системи.</li> <li>- Технічна підтримка після запуску.</li> </ul>

Обмеження проекту	<ul style="list-style-type: none"> <li>- Бюджет проекту обмежений певною сумою.</li> <li>- Необхідно забезпечити сумісність системи з існуючою інфраструктурою компанії.</li> <li>- Проект має бути завершений за 6 місяців.</li> </ul>
Ключові зацікавлені сторони	<ul style="list-style-type: none"> <li>- Замовники (керівництво компанії).</li> <li>- HR-фахівці (кінцеві користувачі системи).</li> <li>- IT-відділ (підтримка та інтеграція).</li> <li>- Постачальники технологій.</li> <li>- Команда проекту (розробники, аналітики, тестувальники).</li> </ul>
Основні етапи проекту	<ol style="list-style-type: none"> <li><b>1. Аналіз вимог та розробка концептуальної моделі (1 місяць).</b></li> <li><b>2. Розробка інтерфейсу та технічного дизайну (1 місяць).</b></li> <li><b>3. Розробка і тестування основних функцій (2 місяці).</b></li> <li><b>4. Інтеграція системи з корпоративними рішеннями (1 місяць).</b></li> <li><b>5. Впровадження та супровід системи (1 місяць).</b></li> </ol>
Ризики	<ul style="list-style-type: none"> <li>- Технічні ризики (несумісність із іншими системами, технологічні обмеження).</li> <li>- Фінансові ризики (обмеження бюджету).</li> <li>- Зміни в законодавстві (регулювання обробки кадрових даних).</li> <li>- Конкуренція (нові рішення на ринку).</li> <li>- Висока залежність від кінцевих користувачів у процесі тестування та налаштування системи.</li> </ul>

Основні метрики успіху	<ul style="list-style-type: none"> <li>- Виконання всіх функцій згідно з вимогами замовника.</li> <li>- Тривалість тестування та коригування після запуску.</li> <li>- Кількість активних користувачів системи.</li> <li>- Задоволеність кінцевих користувачів (HR-фахівців).</li> <li>- Скорочення часу на обробку кадрових даних.</li> <li>- Підвищення ефективності кадрових процесів.</li> </ul>
------------------------	--

#### 2.4. Розробка концептуальної моделі системи «iHRs»

Концептуальна модель системи «iHRs» визначає основні компоненти, їх взаємодію та функціональні можливості. Вона слугує основою для проектування архітектури системи, забезпечуючи задоволення потреб HR-фахівців і компаній, які прагнуть автоматизувати управління кадровими ресурсами. Модель орієнтована на оптимізацію HR-процесів, забезпечення точності даних та підвищення ефективності роботи.

##### Основні компоненти системи «iHRs»:

#### 1. Аутентифікація та реєстрація

- Модуль забезпечує створення облікових записів для користувачів (HR-фахівців, менеджерів і співробітників) та захищений вхід у систему.
- Функції: двофакторна автентифікація, шифрування паролів, контроль рівнів доступу до функціональності залежно від ролей.

#### 2. Головна панель управління

- Інтерактивний інтерфейс для відображення ключової інформації про стан HR-процесів.

- Функції: перегляд основних метрик (кількість співробітників, статус відпусток, звітів), швидкий доступ до ключових модулів.

### **3. Управління профілями співробітників**

- Модуль, що дозволяє вести облік усіх даних про працівників: контактні дані, посада, історія роботи, статус відпусток і лікарняних.
- Функції: додавання та редагування профілів, зміна статусів працівників, зберігання документації (контракти, довідки).

### **4. Управління відпустками та лікарняними**

- Модуль для подання заявок на відпустки або лікарняні, а також автоматичного їх затвердження або відхилення відповідно до політики компанії.
- Функції: автоматичний розрахунок залишків відпусток, сповіщення про затвердження, інтеграція з календарем компанії.

### **5. Управління заробітною платою**

- Модуль для розрахунку зарплати, премій і нарахувань. Інтегрується з бухгалтерськими системами для автоматизації фінансових операцій.
- Функції: створення розрахункових відомостей, автоматичне врахування податків, премій, відпусток та інших факторів.

### **6. Аналітика та звіти**

- Модуль для створення звітів про HR-процеси: продуктивність працівників, плинність кадрів, витрати на персонал тощо.
- Функції: генерація графіків, діаграм, інтеграція з BI-системами для глибокої аналітики

### **7. Оповіщення та сповіщення**

- Модуль для сповіщення користувачів про важливі події, наприклад, затвердження відпустки, нові запити або нагадування про закінчення контракту.
- Функції: персоналізовані сповіщення, інтеграція з електронною поштою та месенджерами.



## 2.5. Опис математичної моделі проєкту та математична постановка задачі

Проєкт iHR's спрямований на автоматизацію процесів управління людськими ресурсами в компаніях. Задача розробки цього додатку може бути сформульована за допомогою математичних моделей і постановки задачі, зокрема для оптимізації управлінських процесів, таких як найм, облік відпусток, нарахування заробітної плати та звітність.

Математична модель:

Нехай у нас є певна кількість працівників в організації, які працюють за різними контрактами та завданнями. Позначимо кількість працівників як  $N$ , кількість типів контрактів (наприклад, тимчасові, постійні, контрактні) як  $M$ , а кількість доступних завдань або ролей як  $PPP$ .

Для кожного працівника  $i$  (де  $1 \leq i \leq N$ ) і кожного типу  $j$  (де  $1 \leq j \leq M$ ) введемо змінну  $x_{ij}$ , яка вказує, чи працівник  $i$  має контракт типу  $j$ .  $x_{ij} = 1$ , то працівник  $i$  має контракт типу  $j$ ; якщо  $x_{ij} = 0$ , то працівник не має такого контракту.

Аналогічно, для кожного працівника  $i$  і кожного завдання  $k$  (де  $1 \leq k \leq P$ ) введемо змінну  $y_{ik}$ , яка вказує, чи працівник  $i$  призначений на завдання  $k$ . Якщо  $y_{ik} = 1$ , то працівник  $i$  виконує завдання  $k$ ; якщо  $y_{ik} = 0$ , то працівник не виконує це завдання.

Цільова функція:

Основною метою проєкту є оптимізація управління людськими ресурсами. Це включає в себе максимізацію продуктивності роботи організації, враховуючи найм працівників, розподіл завдань і типи контрактів. Можна сформулювати цільову функцію для максимізації загальної продуктивності:

$$\text{Max } Z = \sum_{i=1}^N \sum_{j=1}^M p_{ij} \cdot x_{ij} + \sum_{i=1}^N \sum_{k=1}^P q_{ik} \cdot y_{ik} \quad (2.1)$$

де:

- $p_{ij}$  — це показник продуктивності працівника  $i$ , який має контракт типу  $j$ ;
- $q_{ik}$  — це показник продуктивності працівника  $i$ , який виконує завдання  $k$ .

Цільова функція визначає максимізацію загальної продуктивності, яка залежить від типів контрактів та виконаних завдань для кожного працівника.

Обмеження:

Під час розробки додатку для iHR's потрібно врахувати низку обмежень.

Ось кілька з них:

1. Обмеження на кількість доступних ресурсів (робочих годин): Для кожного працівника  $i$  та для кожного завдання  $k$ , кількість годин, які він має працювати, повинна бути обмежена. Позначимо кількість годин, необхідних для виконання завдання  $k$  працівником  $i$ , як  $h_{ik}$ . Загальна кількість годин не повинна перевищувати доступний ліміт  $H_i$  для працівника  $i$ :

$$\sum_{k=1}^P y_{ik} \cdot h_{ik} \leq H_i, \quad \forall i(1 \leq i \leq N) \quad (2.2)$$

2. Обмеження на кількість працівників на завданні: Для кожного завдання  $k$  кількість працівників, які можуть бути призначені на це завдання, обмежена. Позначимо максимальну кількість працівників, які можуть бути призначені на завдання  $k$ , як  $L_k$ . Тому:

$$\sum_{i=1}^N y_{ik} \leq L_k, \quad \forall k(1 \leq k \leq P) \quad (2.3)$$

3. Обмеження на ризик (зміна умов контрактів): Якщо організація може змінювати умови контрактів для певних працівників, необхідно врахувати ризик, пов'язаний з такими змінами. Позначимо ризик зміни умов контракту для працівника  $i$  на контракт типу  $j$  як  $r_{ij}$ . Загальний ризик для всіх працівників не повинен перевищувати певний ліміт  $R_{max}$ :

$$\sum_{i=1}^N \sum_{j=1}^M x_{ij} \cdot r_{ij} \leq R_{max}$$

(2.4)

4. Обмеження на максимальну кількість типів контрактів: Якщо організація має обмеження на максимальну кількість певних типів контрактів, то ми можемо ввести обмеження на кількість працівників з конкретним типом контракту. Позначимо ліміт для контракту типу  $j$  як  $C_j$ :

$$\sum_{i=1}^N x_{ij} \leq C_j, \quad \forall j(1 \leq j \leq M) \quad (2.5)$$

Математична постановка задачі:

Задача полягає в тому, щоб вибрати оптимальну комбінацію типів контрактів та завдань для кожного працівника, яка максимізує загальну продуктивність організації, враховуючи обмеження на кількість годин, кількість працівників на завданнях, ризики та ліміти на типи контрактів.

Математична постановка задачі виглядає наступним чином:

$$\text{Max } Z = \sum_{i=1}^N \sum_{j=1}^M p_{ij} \cdot x_{ij} + \sum_{i=1}^N \sum_{k=1}^P q_{ik} \cdot y_{ik} \quad (2.6)$$

за умови, що:

$$\sum_{k=1}^P y_{ik} \cdot h_{ik} \leq H_i, \quad \forall i(1 \leq i \leq N) \quad (2.7)$$

$$\sum_{i=1}^N y_{ik} \leq L_k, \quad \forall k(1 \leq k \leq P)$$

$$\sum_{i=1}^N \sum_{j=1}^M x_{ij} \cdot r_{ij} \leq R_{max}$$

$$\sum_{i=1}^N x_{ij} \leq C_j, \quad \forall j(1 \leq j \leq M)$$

Це дозволяє оптимізувати розподіл ресурсів і завдань в організації, забезпечуючи максимальний прибуток при врахуванні обмежень на ресурси та ризику.

## **2.6. Аналіз та вибір методології управління проектом**

Управління проектами є ключовим етапом у досягненні успіху в розробці програмного забезпечення. Для проекту iHR's, який створюється для автоматизації процесів управління людськими ресурсами, важливо вибрати таку методологію управління проектом, яка буде відповідати специфіці задач, забезпечить ефективну взаємодію з користувачами, дозволить швидко реагувати на зміни та оптимізує процеси розробки. Ось кілька методологій, які можна застосувати в рамках цього проекту:

### *1. Каскадна модель (Waterfall):*

- *Опис.* Це традиційна лінійна модель управління проектами, в якій кожен етап розробки починається тільки після завершення попереднього. Проект розбивається на чіткі фази, такі як вимоги, дизайн, реалізація, тестування, запуск та обслуговування.
- *Переваги для iHR's.* Структурованість і чіткість етапів може бути корисною на початкових етапах проекту для визначення вимог та створення технічного дизайну системи.
- *Недоліки для iHR's.* Ця модель менш гнучка, що може ускладнити впровадження змін під час розробки. Оскільки проект iHR's передбачає змінні вимоги та необхідність постійної взаємодії з користувачами, каскадна модель може бути не такою ефективною для цієї задачі.

### *2. Agile:*

- *Переваги для iHR's.* Agile є дуже гнучким і дозволяє швидко адаптуватися до змін у вимогах HR-менеджерів. Це важливо для проекту iHR's, оскільки змінюються потреби користувачів і можуть з'являтися нові функції, які потрібно додавати в систему на кожному етапі розробки.

- *Недоліки для iHR's.* Agile вимагає високого рівня взаємодії між розробниками та користувачами, що може бути складним для великих команд або організацій з багатьма рівнями управління.

### 3. Scrum:

- *Опис.* Scrum — одна з найпоширеніших методологій в Agile. Вона зосереджена на виконанні роботи в коротких ітераціях (спринтах), кожен з яких триває від одного до чотирьох тижнів. Команда регулярно проводить короткі зустрічі (Daily Scrum), щоб обговорити прогрес і вирішити проблеми.
- *Переваги для iHR's.* Scrum дозволяє зберігати постійну комунікацію між командою розробників і замовником, що є важливим для коригування вимог і виявлення помилок на ранніх етапах. Для проекту iHR's це ідеальний варіант, оскільки він забезпечує регулярне отримання зворотного зв'язку від HR-менеджерів, що допомагає оперативно вносити зміни в систему.
- *Недоліки для iHR's.* Scrum може бути не настільки ефективним для дуже великих команд або проектів з численними функціональними блоками, де складно організувати тісну взаємодію на кожному спринті[25].

### 4. Kanban:

- *Опис.* Kanban — це методологія, орієнтована на візуалізацію і управління процесами. Всі завдання відображаються на спеціальній дошці, де кожен етап роботи представлений окремою колонкою. Це дозволяє команді чітко бачити поточний стан проекту та оперативно реагувати на зміни.
- *Переваги для iHR's.* Kanban підходить для проекту, який передбачає постійну адаптацію та наявність великої кількості дрібних завдань, таких як оновлення бази даних співробітників, налаштування звітності тощо. Він дозволяє ефективно керувати завданнями і візуалізувати прогрес у розробці.

- *Недоліки для iHR's.* Kanban менш структурований і може бути складним для великих проєктів, де потрібно чітко визначити етапи розробки.

#### 5. *Extreme Programming (XP):*

- *Опис.* XP — це методологія, яка фокусується на забезпеченні високої якості продукту через автоматизацію тестування, швидкі ітерації та постійну комунікацію між членами команди. Вона також включає практики, такі як парне програмування і постійне рефакторинг коду.
- *Переваги для iHR's.* Для проєкту iHR's важливим аспектом є висока якість програмного забезпечення, оскільки помилки можуть мати серйозні наслідки для бізнес-процесів компанії. XP допоможе підтримувати високу якість через регулярне тестування і інтеграцію нових функцій.
- *Недоліки для iHR's.* XP вимагає високого рівня взаємодії між розробниками і може бути важким для реалізації в умовах недостатнього часу або обмежених ресурсів[20, 24].

### **2.7. Обґрунтування вибору моделі управління для проєкту розробки iHR's**

Для проєкту автоматизації управління кадровими ресурсами «iHRs» найбільш доцільним є використання ітеративної моделі розробки у поєднанні з класичною каскадною методологією управління проєктами. Цей підхід дозволить забезпечити чітке планування та поетапну реалізацію системи, одночасно враховуючи потребу у внесенні змін на основі проміжних результатів.

#### **Методологія проєкту**

Методологія управління проєктом базується на принципах каскадної моделі. Основними її аспектами є:

Чіткий поділ етапів роботи: кожен етап має бути завершений до початку наступного, що забезпечує структурованість і передбачуваність процесу.

Детальне планування: створюється чіткий календарний план із визначенням обсягу робіт для кожного етапу.

Документування результатів: після завершення кожного етапу формується звіт, який затверджується перед початком наступного.

Основна перевага каскадного підходу в тому, що він забезпечує чіткість у виконанні завдань, що особливо важливо для складних проектів з інтеграцією багатьох компонентів.

### **Ітеративна модель розробки**

Розробка системи «iHRs» базується на ітеративній моделі, що дозволяє досягати проміжних результатів і отримувати зворотний зв'язок від замовника на кожному етапі.

#### **Основні етапи ітеративної моделі:**

Планування ітерації — визначаються завдання, які будуть виконані в межах ітерації. Описуються вимоги до функціональності, які необхідно реалізувати.

Реалізація — проєктуються і створюються компоненти системи, які відповідають поставленим задачам. Кодуються окремі модулі та інтегруються в систему.

Тестування — перевіряється функціональність, стабільність роботи модулів і їх сумісність з іншими компонентами. Виявляються недоліки, які виправляються до початку наступної ітерації.

Оцінка результатів — збирається зворотний зв'язок від зацікавлених сторін. Коригуються вимоги до наступної ітерації на основі отриманих даних.

Наступна ітерація — на основі оцінки результатів попередньої ітерації формуються нові завдання, і процес повторюється.

#### **Переваги ітеративної моделі для проекту «iHRs»**

Гнучкість у розробці — можливість вносити зміни у функціональність на основі зворотного зв'язку.

Зменшення ризиків — проміжне тестування допомагає вчасно виявляти недоліки й виправляти їх.

Ранній доступ до результатів — вже після перших ітерацій замовник отримує частково функціональний продукт.

Масштабованість — система розробляється модульно, що дозволяє легко додавати нові функції у майбутньому.

## **Особливості застосування ітеративної моделі у «iHRs»**

Перша ітерація: реалізація базових функцій — створення профілів співробітників, облік даних, генерація звітів.

Друга ітерація: додавання модуля управління відпустками, включаючи автоматизацію розрахунків залишків відпусток.

Третя ітерація: інтеграція з бухгалтерськими платформами для автоматизації нарахування зарплати.

Четверта ітерація: впровадження системи звітності та аналітики для HR-відділу.

Остаточна ітерація: тестування системи на повну функціональність і підготовка до розгортання.

## РОЗДІЛ 3. ЗАСТОСУВАННЯ ТЕХНОЛОГІЙ УПРАВЛІННЯ ІТ-ПРОЄКТОМ

### 3.1. Життєвий цикл проєкту

Життєвий цикл проєкту автоматизації управління кадровими ресурсами «iHRs» включає наступні ключові етапи:

#### 1. Ініціація

На цьому етапі формулюються цілі проєкту, визначаються ключові зацікавлені сторони, аналізуються вимоги і розробляється статут проєкту. Також проводиться попередня оцінка можливих ризиків та формування першочергових завдань для реалізації проєкту.

#### 2. Планування

Планування охоплює всі аспекти, від збору та аналізу вимог до створення технічного завдання, визначення обсягу робіт і формування детального календарного плану з ресурсами та бюджетом. Цей етап включає також підготовку та формулювання критеріїв успішності проєкту.

#### 3. Реалізація

На етапі реалізації здійснюється розробка основних компонентів системи (архітектури, бази даних, функціональних модулів). Створюється інтерфейс, функціональність та здійснюється інтеграція всіх частин системи.

#### 4. Тестування

Тестування функціональності, продуктивності, безпеки та інтеграції є критичним етапом. На цьому етапі виявляються всі помилки та недоліки, які необхідно виправити до впровадження системи.

#### 5. Впровадження

Впровадження включає налаштування програмного забезпечення, підготовку кінцевих користувачів, проведення навчальних тренінгів та забезпечення безперебійної роботи в умовах реального використання. На цьому етапі система передається в експлуатацію.

#### 6. Експлуатація та підтримка

Після впровадження проекту проводиться моніторинг системи, підтримка користувачів, оновлення і коригування функціональності у відповідь на зміни вимог або виправлення помилок.

## **3.2. Команда та кошторис проекту**

### **3.2.1. Організаційна структура команди проекту**

#### *1. Проектний менеджер*

Основні обов'язки:

- Керує загальним процесом реалізації проекту: від ініціації до завершення.
- Розробляє та підтримує план проекту, включаючи терміни, бюджет, ресурси та контроль за виконанням завдань.
- Координує комунікацію між усіма учасниками проекту.
- Контролює виконання ключових етапів проекту (планування, розробка, тестування, впровадження).
- Вирішує конфлікти та здійснює управління ризиками.
- Взаємодіє з замовником проекту, а також визначає пріоритети на основі вимог бізнесу.

Ключові навички:

- Управління проектами (Agile, Scrum).
- Ризик-менеджмент.
- Лідерські та організаційні навички.
- Комунікативні навички.

#### *2. Бізнес-аналітик*

Основні обов'язки:

- Збір та аналіз вимог від зацікавлених сторін (HR-фахівців, керівників, кінцевих користувачів).
- Розробка бізнес-стратегії для покращення процесів, що підлягають автоматизації.
- Визначати функціональні вимоги до продукту та працювати з технічною командою над їх реалізацією.

- Оцінювати та аналізувати конкурентні продукти та ринкові тенденції для покращення функціональності системи.
- Формулювати рекомендації щодо впровадження нових технологій та практик для покращення ефективності роботи HR.

Ключові навички:

- Аналіз бізнес-процесів.
- Розробка вимог та документація.
- Комунікація з замовниками.
- Поглиблене розуміння HR-процесів.

### *3. Дизайнер інтерфейсу*

Основні обов'язки:

- Розробляти концептуальний та візуальний дизайн інтерфейсу продукту.
- Забезпечувати зручність користування (UX/UI) для кінцевих користувачів, орієнтуючись на потреби HR-фахівців.
- Створювати прототипи та макети інтерфейсу для мобільних додатків та веб-інтерфейсів.
- Тестувати різні варіанти дизайну для покращення зручності використання продукту.

Ключові навички:

- Досвід роботи з дизайном UX/UI.
- Використання інструментів для створення макетів (Figma, Adobe XD).
- Оцінка зручності користування.
- Тестування інтерфейсу.

### *5. Бекенд-розробник*

Основні обов'язки:

- Розробляти серверну частину програми, створюючи API для взаємодії з мобільними додатками та іншими системами.
- Забезпечувати збереження даних у базах даних та оптимізує запити для швидкого доступу до інформації.

- Працювати над безпекою даних, включаючи захист інформації, що зберігається в системі.
- Співпрацювати з іншими розробниками для інтеграції бек-енду та користувацького інтерфейсу.

Ключові навички:

- Програмування на Python, Java, або Node.js.
- Робота з базами даних (SQL, NoSQL).
- Розробка API та інтеграція з фронтендом.
- Оптимізація серверних запитів та безпека даних.

#### *6. Тестувальник*

Основні обов'язки:

- Виконувати різні види тестування (функціональне, інтеграційне, регресійне) для забезпечення якості програмного продукту.
- Виявляти і відстежувати помилки та взаємодіяти з розробниками для їх виправлення.
- Створювати тестові плани та сценарії тестування на основі вимог.
- Перевіряти продукт на відповідність всім технічним і функціональним вимогам.

Ключові навички:

- Розробка тестових планів і сценаріїв.
- Виявлення багів та помилок.
- Підтримка високої якості коду та продукту.

#### *7. Продакт менеджер*

Основні обов'язки:

- Визначати стратегію розвитку продукту та встановлювати його цілі й ключові характеристики.
- Розробляти та контролювати життєвий цикл продукту, від етапу планування до запуску.
- Визначати пріоритети характеристик на функцій продукту, взаємодіючи з командами розробки та маркетингу.

- Аналізувати результати впровадження та дає рекомендації щодо покращення.

Ключові навички:

- Стратегічне планування продукту.
- Співпраця з розробниками та замовниками.
- Аналіз ринкових потреб та конкурентів.

## 8. Маркетолог

Основні обов'язки:

- Розробляти стратегії просування продукту на ринок.
- Проводити маркетингові дослідження для визначення цільової аудиторії та конкурентного середовища.
- Планувати кампанії щодо залучення стейкхолдерів й користувачів та покращення впізнаваності бренду продукту.
- Аналізувати ефективність маркетингових заходів та коригувати стратегії.

Ключові навички:

- Розробка маркетингових кампаній.
- Дослідження цільових аудиторій.
- Стратегії просування на ринку.

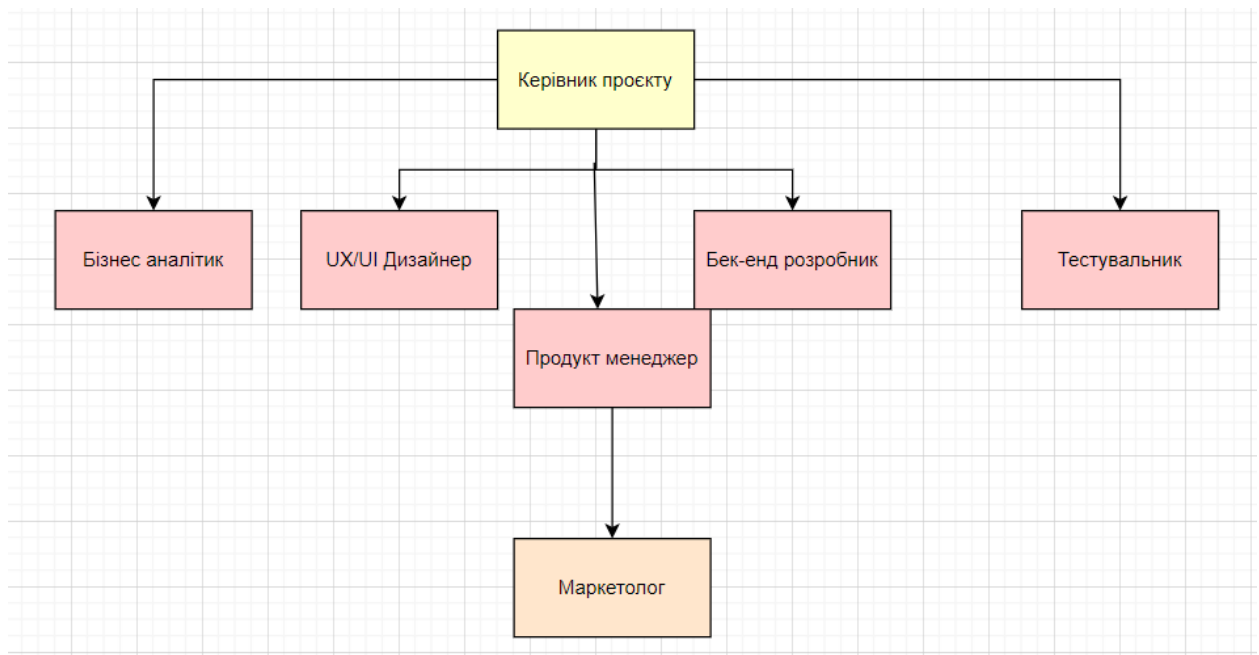


Рис.3.1. Організаційна структура проєкту

### 3.2.2. Кошторис проекту

#### Матеріальні ресурси

##### 1. Інтернет-з'єднання

Для забезпечення безперебійного доступу до мережі, обміну інформацією та спілкування в команді.

- Вартість: \$300 на місяць
- Тривалість: 6 місяців
- Загальні витрати: \$1,800

##### 2. Програмне забезпечення

Необхідне для розробки, тестування та впровадження системи iHR.

Ліцензії на спеціалізоване програмне забезпечення, інструменти для розробки і тестування.

- Вартість ліцензій для розробки та тестування: \$3,000
- Операційні системи та інструменти для розробки (наприклад, IDE, бази даних): \$1,500
- Загальні витрати на програмне забезпечення: \$4,500

##### 3. Серверне обладнання

Оренда серверів для забезпечення роботи iHR-системи, зберігання даних і високої доступності для користувачів.

- Оренда серверів: \$1,500 на місяць
- Тривалість: 6 місяців
- Загальні витрати: \$9,000

##### 4. Офісне обладнання та меблі

Офісне приміщення, меблі та робочі місця для команди розробників, тестувальників та інших учасників проекту.

- Оренда офісу: \$2,000 на місяць
- Тривалість: 6 місяців
- Загальні витрати: \$12,000
- Меблі та робочі місця: \$3,000

## 5. Засоби зв'язку

Телефони, відеоконференційні системи та інші інструменти для ефективної комунікації між учасниками проекту.

- Вартість телефонних послуг та відеоконференцій: \$500 на місяць
- Тривалість: 6 місяців
- Загальні витрати: \$3,000

## 6. Спеціалізоване обладнання

Для розробки і підтримки системи можуть знадобитися спеціалізовані програмні засоби або обладнання.

- Вартість спеціалізованого обладнання: \$3,000

## 7. Резервні матеріали та компоненти

Витрати на непередбачувані ситуації, резервні частини для підтримки проекту.

- Загальні витрати: \$1,000

*Таблиця 3.1.*

### **Витрати на людські ресурси**

Посада	Ціна (\$/міс)	Тривалість(міс)	Загальні витрати
Проектний менеджер	\$5,000	6	\$30,000
Бізнес-аналітик	\$4,500	6	\$27,000
Дизайнер інтерфейсу	\$3,500	3	\$10,500
Фронтенд-розробник	\$4,500	6	\$27,000
Бекенд-розробник	\$4,500	6	\$27,000
Тестувальник	\$3,500	6	\$21,000
Продакт менеджер	\$5,000	6	\$30,000
Маркетолог	\$4,000	6	\$24,000

Таблиця 3.2

**Додаткові витрати**

Категорія	Вартість	Тривалість(міс)	Загальні витрати (\$)
Оренда офісного простору	\$2,000	6	\$12,000
Обладнання та програмне забезпечення	\$1,050	6	\$6,300
Реклама і маркетинг	\$800	6	\$4,800
Логіка та транспорт	\$500	6	\$3,000
Витрати управління ризиками	\$5,000	6	\$30,000
Правова підтримка	\$1,000	6	\$6,000
Інновації дослідження	\$1,000	4	\$4,000

Таблиця 3.3

**Загальні витрати**

Категорія	Загальні витрати (\$)
Витрати на команду	\$196,000
Додаткові витрати	\$66,100
Загальний бюджет проекту	\$262,100

Пояснення витрат:

### 1. Персонал

Загальні витрати на команду складають **\$196,000**. Ця сума включає зарплати основних учасників проекту: проектного менеджера, бізнес-аналітика, дизайнерів, розробників, тестувальників, продакт-менеджера та маркетолога протягом 6 місяців. Наприклад:

- Проектний менеджер – \$30,000
- Бізнес-аналітик – \$27,000
- Тестувальник – \$21,000

### 2. Матеріальні ресурси

Витрати на необхідні для реалізації проекту матеріальні ресурси складають:

- **Інтернет-з'єднання:** \$1,800 (по \$300/місяць за 6 місяців).
- **Програмне забезпечення:** \$4,500 (ліцензії, інструменти розробки та тестування).
- **Серверне обладнання:** \$9,000 (по \$1,500/місяць за 6 місяців).
- **Офісне обладнання та меблі:** \$12,000 за оренду офісу на 6 місяців і \$3,000 за меблі та робочі місця.
- **Засоби зв'язку:** \$3,000 (по \$500/місяць за 6 місяців).
- **Спеціалізоване обладнання:** \$3,000 для підтримки роботи проекту.
- **Резервні матеріали:** \$1,000 для непередбачуваних витрат.

**Загальна сума витрат на матеріальні ресурси: \$34,300.**

### 3. Додаткові витрати

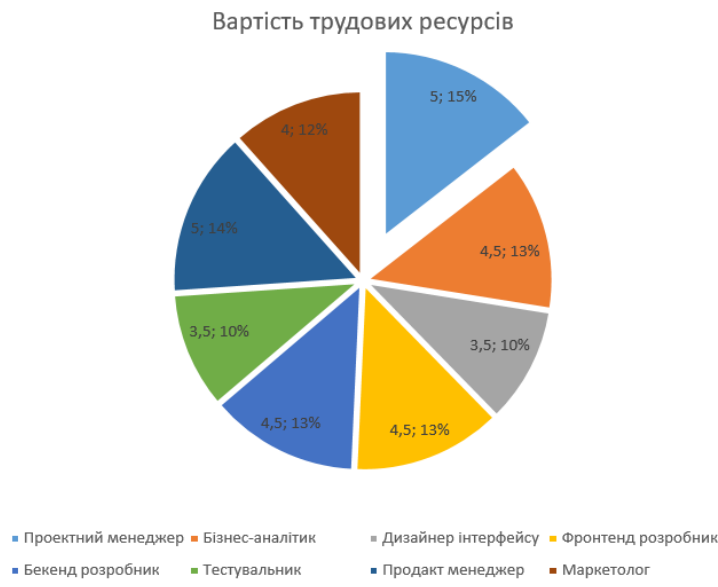
Включають витрати, необхідні для підтримки проекту:

- **Оренда офісного простору:** \$12,000 (по \$2,000/місяць за 6 місяців).
- **Обладнання та програмне забезпечення:** \$6,300 (по \$1,050/місяць за 6 місяців).
- **Реклама і маркетинг:** \$4,800 (по \$800/місяць за 6 місяців).

- **Логістика та транспорт:** \$3,000 (по \$500/місяць за 6 місяців).
- **Управління ризиками:** \$30,000 (по \$5,000/місяць за 6 місяців).
- **Правова підтримка:** \$6,000 (по \$1,000/місяць за 6 місяців).
- **Інновації та дослідження:** \$4,000 (по \$1,000/місяць за 4 місяці).

**Загальна сума додаткових витрат: \$66,100**

На рис.3.2. зображена діаграма, яка показує вартість трудових ресурсів. На ній, видно, що найбільше ресурсів витрачатиметься на проектного менеджера.



**Рис. 3.2. Вартість трудових ресурсів**

### 3.3. Управління змістом проекту. WBS і роботи проекту

WBS, або структура розбиття робіт (Work Breakdown Structure), визначається як інструмент управління проектами, який сприяє розкладенню проекту на менші більш керовані компоненти. Це ієрархічна структура, яка включає в себе всі роботи, необхідні для досягнення цілей проекту.

WBS вирізняється тим, що дозволяє ефективно організувати та контролювати весь проект, а також визначати залежності між окремими роботами та розподіляти їх між командою проекту. Кожен рівень ієрархії WBS представляє собою все більш деталізовані етапи чи елементи проекту, що дозволяє керівникам та членам команди легше розуміти та виконувати свої завдання.

WBS виявляється корисним інструментом для визначення обсягу проекту, планування ресурсів, встановлення контрольних точок та забезпечення здійснення проекту в рамках чітко визначених меж. Цей метод сприяє збільшенню ефективності управління та зменшенню ризиків, пов'язаних із реалізацією проектних завдань.

## 1. Проект iHR's

### 1.1. Ініціація проекту

- 1.1.1. Визначення цілей проекту
- 1.1.2. Аналіз зацікавлених сторін
- 1.1.3. Визначення обсягу проекту
- 1.1.4. Складання статуту проекту
- 1.1.5. Призначення менеджера проекту

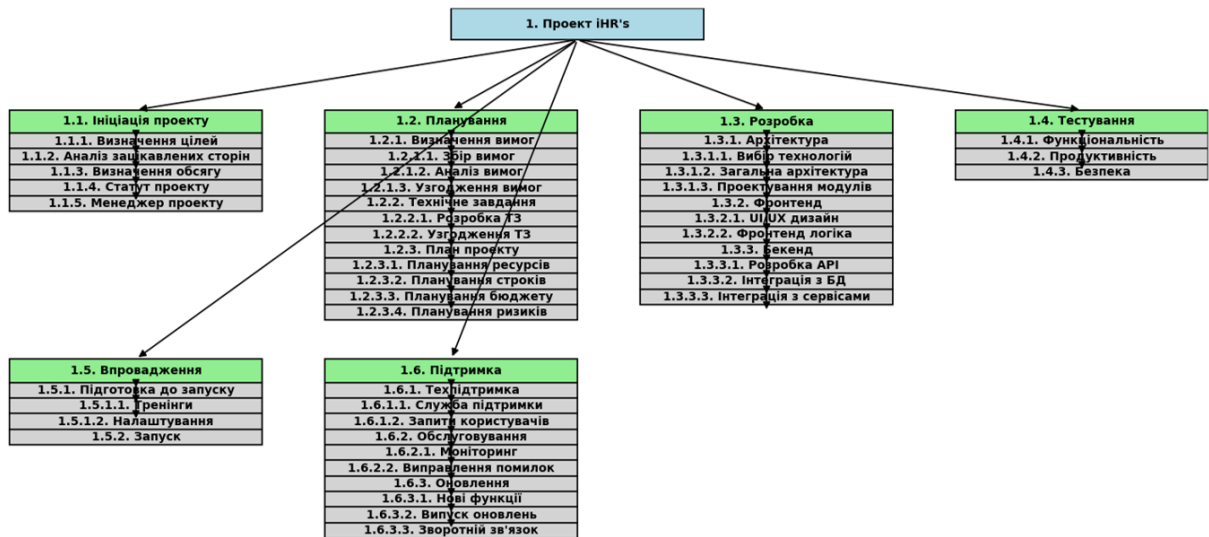
### 1.2. Планування

- 1.2.1. Визначення вимог
  - 1.2.1.1. Збір вимог від зацікавлених сторін
  - 1.2.1.2. Аналіз вимог
  - 1.2.1.3. Узгодження вимог
- 1.2.2. Створення технічного завдання
  - 1.2.2.1. Розробка ТЗ
  - 1.2.2.2. Узгодження ТЗ
- 1.2.3. Розробка плану проекту
  - 1.2.3.1. Планування ресурсів
  - 1.2.3.2. Планування строків
  - 1.2.3.3. Планування бюджету
  - 1.2.3.4. Планування ризиків

### 1.3. Розробка

- 1.3.1. Проектування архітектури
  - 1.3.1.1. Вибір технологій
  - 1.3.1.2. Створення загальної архітектури системи
  - 1.3.1.3. Проектування модулів
- 1.3.2. Розробка фронтенду

- 1.3.2.1. Розробка UI/UX дизайну
- 1.3.2.2. Імплементация фронтенд логіки
- 1.3.3. Розробка бекенду
  - 1.3.3.1. Розробка API
  - 1.3.3.2. Інтеграція з базою даних
  - 1.3.3.3. Інтеграція з зовнішніми сервісами
- 1.4. Тестування
  - 1.4.1. Тестування функціональності
  - 1.4.2. Тестування продуктивності
  - 1.4.3. Тестування безпеки
- 1.5. Впровадження
  - 1.5.1. Підготовка до запуску
    - 1.5.1.1. Тренінги для користувачів
    - 1.5.1.2. Налаштування системи
  - 1.5.2. Запуск системи
- 1.6. Підтримка
  - 1.6.1. Технічна підтримка
    - 1.6.1.1. Організація служби підтримки
    - 1.6.1.2. Відповіді на запити користувачів
  - 1.6.2. Обслуговування системи
    - 1.6.2.1. Моніторинг системи
    - 1.6.2.2. Виправлення помилок
  - 1.6.3. Оновлення
    - 1.6.3.1. Розробка нових функцій
    - 1.6.3.2. Випуск оновлень
    - 1.6.3.3. Зворотній зв'язок з користувачами



**Рис. 3.3. Декомпозиція по процесам**

## 1. Продукт iHR's

### 1.1. Інтерфейс користувача (UI)

#### 1.1.1. Головна сторінка

- 1.1.1.1. Логотип
- 1.1.1.2. Навігаційне меню
- 1.1.1.3. Панель пошуку
- 1.1.1.4. Секція новин
- 1.1.1.5. Футер

#### 1.1.2. Сторінка реєстрації

- 1.1.2.1. Форма реєстрації
- 1.1.2.2. Політика конфіденційності
- 1.1.2.3. Кнопка реєстрації

#### 1.1.3. Сторінка входу

- 1.1.3.1. Форма входу
- 1.1.3.2. Забули пароль
- 1.1.3.3. Кнопка входу

#### 1.1.4. Панель адміністратора

- 1.1.4.1. Управління користувачами
- 1.1.4.2. Налаштування системи
- 1.1.4.3. Звіти та аналітика

## 1.2. Система обробки даних

### 1.2.1. База даних

1.2.1.1. Схема бази даних

1.2.1.2. Таблиці користувачів

1.2.1.3. Таблиці транзакцій

1.2.1.4. Таблиці логів

### 1.2.2. Серверна логіка (бекенд)

1.2.2.1. Аутентифікація

1.2.2.2. Авторизація

1.2.2.3. Обробка запитів API

1.2.2.4. Обробка транзакцій

1.2.2.5. Логування та моніторинг

## 1.3. Система обробки запитів

### 1.3.1. Веб-сервер

1.3.1.1. Налаштування веб-сервера

1.3.1.2. Сертифікати безпеки (SSL)

### 1.3.2. API

1.3.2.1. API для користувачів

1.3.2.2. API для адміністратора

1.3.2.3. API для сторонніх сервісів

### 1.3.3. Балансувальник навантаження

1.3.3.1. Конфігурація балансувальника

1.3.3.2. Моніторинг трафіку

## 1.4. Документація

### 1.4.1. Технічна документація

1.4.1.1. Документація API

1.4.1.2. Схема архітектури

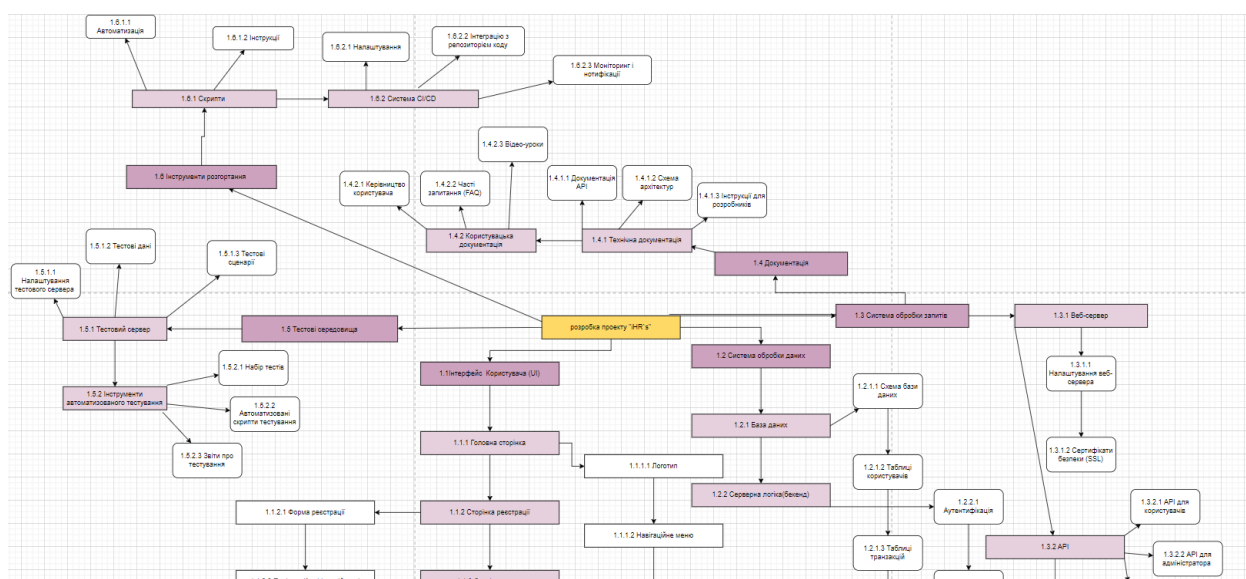
1.4.1.3. Інструкції для розробників

### 1.4.2. Користувацька документація

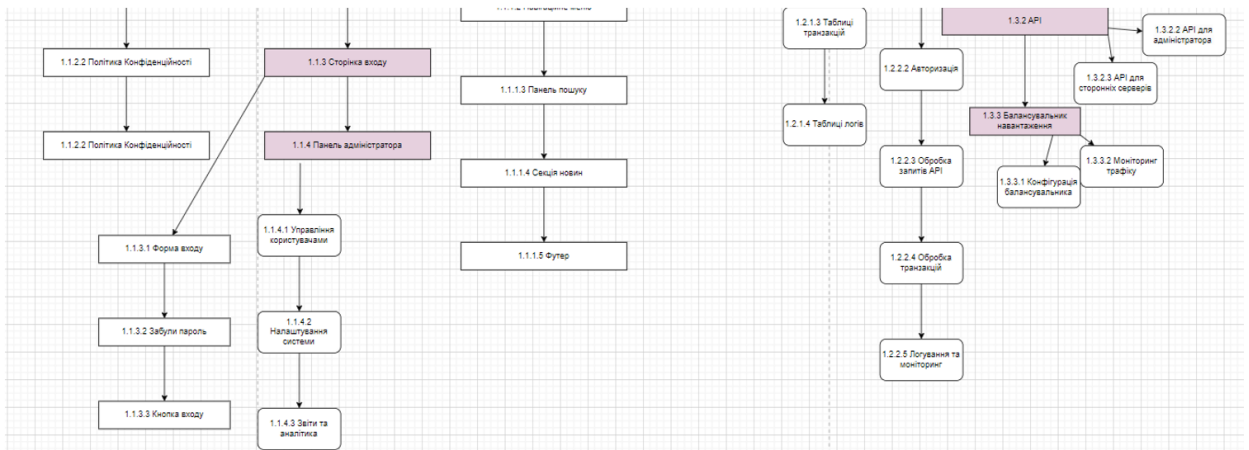
1.4.2.1. Керівництво користувача

1.4.2.2. Часті запитання (FAQ)

- 1.4.2.3. Відео-уроки
- 1.5. Тестові середовища
  - 1.5.1. Тестовий сервер
    - 1.5.1.1. Налаштування тестового сервера
    - 1.5.1.2. Тестові дані
    - 1.5.1.3. Тестові сценарії
  - 1.5.2. Інструменти автоматизованого тестування
    - 1.5.2.1. Набір тестів
    - 1.5.2.2. Автоматизовані скрипти тестування
    - 1.5.2.3. Звіти про тестування
- 1.6. Інструменти розгортання
  - 1.6.1. Скрипти розгортання
    - 1.6.1.1. Автоматизація розгортання
    - 1.6.1.2. Інструкції по розгортанню
  - 1.6.2. Система CI/CD
    - 1.6.2.1. Налаштування CI/CD
    - 1.6.2.2. Інтеграція з репозиторієм коду
    - 1.6.2.3. Моніторинг і нотифікації



**Рис. 3.4. Декомпозиція проекту по продуктам**



**Рис. 3.5 (Продовження) Декомпозиція проекту по продуктам**

### **3.4. Управління часом проекту. Діаграма Ганта**

#### **Визначення термінів та етапів реалізації**

**Ініціація проекту** — передбачає визначення цілей, складання статуту проекту, аналіз зацікавлених сторін та їхніх очікувань.

Аналіз зацікавлених сторін: 1 тиждень.

Формування обсягу проекту: 2 тижні.

**Планування** — етап включає збір та аналіз вимог, створення технічного завдання (ТЗ) та формування календарного плану. Всі залежності між завданнями враховуються при розробці графіка.

#### **Розробка**

Проектування архітектури: 2 тижні.

Розробка модулів системи (інтерфейс, функціональність, база даних): 3 ітерації по 3 тижні кожна.

Інтеграція компонентів системи: 2 тижні.

#### **Тестування та впровадження**

Тестування функціональності та продуктивності: 2 тижні.

Налаштування системи та підготовка користувачів: 1 тиждень.

#### **Вибір моделі управління проектом**

Для проекту застосовується каскадна модель управління, а розробка здійснюється за ітеративним підходом. Завдання виконуються послідовно з можливістю внесення коригувань після завершення кожної ітерації.

## **Створення графіка**

- Діаграма Ганта відображає основні етапи (ініціація, планування, розробка, тестування, впровадження), їх тривалість і залежності.
- Критичні завдання (наприклад, інтеграція модулів) включають часові буфери для мінімізації ризиків.

## **Графік проведення мітингів**

- Щотижневі зустрічі для підведення підсумків кожного етапу.
- Контрольні точки (milestones): завершення етапу проектування, розробки модулів та тестування.

## **Контроль якості**

- Верифікація відповідності результатів початковим вимогам на кожному етапі.
- Тестування модулів на етапі інтеграції та перевірка стабільності системи.

## **Контроль виконання задач**

- Моніторинг прогресу за допомогою інструментів управління проектами (наприклад, MS Project).
- Щотижневе оновлення статусу завдань із визначенням пріоритетів для уникнення затримок.

	⊕	Назва	Тривалість	Початок	Відзначити задачу як в...	Resource Initials	Закінчення
1		1. Підготовчий етап	6 днів	09.09.24 8:00	<input checked="" type="checkbox"/>		16.09.24 17:00
2	📅	Формування команди проєкту	2 днів	10.09.24 8:00	<input type="checkbox"/>		11.09.24 17:00
3	📅	Зустріч команди	1 день	12.09.24 8:00	<input type="checkbox"/>		12.09.24 17:00
4	📅	Попередній аналіз	1 день	13.09.24 8:00	<input type="checkbox"/>		13.09.24 17:00
5	📅	Організація інструментів	1 день	16.09.24 8:00	<input type="checkbox"/>		16.09.24 17:00
6	📅	2. Аналіз і проектування	46 днів?	17.09.24 8:00	<input checked="" type="checkbox"/>		19.11.24 17:00
7	📅	Збір і аналіз вимог	11 днів?	18.09.24 8:00	<input type="checkbox"/>		02.10.24 17:00
8	📅	Проектування інтерфейсу	21 днів?	03.10.24 8:00	<input type="checkbox"/>		31.10.24 17:00
9	📅	Архітектура системи	12,875 днів?	01.11.24 9:00	<input type="checkbox"/>		19.11.24 17:00
10	📅	3. Розробка системи	28,875 днів?	20.11.24 9:00	<input checked="" type="checkbox"/>		30.12.24 17:00
11	📅	Frontend-розробка	9,875 днів?	21.11.24 9:00	<input type="checkbox"/>		04.12.24 17:00
12	📅	Backend-розробка	15,875 днів?	05.12.24 9:00	<input type="checkbox"/>		26.12.24 17:00
13	📅	Інтеграція компонентів	1 день?	27.12.24 9:00	<input type="checkbox"/>		30.12.24 9:00
14	📅	4. Тестування	46,875 днів?	02.01.25 9:00	<input checked="" type="checkbox"/>		07.03.25 17:00
15	📅	Функціональне тестування	8,875 днів?	02.01.25 9:00	<input type="checkbox"/>		14.01.25 17:00
16	📅	Ресурсне тестування	11,875 днів?	15.01.25 9:00	<input type="checkbox"/>		30.01.25 17:00
17	📅	Навантажувальне тестування	5,875 днів?	03.02.25 9:00	<input type="checkbox"/>		10.02.25 17:00
18	📅	Усунення помилок	18,875 днів?	11.02.25 9:00	<input type="checkbox"/>		07.03.25 17:00
19	📅	5. Впровадження	20,875 днів?	07.03.25 9:00	<input checked="" type="checkbox"/>		04.04.25 17:00
20	📅	Пілотне впровадження	5,875 днів?	10.03.25 9:00	<input type="checkbox"/>		17.03.25 17:00
21	📅	Навчання персоналу	8,875 днів?	18.03.25 9:00	<input type="checkbox"/>		28.03.25 17:00
22	📅	Розгортання у продакшн	5 днів?	31.03.25 8:00	<input type="checkbox"/>		04.04.25 17:00
23	📅	6. Завершення проєкту	14 днів?	07.04.25 8:00	<input checked="" type="checkbox"/>		24.04.25 17:00
24	📅	Оцінка результатів	5 днів?	07.04.25 8:00	<input type="checkbox"/>		11.04.25 17:00
25	📅	Передача технічної документації	1 день?	14.04.25 8:00	<input type="checkbox"/>		14.04.25 17:00
26	📅	Усунення дрібних помилок	4 днів?	15.04.25 8:00	<input type="checkbox"/>		18.04.25 17:00
27	📅	Підготовка презентації проєкту	1 день?	21.04.25 8:00	<input type="checkbox"/>		21.04.25 17:00
28	📅	Проведення фінальної зустрічі	1 день?	23.04.25 8:00	<input type="checkbox"/>		23.04.25 17:00

Рис. 3.6. Етапи проєкту

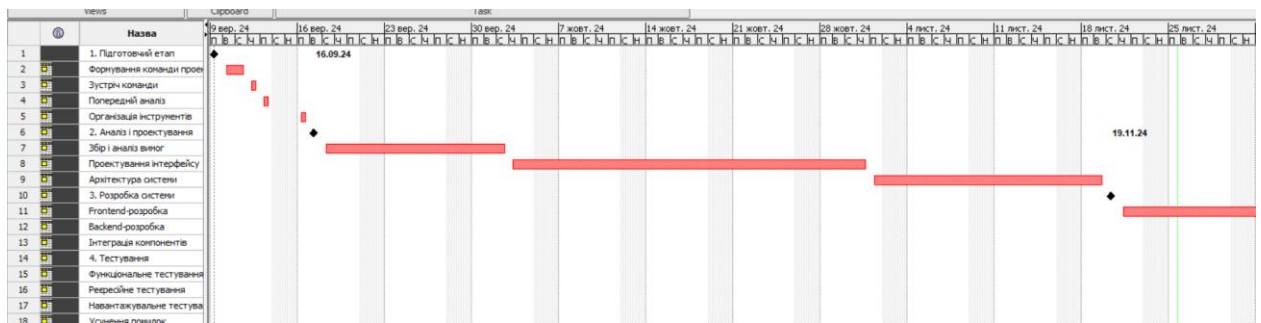


Рис. 3.7. Діаграма Ганта

### 3.5. Управління ризиками проєкту

Управління ризиками є ключовим аспектом для забезпечення успіху проєкту. Оскільки проєкт включає в себе технічні та організаційні складнощі, важливо ідентифікувати можливі ризики на всіх етапах його реалізації, оцінити їх вплив і розробити стратегії для їхнього зменшення чи контролю.

#### Властивості ризиків у проєкті iHR's:

1. Ймовірність виникнення. Оцінка ймовірності того, що конкретний ризик здійсниться.

-Висока ймовірність. Наприклад, затримки в розробці через неефективну комунікацію в команді.

-Низька ймовірність. Проблеми з безпекою в разі належного впровадження протоколів захисту даних.

## 2. **Величина впливу.** Оцінка масштабу впливу ризику на проект.

-Фінансові втрати. Затримка термінів може призвести до фінансових втрат через надмірні витрати на розробку або штрафи.

-Затримки у графіку. Якщо певні етапи проекту не будуть виконані вчасно, це може призвести до відставання від основного плану.

-Втрату ресурсів. Зміна вимог замовника може привести до перевитрати ресурсів на реалізацію нових функцій.

## 3. **Ступінь контролю.** Оцінка того, наскільки ефективно можна контролювати або уникнути ризик.

-Керовані ризики. Наприклад, ризик затримок через неефективну комунікацію — його можна мінімізувати за допомогою чітко визначених регулярних мітингів.

-Не керовані ризики. Ризики, що виникають через зовнішні фактори, такі як зміни в законодавстві.

## **Ризики проекту iHR's**

### 1. **Внутрішні ризики**

-Технічні проблеми. Проблеми з інтеграцією компонентів системи або вибір ненадійних технологій для автоматизації процесів.

-Зміни вимог. Клієнт може вносити зміни у вимоги до функціональності системи після початку розробки, що призведе до додаткових витрат та затримок.

-Недостатня кваліфікація персоналу. Невідповідність рівня кваліфікації членів команди потребам проекту може привести до помилок або затримок у розробці.

### 2. **Зовнішні ризики**

-Зміни в законодавстві. Несподівані зміни в нормативно-правовій базі, що регулює сферу управління персоналом (наприклад, зміни в законах про захист даних), можуть вимагати коригування проекту.

-Економічні коливання. Зміни в економічній ситуації, наприклад, інфляція або кризові явища, можуть вплинути на фінансування або ресурси проекту.

### **3. Ризики людського фактору**

-Конфлікти в команді. Невідповідність у підходах або поганою комунікацією може призвести до непорозумінь та затримок.

-Втрати ключового персоналу. Зміна в складі команди, особливо при втраті ключових розробників або менеджера проекту, може серйозно вплинути на хід проекту.

### **4. Технічні ризики**

-Програмні помилки. Наявність багів або недопрацьованих компонентів у системі, які можуть призвести до збоїв у роботі продукту.

-Неузгодженість технологій. Використання несумісних технологій може створити складнощі в інтеграції різних частин системи.

### **5. Фінансові ризики**

-Неспроможність залучити фінансування. Проблеми з отриманням необхідних ресурсів можуть призвести до затримок або зупинки проекту.

-Перевитрати бюджету. Недооцінка вартості ресурсів або необхідних технологій може призвести до перевищення бюджету проекту.

## **Етапи управління ризиками для проекту iHR's**

### **1. Ідентифікація ризиків**

-Проведення аналізу проекту та складання списку можливих ризиків.

-Використання методів, таких як **SWOT-аналіз**, для виявлення сильних і слабких сторін проекту, а також можливих загроз.

-Оцінка ризиків, пов'язаних з вимогами клієнта, технологіями, командою, і зовнішніми факторами.

### **2. Аналіз ризиків**

-Оцінка ймовірності та величини впливу кожного ризику.

-Створення матриці ризиків для визначення пріоритетів у їхньому управлінні.

-Розробка сценаріїв впливу для кожного ризику та оцінка варіантів їхнього зменшення.

### **3. Планування відповіді на ризики**

-**Уникнення.** Якщо ризик може бути усунутий, планується зміна стратегії для його мінімізації.

-**Зменшення.** Розробка плану дій для зменшення ймовірності або впливу ризику (наприклад, додаткове тестування для зменшення технічних проблем).

-**Перенесення.** Перехід відповідальності на стороннього постачальника або партнера (наприклад, аутсорсинг деяких функцій).

-**Прийняття.** У деяких випадках ризики можуть бути прийняті, якщо їхній вплив малий або якщо контролювати їх неможливо.

### **4. Втілення стратегій**

-Впровадження розроблених стратегій і тактик для управління ризиками.

-Забезпечення належного моніторингу виконання планів і стратегій зменшення ризиків.

### **5. Моніторинг і контроль ризиків**

- Постійний моніторинг можливих ризиків протягом усього проекту.

-Оцінка змін в обставинах, що можуть призвести до нових або змінити існуючі ризики.

-Коригування стратегій у разі зміни ситуації (наприклад, якщо виникають нові законодавчі вимоги або змінюється економічна ситуація)[15].

На рис.3.8. у табличній формі описані ризики(тип, ризикова подя, сила впливу ризику, керованість ризиком).

Також на рис. 3.9. показано протиризикові заходи.

Тип ризику	Ризикова подія	Сила впливу	Керованість
Технічні ризики	Вибір ненадійних технологій для розробки	Висока	Частково керований
	Проблеми з інтеграцією компонентів системи	Висока	Частково керований
	Невідповідність технічних характеристик системи вимогам замовника	Середня	Частково керований
Фінансові ризики	Перевитрата бюджету через додаткові витрати на розробку	Висока	Частково керований
	Зменшення фінансування через економічну нестабільність	Висока	Низька
	Недостатньо фінансів для завершення проекту	Висока	Низька
Зовнішні ризики	Зміни в законодавстві щодо обробки персональних даних	Середня	Низька
	Зовнішні економічні фактори (інфляція, економічна криза)	Висока	Низька
	Зміни в політичній ситуації або регуляціях	Середня	Низька
Ризики людського фактору	Втрати ключового персоналу	Висока	Частково керований
	Конфлікти в команді	Середня	Частково керований
	Недостатня кваліфікація співробітників	Середня	Частково керований
Технічні складності в розробці	Помилки в коді або алгоритмах	Висока	Висока
	Проблеми з сумісністю програмного забезпечення	Середня	Частково керований
	Труднощі з адаптацією нових технологій до існуючої інфраструктури	Висока	Частково керований
Ризики безпеки та конфіденційності	Порушення конфіденційності даних користувачів	Висока	Висока
	Кібератаки або несанкціонований доступ до даних	Висока	Висока
	Недотримання стандартів безпеки та конфіденційності	Висока	Висока
Ризики, пов'язані з вимогами замовника	Зміна вимог замовника в процесі виконання проекту	Висока	Частково керований
	Неясні або неповні вимоги замовника	Висока	Частково керований
	Затримки в отриманні необхідної інформації від замовника	Середня	Частково керований
Ризики, пов'язані з тестуванням	Невиявлені помилки під час тестування системи	Висока	Частково керований
	Недостатньо тестування на різних етапах розробки	Висока	Частково керований
	Несвоєчасне виявлення дефектів на етапі приймальних тестів	Середня	Частково керований

Рис. 3.8. Ідентифікація ризиків проекту

	Ризикова подія	ПІРЗ 1 профілактика	Симптоми рання ознака	ПІРЗ 2 при симптомі	ПІРЗ 3 при проблемі
1	Невідповідність програмного забезпечення вимогам проекту	1. Проведення ринкового дослідження для вибору відповідного програмного забезпечення. 2. Розробка та затвердження детальних технічних вимог перед закупівлею.	Виявлення функціональних обмежень ПЗ під час первинного тестування.	Швидке коригування технічних вимог для внесення змін у програму.	Повна заміна програмного забезпечення на альтернативне.
2	Втрата ключового члена команди	Формування резервної команди для підтримки ключових ролей.	1. Ознаки зниження продуктивності або залучення до роботи. 2. Часті відгули або пропуски ключових зустрічей.	1. Підготовка детального плану заміни для ключової ролі. 2. Запровадження індивідуальних розмов для виявлення проблем.	Оперативний пошук нового фахівця через рекрутингові платформи.
3	Затримки у виконанні проекту через недостатню комунікацію	1. Регулярне проведення щотижневих зустрічей команди для обговорення прогресу. 2. Використання спеціалізованих інструментів управління проектами (наприклад, Trello, Jira).	Затримки у виконанні завдань за проміжними етапами проекту.	1. Ревізія каналів комунікації в команді. 2. Залучення додаткового проектного менеджера для оптимізації процесів.	1. Перегляд графіка проекту із врахуванням критичних затримок. 2. Розробка альтернативних шляхів досягнення ключових етапів.
4	Невідповідність кваліфікації команди вимогам проекту	1. Проведення попереднього аналізу компетенцій команди. 2. Запровадження програми регулярного навчання та сертифікації. Використання хмарних технологій із резервуванням даних.	1. Низька ефективність виконання складних задач. 2. Постійні помилки або перегляди раніше виконаних завдань.	1. Організація термінових навчальних сесій для ключових членів команди. 2. Залучення досвідчених менторів для консультування.	1. Тимчасове перекладення частини обов'язків на зовнішніх фахівців. 2. Масштабний набір додаткового персоналу для посилення команди.
5	Збої в роботі серверного обладнання		Збільшення часу відповіді серверів.	Проведення термінової технічної діагностики серверів.	Переміщення основних операцій до резервного серверного обладнання.

Рис. 3.9. Розробка протиризикових заходів

### **3.6. Управління якістю проєкту**

Управління якістю є невід'ємною частиною проєкту автоматизації HR-процесів «iHRs» і охоплює всі етапи проєкту, від планування до впровадження та підтримки. Основні принципи управління якістю полягають у забезпеченні відповідності вимогам та стандартам, задоволенні очікувань зацікавлених сторін і досягненні високої ефективності роботи системи.

#### **1. Планування якості**

На етапі планування якості визначаються критерії успішності проєкту, вимоги до продукту, методи і інструменти для забезпечення якості, а також плани тестування. Визначаються також основні показники якості, такі як надійність, продуктивність, безпека та зручність використання.

#### **2. Контроль якості**

На етапах розробки та тестування проводяться перевірки для забезпечення відповідності системи вимогам. Регулярно здійснюються перевірки коду, тестування функціональності та продуктивності, а також проведення тестів безпеки. Автоматизоване тестування дає змогу перевіряти стабільність та ефективність системи на кожному етапі.

#### **3. Управління змінами**

Управління якістю також включає в себе процеси управління змінами, які виникають в результаті коригувань, уточнень вимог або помилок, що виникають під час реалізації. Процеси управління змінами мають чітко зафіксовані процедури для оцінки, затвердження та реалізації змін.

#### **4. Заключний контроль якості**

Перед впровадженням продукту проводиться остаточна перевірка якості, щоб переконатися в досягненні усіх критеріїв якості. Це включає тестування стабільності та продуктивності, перевірку безпеки даних та відповідність стандартам.

## РОЗДІЛ 4. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ РЕАЛІЗАЦІЇ ІТ ПРОЄКТУ

*Проектування* — це важливий етап у розробці будь-якого програмного продукту, який передує його безпосередній реалізації. На цьому етапі фахівці з різних напрямків (дизайну, розробки, тестування) працюють над визначенням концепції, архітектури та створенням інтерфейсів продукту. Головною метою проектування є не тільки створення естетично привабливого та зручного для користувачів дизайну, але й забезпечення високої функціональності, ергономічності та відповідності технічним вимогам.

Проектування охоплює кілька важливих аспектів, таких як:

- Аналіз потреб користувачів: Це етап, на якому визначаються вимоги кінцевих користувачів системи, щоб продукт відповідав їхнім очікуванням і потребам.
- Визначення функціональності: На цьому етапі визначаються функції та можливості, які має надавати система для ефективного виконання завдань.
- Розробка структури продукту: Створюється архітектура системи, визначаються основні компоненти та взаємодія між ними.

Дизайнери, у свою чергу, працюють над концепцією інтерфейсів, визначенням кольорових схем, шрифтів і інших елементів візуального оформлення продукту. Це дозволяє створити інтерфейс, який буде не тільки естетично привабливим, але й зручним для користувачів.

### *Прототипування*

Прототипування є важливим етапом у розробці продукту, яке допомагає створити перші моделі або прототипи системи. Цей етап дозволяє:

1. Оцінити функціональність: Прототипи дають змогу перевірити, як працюють основні функції системи, перед початком повної розробки.

2. Виявити недоліки дизайну: На основі отриманих зворотних зв'язків можна вдосконалювати інтерфейс та функціональність продукту, адаптуючи його до потреб користувачів.
3. Зібрати зворотній зв'язок від користувачів: Прототипи допомагають команді проекту та замовникам зрозуміти, як система буде працювати на практиці та чи відповідає вона вимогам.

Завдяки прототипуванню, команда проекту має можливість на ранніх етапах тестувати концептуальні ідеї, оптимізувати продукт і вносити корективи до його фінальної версії.

#### **4.1. Вибір технологій та архітектури системи**

Проект iHR's, спрямований на автоматизацію управління кадровими ресурсами, вимагає ретельного підходу до вибору технологій та архітектури системи. Оскільки метою є створення зручної, безпечної та масштабованої системи для обробки персональних даних, ефективного управління кадрами та забезпечення високого рівня автоматизації, вибір правильних технологій і архітектури є критичним для успіху проекту.

##### **Архітектура системи**

Для реалізації проекту iHR's буде використана **багат шарова архітектура (N-Tier)**, яка забезпечує чіткий розподіл функцій між рівнями системи, підвищуючи її гнучкість, масштабованість і підтримуваність. Основні шари архітектури:

1. **Шар представлення (UI Layer)** — відповідає за взаємодію користувача із системою, включаючи інтерфейс адміністратора, HR-фахівця та співробітника.
2. **Шар бізнес-логіки (Business Logic Layer)** — реалізує правила бізнес-процесів, обробляє запити та відповідає за інтеграцію з іншими системами.
3. **Шар даних (Data Layer)** — забезпечує зберігання та обробку даних у реляційній базі даних.

Ця архітектура дозволяє легко додавати нові функції без значних змін у системі, забезпечує розподіл навантаження та спрощує масштабування.

## Вибір технологій

Для створення системи iHR's обрано сучасні, надійні та перевірені технології, які відповідають вимогам до безпеки, продуктивності та масштабованості.

### 1. Мова програмування

**Python** — використовуватиметься для розробки серверної частини (бекенду) завдяки його універсальності, високій продуктивності та великій кількості бібліотек для автоматизації та обробки даних.

### 2. Фреймворки та бібліотеки

**Django** — фреймворк для швидкої розробки бекенду з вбудованими інструментами для обробки даних, безпеки та управління API.

**React** — бібліотека для створення інтерактивного та динамічного користувацького інтерфейсу.

### 3. База даних

**PostgreSQL**. Реляційна база даних для зберігання структурованих даних, таких як профілі співробітників, історія відпусток, нарахування заробітної плати.

### 4. Інтеграція

**REST API** — для обміну даними між системою iHR's та іншими корпоративними платформами, такими як ERP або бухгалтерські системи.

### 5. Хмарні сервіси

**AWS** — для хостингу серверної частини системи, що забезпечить високу доступність та масштабованість.

### 6. Безпека

**OAuth 2.0** — для управління автентифікацією користувачів.

**SSL/TLS** — для шифрування з'єднань між клієнтами та сервером.

**JWT (JSON Web Tokens)** — для підтримки безпечних сесій.

### 7. Інструменти для розробки

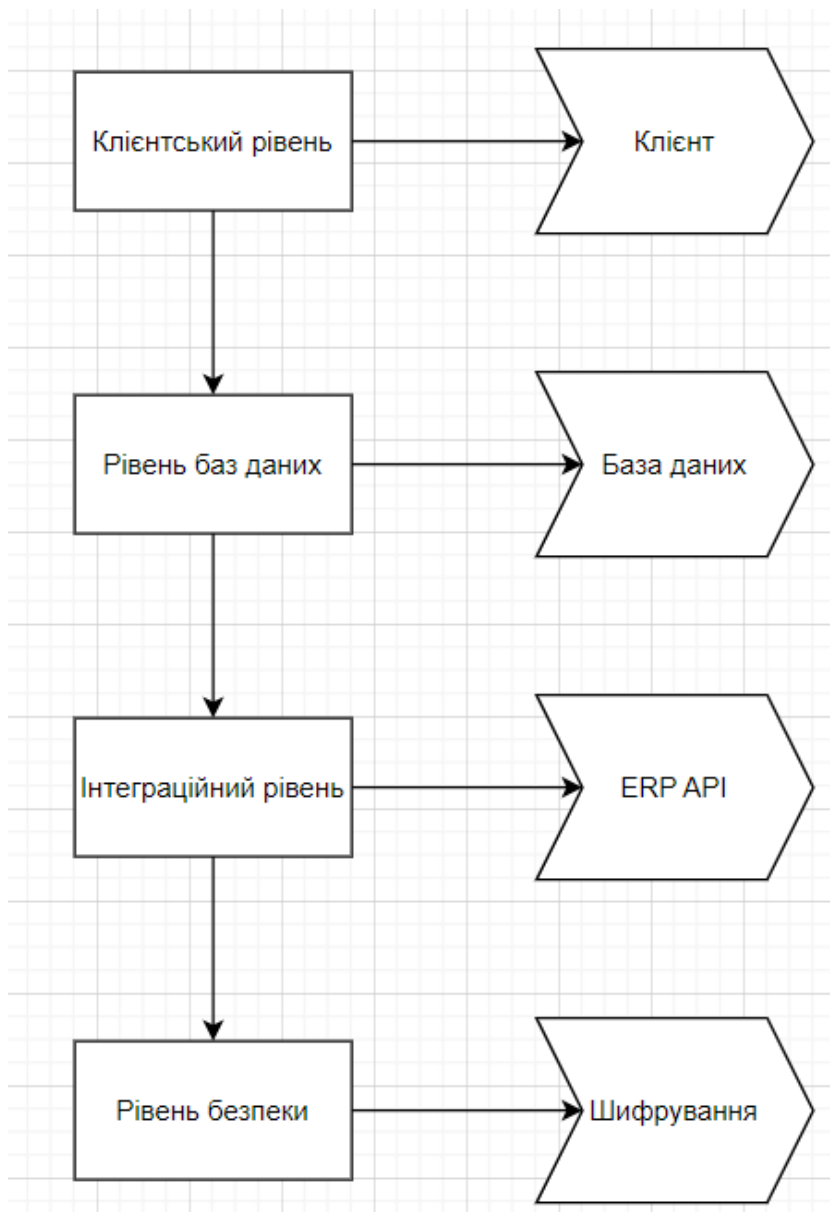
**Visual Studio Code** як основне середовище розробки для написання та налагодження коду.

**GitLab** — для управління версіями та командної співпраці.

## 8. Масштабованість

**Docker** — використовується для створення ізольованих середовищ, у яких працюють різні частини системи. Це дозволяє забезпечити стабільність роботи додатків і спростити їх перенесення між серверами.

**Kubernetes** — система управління та автоматизації роботи ізольованих середовищ, створених за допомогою Docker. Вона дозволяє автоматично розподіляти навантаження між компонентами системи та масштабувати їх залежно від потреб.



**Рис.4.1. Архітектура системи**

## 4.2 Розробка флоу системи

Перш ніж розробляти концептуальну модель, для мого проекту було розроблено флоу системи, яка має такий вигляд:

### Потік даних

#### 1. Процес найму:

-Кандидати додаються в таблицю Candidates.

-Кандидати подають заявки на певні позиції, створюються записи в таблиці Recruitment.

-Кандидати проходять інтерв'ю, статус оновлюється в таблиці Recruitment.

-Якщо кандидат прийнятий, його запис переноситься в таблицю Employees з посиланням на його CandidateID.

#### 2. Життєвий цикл співробітника:

-Після найму співробітника, його поточна інформація зберігається в таблиці Employees.

-При зміні позиції, департаменту або керівника створюється запис у таблиці EmployeeHistory.

-Звільнені співробітники переносяться в таблицю TerminatedEmployees.

#### 3. Історія змін

Всі зміни позицій, департаментів та керівників відображаються в таблиці EmployeeHistory.

#### 4. Процес звільнення

При звільненні співробітника, його запис переноситься в таблицю TerminatedEmployees з вказанням дати та причини звільнення.

Ця схема забезпечує повний контроль за рухом даних в системі, зберігаючи історію змін та забезпечуючи плавний перехід від кандидата до співробітника.

"Флоу" (від англ. "flow") в контексті інформаційних систем і баз даних означає потік або процес руху даних. Це опис того, як дані переміщуються між

різними частинами системи, які дії відбуваються з цими даними на кожному етапі, і як ці дані змінюються з часом. Флоу включає в себе наступні аспекти:

1. **Введення даних.** Як і де дані створюються або вводяться в систему.
2. **Обробка даних.** Які дії виконуються з даними після їх введення. Це може включати перевірку, модифікацію, розрахунки, збереження тощо.
3. **Переміщення даних.** Як дані переміщуються між різними модулями системи або між різними таблицями в базі даних.
4. **Збереження даних.** Де і як дані зберігаються, в яких таблицях і в якому форматі.
5. **Виведення даних.** Як дані витягуються з системи для перегляду або використання в інших процесах.
6. **Життєвий цикл даних.** Весь процес від створення до зберігання і видалення даних.

### **Приклад флоу для системи управління кадровими ресурсами:**

#### **1. Додавання нового кандидата**

- Введення. Кандидат подає заявку через онлайн форму.
- Обробка. Дані про кандидата (ім'я, контактна інформація, резюме) перевіряються на правильність і зберігаються в таблиці Candidates.
- Переміщення. Інформація про кандидата використовується для створення запису в таблиці Recruitment.

#### **2. Процес найму**

- Введення. HR проводить інтерв'ю і оновлює статус у таблиці Recruitment.
- Обробка. Після успішного інтерв'ю, статус кандидата змінюється на "Прийнятий".
- Переміщення. Дані про кандидата переміщуються в таблицю Employees, створюючи нового співробітника.

### **3. Управління співробітником**

- Обробка. При зміні позиції, департаменту або керівника створюється запис у таблиці EmployeeHistory.
- Збереження. Поточні дані про позицію, департамент і керівника зберігаються в таблиці Employees.

### **4. Звільнення співробітника**

Обробка. При звільненні співробітника його дані видаляються з таблиці Employees і додаються в таблицю TerminatedEmployees.

#### **Розробка флоу для системи**

Для системи управління кадровими ресурсами слід чітко описати всі ключові процеси, такі як:

1. Найм нового співробітника.
2. Зміна позиції або департаменту співробітника.
3. Процес звільнення співробітника.
4. Процес проходження тренінгів.
5. Оцінка продуктивності співробітників.

Опис кожного процесу має включати всі кроки від введення даних до їх зберігання і подальшого використання.

#### **Модулі**

1. Рекрутинг
2. Управління персоналом
3. Навчання
4. Оцінка продуктивності

#### **Флоу для кожного модуля**

##### **1. Модуль рекрутингу**

##### **Флоу:**

- Додавання кандидата
- Прийом заявок на позиції

- Процес інтерв'ю
- Прийняття рішення про найм

## **2. Модуль управління персоналом**

### **Флоу:**

- Додавання нового співробітника
- Зміна позиції, департаменту, менеджера
- Звільнення співробітника

## **3. Модуль навчання**

### **Флоу:**

- Планування тренінгів
- Призначення тренінгів співробітникам
- Завершення тренінгів

## **4. Модуль оцінки продуктивності**

### **Флоу:**

- Заплановані перегляди продуктивності
- Проведення перегляду
- Оцінка і коментарі

## **Повний Флоу для всієї системи**

### **1. Процес найму кандидата:**

- Дані про кандидата вводяться в таблицю Candidates.
- Заявка кандидата створює запис в таблиці Recruitment.
- Кандидат проходить інтерв'ю і статус оновлюється в таблиці Recruitment.
- Прийнятий кандидат переноситься в таблицю Employees, зберігаючи зв'язок з таблицею Candidates.

## **2. Процес управління співробітником:**

- При зміні позиції, департаменту або керівника створюється запис у таблиці EmployeeHistory.
- Поточні дані співробітника зберігаються в таблиці Employees.
- Звільнений співробітник переноситься в таблицю TerminatedEmployees.

## **3. Процес навчання:**

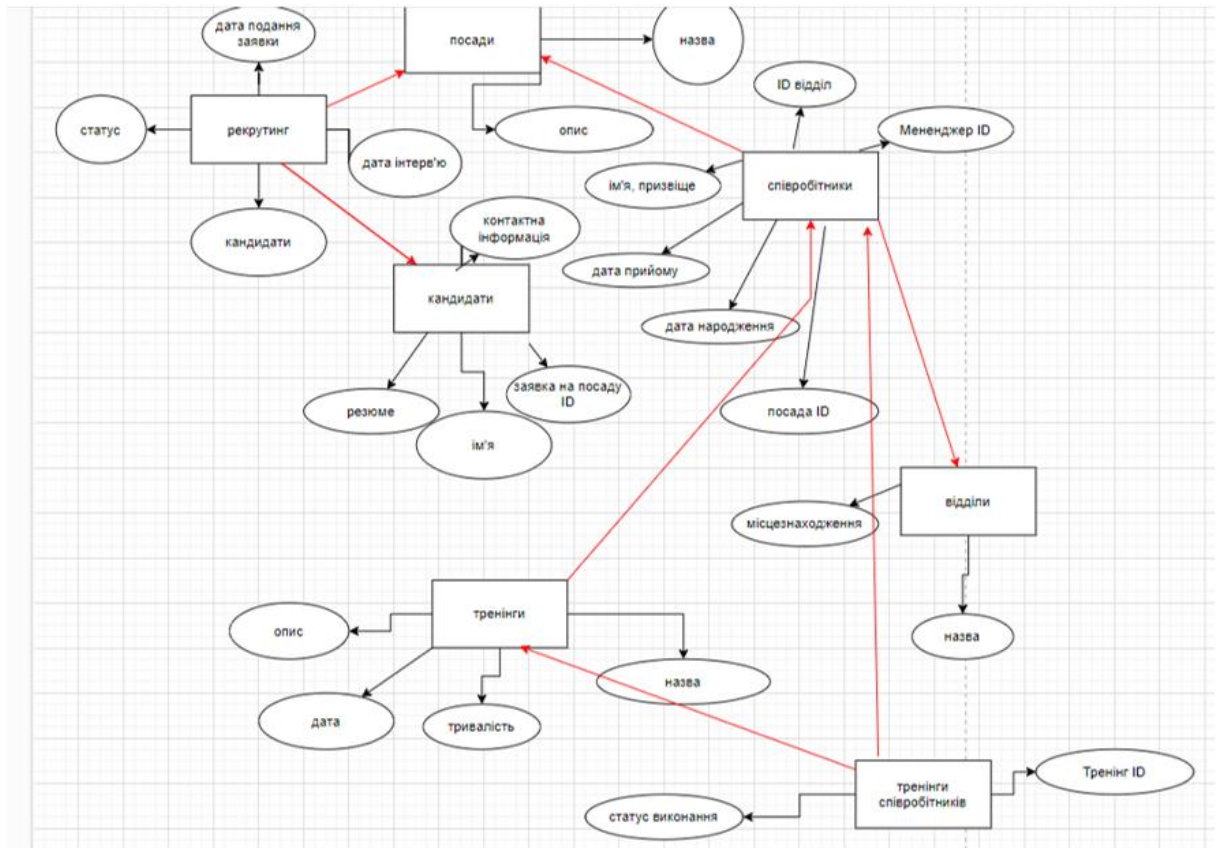
- Планування тренінгів зберігається в таблиці Trainings.
- Призначення тренінгів співробітникам створює записи в таблиці EmployeeTrainings.
- Завершення тренінгу оновлює статус у таблиці EmployeeTrainings.

## **4. Процес оцінки продуктивності:**

- Оцінка продуктивності співробітника створює запис у таблиці PerformanceReviews.

### **4.3. Розробка концептуальної моделі бази даних проєкту**

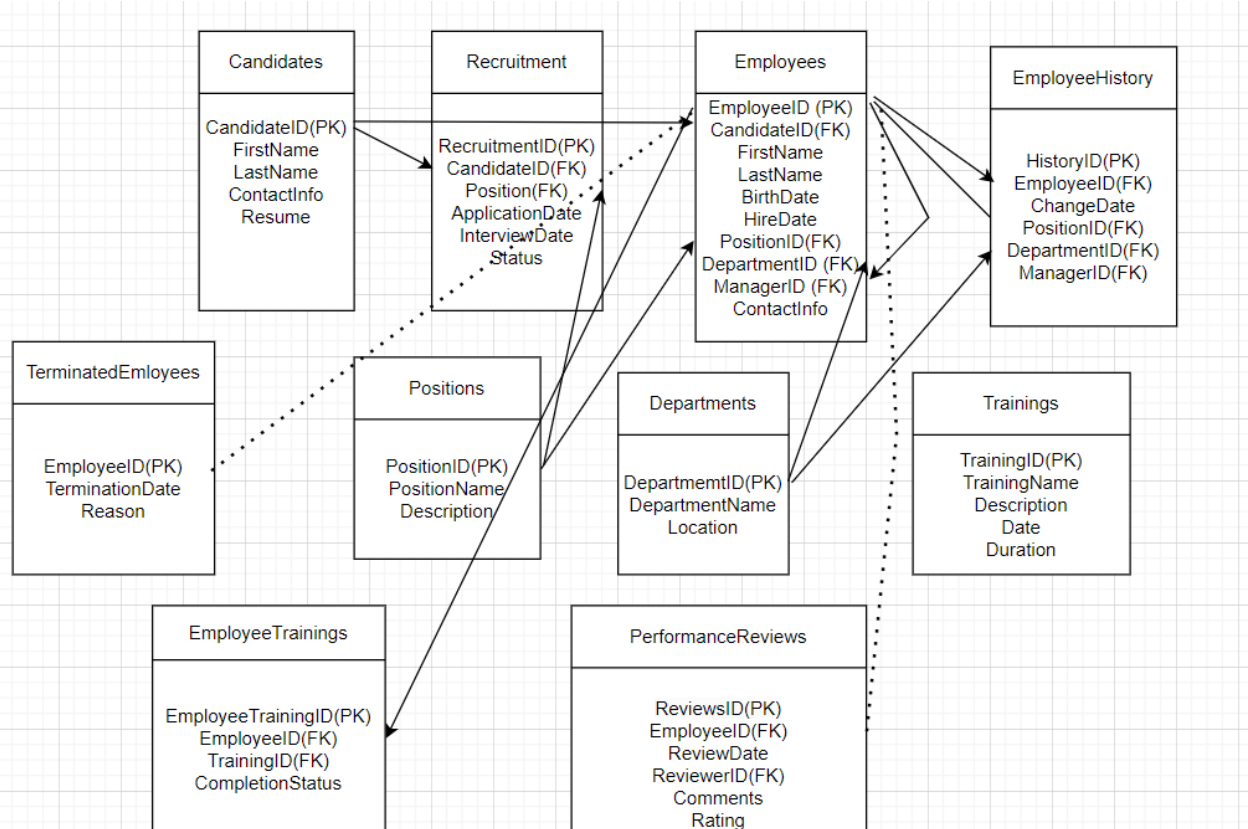
Для створення концептуальної моделі бази даних(рис.4.2) для проєкту автоматизованої системи управління кадровими ресурсами "iHR`s", необхідно визначити основні сутності (таблиці) та їх зв'язки. Концептуальна модель є високорівневим описом бази даних, що відображає основні об'єкти і їхні взаємозв'язки.



**Рис. 4.2. Концептуальна модель БД**

#### **4.4. Розробка логічної моделі бази даних**

Логічна модель даних (рис. 4.3) використовується для визначення структури елементів даних і встановлення зв'язків між ними. Логічна модель даних додає додаткову інформацію до елементів концептуальної моделі даних. Перевагою використання логічної моделі даних є надання а foundation створити основу для фізичної моделі. Однак структура моделювання залишається загальною [6].



**Рис. 4.3. Логічна модель БД**

### Визначення зв'язків

#### 1. Зв'язок між Candidates та Recruitment

- Candidates (PK: CandidateID) -> Recruitment (FK: CandidateID)
- Один кандидат може подавати багато заявок, але кожна заявка стосується одного кандидата.

#### 2. Зв'язок між Recruitment та Positions

- Positions (PK: PositionID) -> Recruitment (FK: PositionID)
- Кожна заявка стосується однієї позиції, але одна позиція може мати багато заявок.

#### 3. Зв'язок між Candidates та Employees

- Candidates (PK: CandidateID) -> Employees (FK: CandidateID)
- Один кандидат може стати співробітником (один запис у Employees на одного кандидата).

#### 4. Зв'язок між Employees та Positions

- Positions (PK: PositionID) -> Employees (FK: PositionID)

- Кожен співробітник має одну позицію, але одна позиція може мати багато співробітників.

#### **5. Зв'язок між Employees та Departments**

- Departments (PK: DepartmentID) -> Employees (FK: DepartmentID)
- Кожен співробітник належить до одного департаменту, але один департамент може мати багато співробітників.

#### **6. Зв'язок між Employees та Employees (керівник)**

- Employees (PK: EmployeeID) -> Employees (FK: ManagerID)
- Один співробітник може мати одного керівника, але один керівник може мати багато підлеглих.

#### **7. Зв'язок між Employees та EmployeeHistory**

- Employees (PK: EmployeeID) -> EmployeeHistory (FK: EmployeeID)
- Один співробітник може мати багато записів в історії, але кожен запис стосується одного співробітника.

#### **8. Зв'язок між Positions, Departments, Employees та EmployeeHistory**

- Positions (PK: PositionID) -> EmployeeHistory (FK: PositionID)
- Departments (PK: DepartmentID) -> EmployeeHistory (FK: DepartmentID)
- Employees (PK: EmployeeID) -> EmployeeHistory (FK: ManagerID)
- Один запис історії стосується однієї позиції, одного департаменту і одного керівника.

#### **9. Зв'язок між Employees та TerminatedEmployees**

- Employees (PK: EmployeeID) -> TerminatedEmployees (FK: EmployeeID)
- Один співробітник може мати один запис про звільнення.

#### **10. Зв'язок між Employees та EmployeeTrainings**

- Employees (PK: EmployeeID) -> EmployeeTrainings (FK: EmployeeID)
- Один співробітник може мати багато записів про тренінги.

#### **11. Зв'язок між Trainings та EmployeeTrainings**

- Trainings (PK: TrainingID) -> EmployeeTrainings (FK: TrainingID)
- Один тренінг може бути пройдено багатьма співробітниками.

## 12.Зв'язок між Employees та PerformanceReviews

- Employees (PK: EmployeeID) -> PerformanceReviews (FK: EmployeeID)
- Один співробітник може мати багато записів про оцінки продуктивності.

## 13.Зв'язок між Employees та PerformanceReviews (рецензент)

- Employees (PK: EmployeeID) -> PerformanceReviews (FK: ReviewerID)
- Один рецензент може мати багато оцінок, які він дав іншим співробітникам.

### Пояснення до діаграми

- Всі стрілочки виходять від зовнішніх ключів (FK) і вказують на первинні ключі (PK), що забезпечує цілісність даних і правильність зв'язків між таблицями.
- Основні таблиці Candidates, Employees, Positions, Departments, Trainings є центральними вузлами, від яких йдуть зв'язки до інших таблиць, таких як Recruitment, EmployeeTrainings, EmployeeHistory, PerformanceReviews, і TerminatedEmployees.

Далі була розроблена фізична база даних (рис.4.4, рис. 4.5, рис. 4.6, рис. 4.7, рис. 4.8, рис. 4.9)

Имя

- ▼ Таблицы (6)
  - > candidates
  - > employee\_trainings
  - > employees
  - > interviews
  - > positions
  - > trainings

**Рис.4.4. База даних (таблиці)**

Таблица: candidates         Filter in any column

	id	first_name	last_name	position_applied	application_date	interview_date	status
	Фи...	Фильтр	Фильтр	Фильтр	Фильтр	Фильтр	Фильтр
1	1	Alice	Brown	Engineer	2024-05-01	2024-05-15	Interview Scheduled
2	2	Bob	Davis	Manager	2024-05-03	2024-05-17	Application Received

**Рис.4.5. Інформація про кандидатів**

Таблица: employees         Filter in any column

	id	first_name	last_name	birth_date	address	marital_status	position	salary
	Фи...	Фильтр	Фильтр	Фильтр	Фильтр	Фильтр	Фильтр	Фильтр
1	1	John	Doe	1980-01-01	123 Main St	Married	Manager	60000.0
2	2	Jane	Smith	1990-02-14	456 Elm St	Single	Engineer	50000.0
3	3	Mike	Johnson	1985-03-20	789 Oak St	Married	Technician	45000.0

**Рис.4.6. Інформація про Співробітників**

Таблиця: interviews

	id	candidate_id	interview_date	interviewer	feedback	decision
	Фи...	Фильтр	Фильтр	Фильтр	Фильтр	Фильтр
1	1	1	2024-05-15	Jane Smith	Strong technical skills	Pending
2	2	2	2024-05-17	John Doe	Excellent leadership qualities	Pending

**Рис.4.7. Інформація про співбесіди**

Таблиця: positions

	id	position_name	department	requirements	status
	Фи...	Фильтр	Фильтр	Фильтр	Фильтр...
1	1	Engineer	Engineering	BSc in Engineering, 3 years experience	Open
2	2	Manager	Management	MBA, 5 years experience	Open

**Рис. 4.8. Інформація про посаду, досвід та статус**

Таблиця: trainings

	id	training_name	description	scheduled_date	duration
	Фи...	Фильтр	Фильтр	Фильтр	Фильтр
1	1	Python Basics	Introduction to Python programming	2024-06-01	2 days
2	2	Advanced Management	Advanced skills for management	2024-06-15	3 days

**Рис.4.9. Інформація про тренінги(опис, дата, тривалість навчання)**

#### 4.5. Опис архітектури програмного забезпечення

Розробка архітектури програмного забезпечення є критичним етапом у створенні інформаційної системи, яка відповідає вимогам бізнесу та забезпечує високу продуктивність, надійність та масштабованість. Архітектура програмного забезпечення проекту "iHR's" забезпечує інтеграцію різних компонентів системи, оптимізацію роботи та підтримку всіх необхідних функцій для ефективного управління кадровими ресурсами.

## Загальна архітектура системи

Архітектура програмного забезпечення проекту "iHR's" базується на багатошаровій (мультирівневій) моделі, яка включає такі основні шари:

- Клієнтський шар (Presentation Layer)
- Логічний шар (Business Logic Layer)
- Шар даних (Data Access Layer)
- Шар зберігання даних (Database Layer)

### Клієнтський шар

Клієнтський шар відповідає за взаємодію з користувачами системи та забезпечення інтуїтивно зрозумілого інтерфейсу. Цей шар включає:

- **Веб-інтерфейс.** Веб-додаток, який надає доступ до функціоналу системи через веб-браузери. Він розроблений з використанням сучасних веб-технологій, таких як HTML5, CSS3, та JavaScript, з фреймворками Angular або React.

### Логічний шар

Логічний шар обробляє бізнес-логіку та правила, які визначають функціонування системи. Він включає кілька модулів, кожен з яких реалізований у вигляді окремих мікросервісів:

- **Модулі рекрутингу.** Включають сервіси для обробки заявок кандидатів, планування та проведення інтерв'ю, оцінки кандидатів. Ці сервіси можуть взаємодіяти з зовнішніми рекрутинговими платформами через API.
- **Модулі управління персоналом.** Включають сервіси для управління інформацією про співробітників, відстеження кар'єрного росту, оцінки продуктивності. Реалізовані як RESTful сервіси, що надають функції CRUD (Create, Read, Update, Delete) для управління даними співробітників.
- **Модулі навчання та розвитку.** Включають сервіси для планування та відстеження навчальних заходів, управління тренінгами. Взаємодіють з платформами навчання для імпорту та експорту даних про тренінги та курси.

- **Модулі оцінки ефективності.** Включають сервіси для проведення оцінок роботи співробітників, збирання зворотного зв'язку та аналітики. Модулі можуть генерувати звіти про продуктивність та інтегруватися з аналітичними інструментами.

### **Шар даних**

Шар даних відповідає за доступ до даних і забезпечення безпечної та ефективної роботи з базами даних. Він включає:

- **Репозиторій даних.** Інтерфейс для доступу до бази даних, який приховує складність взаємодії з даними від вищих шарів системи. Реалізовано за допомогою ORM (Object-Relational Mapping) таких як Hibernate або Entity Framework.
- **Сервіси даних.** API для обміну даними між логічним шаром і шаром зберігання даних. Ці сервіси інкапсулюють доступ до бази даних і забезпечують виконання бізнес-логіки.

### **Шар зберігання даних**

Шар зберігання даних відповідає за фізичне зберігання даних та забезпечення їхньої цілісності. Він включає:

- **Реляційна база даних.** Зберігання структурованих даних, таких як інформація про співробітників, вакансії, тренінги. Використання СУБД, таких як MySQL або PostgreSQL.
- **Хмарне сховище.** Зберігання великих об'ємів даних, таких як резюме кандидатів, документи співробітників, навчальні матеріали. Може бути реалізоване за допомогою AWS S3, Google Cloud Storage або Azure Blob Storage.
- **Резервне копіювання та відновлення.** Механізми для резервного копіювання даних та відновлення їх у разі збою. Використання рішень для автоматизованого резервного копіювання та регулярного тестування відновлення.

### 4.5.1. Взаємодія компонентів

Компоненти системи взаємодіють між собою через визначені інтерфейси та API. Взаємодія між клієнтським шаром та логічним шаром здійснюється через RESTful API або GraphQL, що забезпечує зручність та ефективність обміну даними. Логічний шар взаємодіє з шаром даних через сервіси даних, які інкапсулюють доступ до бази даних.

### 4.5.2. Безпека та управління доступом

Безпека системи забезпечується за рахунок використання сучасних методів аутентифікації та авторизації, шифрування даних та контрольованого доступу до ресурсів. Використання протоколів OAuth2 та JWT для аутентифікації користувачів, а також ролей та привілеїв для контролю доступу до функціоналу системи[26].

### 4.5.3. Масштабованість та продуктивність

Архітектура системи розроблена з урахуванням вимог до масштабованості та продуктивності. Використання мікросервісної архітектури дозволяє масштабувати окремі компоненти системи незалежно один від одного. Хмарні технології забезпечують можливість динамічного розширення ресурсів у разі зростання навантаження.

Архітектура програмного забезпечення проекту "iHR's" спрямована на забезпечення гнучкості, надійності та ефективності системи, що дозволяє ефективно вирішувати завдання управління кадровими ресурсами компанії.

## Опис компонентів (рис. 4.10)

### Клієнтський шар

- **Веб-інтерфейс:** забезпечує доступ до системи через веб-браузери, може бути розроблений на Angular або React. Інтерфейс включає:

**Панель керування:** основний інтерфейс для адміністраторів та менеджерів, що дозволяє керувати всіма аспектами системи.

**Форма заявок:** для кандидатів, що дозволяє подавати заявки на вакансії.

**Особистий кабінет співробітника:** інтерфейс для доступу до персональних даних, перегляду історії кар'єри, планування навчання та перегляду оцінок продуктивності.

## Логічний шар

Логічний шар обробляє всю бізнес-логіку системи. Він складається з кількох основних модулів, кожен з яких реалізований у вигляді мікросервісу:

- **Модулі рекрутингу**

**Сервіс обробки заявок кандидатів:** обробляє заявки кандидатів, перевіряє їх на відповідність вимогам.

**Сервіс планування інтерв'ю:** дозволяє рекрутерам планувати інтерв'ю з кандидатами.

**Сервіс оцінки кандидатів:** забезпечує можливість оцінки кандидатів та зберігання результатів інтерв'ю.

- **Модулі управління персоналом**

**Сервіс управління інформацією про співробітників:** забезпечує можливість створення, читання, оновлення та видалення інформації про співробітників.

**Сервіс відстеження кар'єрного росту:** дозволяє відстежувати кар'єрний ріст співробітників та планувати їхній розвиток.

**Сервіс оцінки продуктивності:** проводить оцінки продуктивності співробітників та зберігає результати.

- **Модулі навчання та розвитку**

**Сервіс планування навчальних заходів:** забезпечує планування та організацію навчальних заходів для співробітників.

**Сервіс управління тренінгами:** забезпечує можливість створення, оновлення та видалення тренінгів та курсів.

- **Модулі оцінки ефективності:**

**Сервіс проведення оцінок:** забезпечує проведення оцінок роботи співробітників.

**Сервіс збору зворотного зв'язку:** забезпечує збір зворотного зв'язку від співробітників.

**Сервіс аналітики:** забезпечує аналітику та генерацію звітів на основі зібраних даних.

## Шар даних

- **Репозиторій даних:** забезпечує інтерфейс для доступу до бази даних, приховуючи складність взаємодії з даними від вищих шарів системи. Реалізований за допомогою ORM, таких як Hibernate або Entity Framework.
- **Сервіси даних:** API для обміну даними між логічним шаром і шаром зберігання даних. Ці сервіси інкапсулюють доступ до бази даних і забезпечують виконання бізнес-логіки.

## Шар зберігання даних

- **Реляційна база даних:** використовується для зберігання структурованих даних, таких як інформація про співробітників, вакансії, тренінги. Використовуються СУБД, такі як MySQL або PostgreSQL.
- **Хмарне сховище:** використовується для зберігання великих об'ємів даних, таких як резюме кандидатів, документи співробітників, навчальні матеріали. Використовуються рішення, такі як AWS S3, Google Cloud Storage або Azure Blob Storage.
- **Резервне копіювання та відновлення:** забезпечує резервне копіювання даних та їх відновлення у разі збою. Використовуються автоматизовані рішення для резервного копіювання та регулярного тестування відновлення.

## External Services

- **Інтеграція з рекрутинговими платформами:** забезпечує взаємодію з зовнішніми рекрутинговими платформами для обробки заявок кандидатів.
- **Інтеграція з навчальними платформами:** забезпечує взаємодію з зовнішніми навчальними платформами для імпорту та експорту даних про тренінги та курси.

- **Інтеграція з сервісами для оцінки ефективності:** забезпечує взаємодію з зовнішніми сервісами для проведення оцінок ефективності співробітників.

## Security Layer

- **Аутентифікація і авторизація користувачів:** використовуються протоколи OAuth2 та JWT для забезпечення безпеки аутентифікації користувачів.
- **Захист даних:** забезпечується шифруванням даних як у стані спокою, так і під час передачі. Використовуються сучасні методи шифрування для захисту конфіденційної інформації.
- **Контроль доступу:** реалізовано через ролі та привілеї для контролю доступу до функціоналу системи.

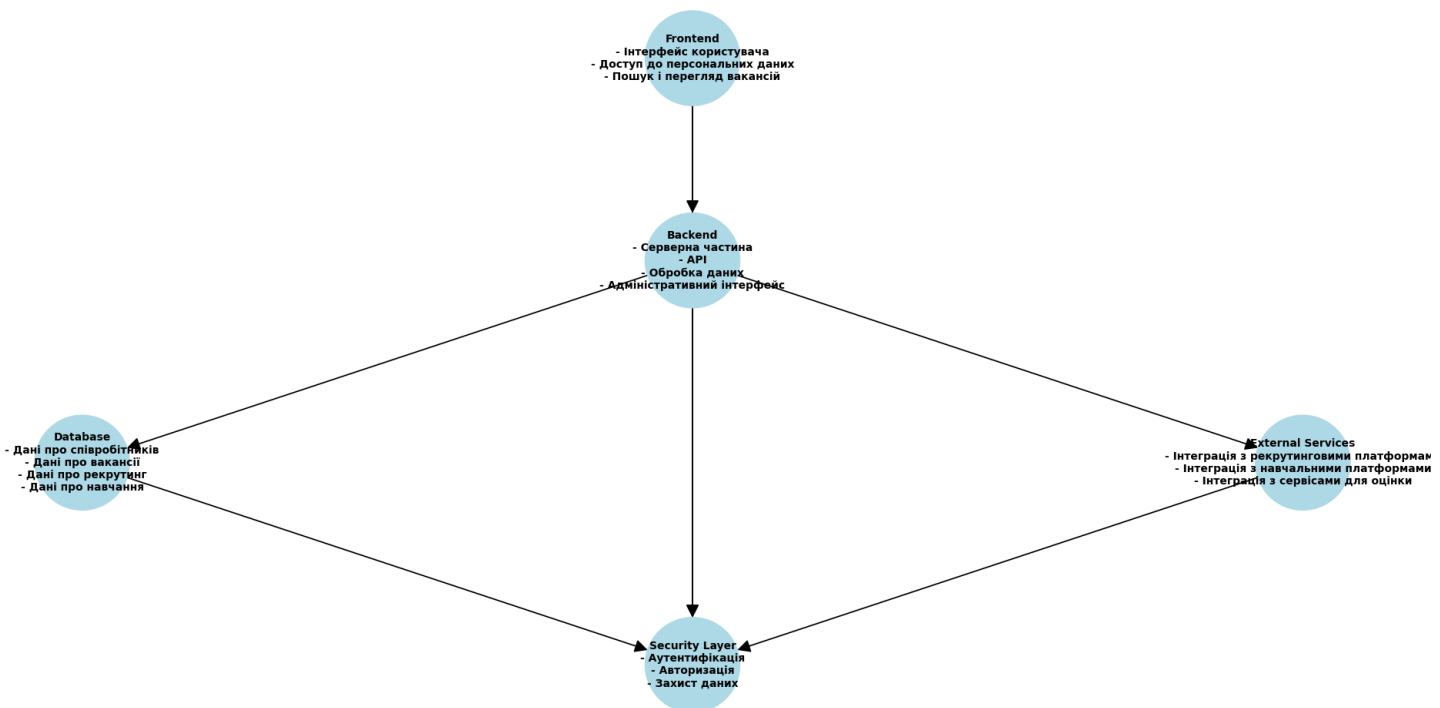


Рис. 4.10. Схема архітектури програмного забезпечення

## **4.6. Розробка алгоритмів та інтерфейсів програмного забезпечення**

### **Алгоритм автоматичного виявлення кандидатів**

Опис: цей алгоритм буде використовуватись для автоматичного аналізу резюме кандидатів та виявлення найбільш підходящих для певної вакансії.

Кроки:

Зчитування резюме кандидатів у текстовому форматі.

Використання методів обробки природної мови для аналізу ключових слів та фраз, що вказують на навички, досвід та освіту кандидата.

Порівняння аналізованих даних з вимогами до вакансії та призначення кожного кандидата певному рівню відповідності.

Створення звіту з найкращими кандидатами для подальшого розгляду HR-фахівцями.

### **Інтерфейс керування відпустками**

Цей інтерфейс дозволить співробітникам подавати запити на відпустки та HR-фахівцям керувати цими запитами.

Основні функції:

Форма подання запиту на відпустку, де співробітник може вказати дату початку та кінця відпустки, тип відпустки тощо.

Панель адміністратора для перегляду та управління всіма запитами на відпустку.

Можливість затвердження або відхилення запитів з боку HR-фахівця.

Система повідомлень для співробітників щодо статусу їхніх запитів.

### **Алгоритм автоматизованої генерації звітів**

Цей алгоритм буде автоматично генерувати звіти з різних аспектів управління кадровими ресурсами.

Кроки:

Вибір типу звіту (наприклад, аналіз відпусток за місяць або рік, огляд найкращих кандидатів на певну вакансію тощо).

Збір даних з бази даних системи про відповідну тему звіту.

Аналіз та обробка даних для підготовки графіків, діаграм, таблиць та інших візуальних елементів.

Форматування та генерація остаточного звіту у зручному для користувача форматі (наприклад, PDF або Excel).

Опціонально: автоматичне розсилання звіту за встановленою розкладом.

### **Алгоритм рекомендацій для розвитку персоналу**

Цей алгоритм аналізує навички, досвід та освіту співробітників, а також їхні цілі та інтереси, і надає рекомендації щодо навчання та розвитку для кожного працівника.

Кроки:

Збір даних про навички, досвід, освіту та професійні цілі співробітників.

Аналіз цих даних для ідентифікації прогалин у знаннях та навичках.

Вибір навчальних матеріалів, курсів або тренінгів, які найкраще відповідають потребам кожного співробітника.

Створення індивідуалізованих рекомендаційних планів для кожного працівника.

Моніторинг прогресу та адаптація рекомендаційних планів з часом.

### **Алгоритм аналізу задоволеності працівників**

Алгоритм оцінює задоволеність працівників різними аспектами роботи, такими як рівень заробітної плати, розвиток кар'єри, робоче середовище тощо, і надає рекомендації для поліпшення робочих умов.

Кроки:

Проведення анонімного опитування серед працівників з різних аспектів їхнього досвіду роботи.

Аналіз результатів опитування для ідентифікації проблемних аспектів та тенденцій.

Вибір конкретних заходів для поліпшення задоволеності працівників на основі аналізу.

Впровадження вибраних заходів та моніторинг їх впливу на задоволеність працівників.

Повторення опитувань та аналіз результатів для постійного вдосконалення робочих умов.

## **Алгоритм автоматизованого аналізу витрат на персонал**

Алгоритм дозволяє автоматично аналізувати витрати на персонал, включаючи заробітну плату, витрати на найм та звільнення, бонуси тощо, для ефективного управління бюджетом компанії.

Кроки:

Збір даних про витрати на персонал з різних джерел, таких як бухгалтерська система та система управління персоналом.

Категоризація цих витрат за типами (наприклад, заробітна плата, податки, страхування, бонуси).

Аналіз витрат та їх динаміки з часом для ідентифікації тенденцій та можливих витратних резервів.

Генерація звітів та рекомендацій щодо оптимізації витрат на персонал.

Моніторинг впровадження рекомендацій та їх впливу на бюджет компанії.

Розробка інтерфейсу програмного забезпечення (UI) - це важлива частина процесу створення програмного продукту, оскільки вона визначає, як користувачі будуть взаємодіяти з додатком. Ось кілька кроків, які можна виконати для розробки ефективного інтерфейсу програмного забезпечення:

**Визначення вимог користувача:** вивчення потреб та очікувань цільової аудиторії. Розуміння, як користувачі будуть використовувати продукт, допоможе визначити ключову функціональність та інтерфейсні елементи.

**Створення скетчів та макетів:** створити прототипи інтерфейсу, які відображають основні функції та елементи додатка. Використання програм для створення макетів, таких як Adobe XD, Sketch або Figma

**Дизайн інтерфейсу:** розробити дизайн інтерфейсу, який відповідає бренду та враховує принципи дизайну користувацького інтерфейсу (UI Design Principles), такі як простота, зручність використання та консистентність.

**Розробка інтерфейсу:** використати фреймворки або бібліотеки для розробки фронтенду (наприклад, React.js, Angular або Vue.js), щоб швидко реалізувати дизайн інтерфейсу.

**Тестування і зміни:** провести тестування інтерфейсу з метою перевірки зручності використання та відповідності вимогам користувачів. За необхідності внести зміни до дизайну та інтерфейсу відповідно до результатів тестування.

**Документація інтерфейсу:** забезпечення належної документації для інтерфейсу, включаючи опис кожного елемента інтерфейсу та інструкції з використання.

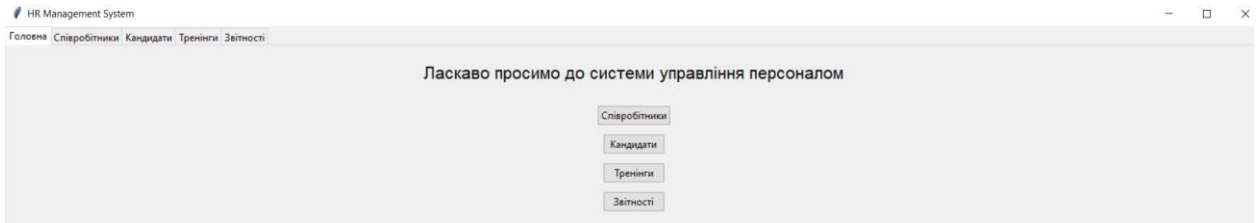
**Підтримка і вдосконалення:** надати можливість користувачам звертатися зі своїми пропозиціями та відгуками щодо інтерфейсу. Враховувати ці звернення при плануванні майбутніх оновлень та вдосконалень інтерфейсу

Початковий інтерфейс системи управління кадровими ресурсами був розроблений мовою програмування Python. (рис.4.6, 4.7, 4.8, 4.9, 4.10, 4.11, )

HR Management System

First Name	<input type="text"/>
Last Name	<input type="text"/>
Birth Date (YYYY-MM-DD)	<input type="text"/>
Address	<input type="text"/>
Marital Status	<input type="text"/>
Position	<input type="text"/>
Salary	<input type="text"/>
	<input type="button" value="Add Employee"/>
Employee ID	<input type="text"/>
Date (YYYY-MM-DD)	<input type="text"/>
Hours Worked	<input type="text"/>
	<input type="button" value="Add Work Hours"/>
Employee ID	<input type="text"/>
	<input type="button" value="Delete Employee"/>
Candidate First Name	<input type="text"/>
Candidate Last Name	<input type="text"/>
Candidate Birth Date (YYYY-MM-DD)	<input type="text"/>
Candidate Address	<input type="text"/>
Candidate Marital Status	<input type="text"/>
Position Applied	<input type="text"/>
Resume	<input type="text"/>
	<input type="button" value="Add Candidate"/>
Candidate ID	<input type="text"/>
Position	<input type="text"/>
Application Date (YYYY-MM-DD)	<input type="text"/>
Status	<input type="text"/>
	<input type="button" value="Add Job Application"/>
Application ID	<input type="text"/>
Interview Date (YYYY-MM-DD)	<input type="text"/>
Interviewer	<input type="text"/>
Feedback	<input type="text"/>
Result	<input type="text"/>
	<input type="button" value="Add Interview"/>
Employee ID	<input type="text"/>
Training Name	<input type="text"/>
Training Date (YYYY-MM-DD)	<input type="text"/>
Status	<input type="text"/>

**Рис. 4.11.** Інтерфейс для додавання співробітників, кандидатів, інтерв'ю, робочих годин.



**Рис. 4.12. Початкова сторінка системи**

HR Management System

Employees Candidates Trainings

ID	First Name	Last Name	Birth Date	Address	Marital Status	Position	Salary
1	John	Doe	1980-01-01	123 Main St	Married	Manager	60000.0
2	Jane	Smith	1990-02-14	456 Elm St	Single	Engineer	50000.0
3	Mike	Johnson	1985-03-20	789 Oak St	Married	Technician	45000.0

**Рис. 4.13. Список співробітників та інформація про них**

HR Management System

Employees Candidates Trainings

ID	First Name	Last Name	Position Applied	Application Date	Interview Date	Status
1	Alice	Brown	Engineer	2024-05-01	2024-05-15	Interview Scheduled
2	Bob	Davis	Manager	2024-05-03	2024-05-17	Application Received

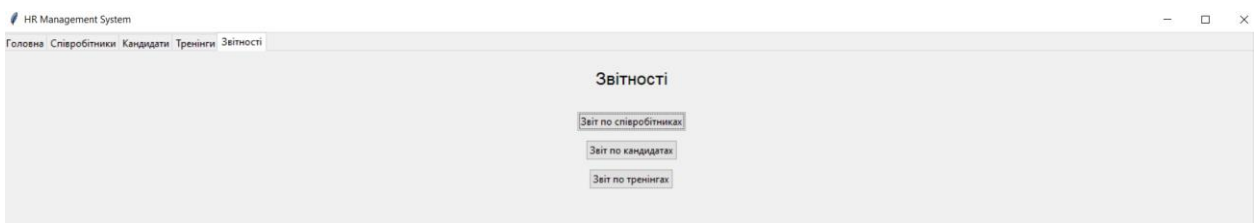
**Рис. 4.14. Список кандидатів та інформація про них**

HR Management System

Employees Candidates Trainings

ID	Training Name	Description	Scheduled Date	Duration
1	Python Basics	Introduction to Python programming	2024-06-01	2 days
2	Advanced Management	Advanced skills for management	2024-06-15	3 days

**Рис. 4.15. Список тренінгів та їх опис(назва, пояснення, дата, тривалість, статус)**



**Рис. 4.16. Вікно звітності**

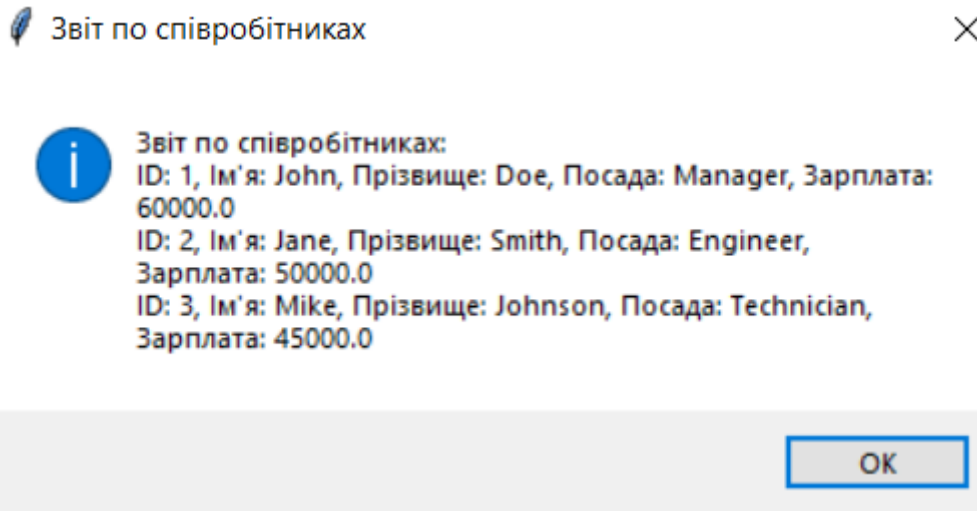


Рис. 4.17. Вікно «Звіт по співробітниках»

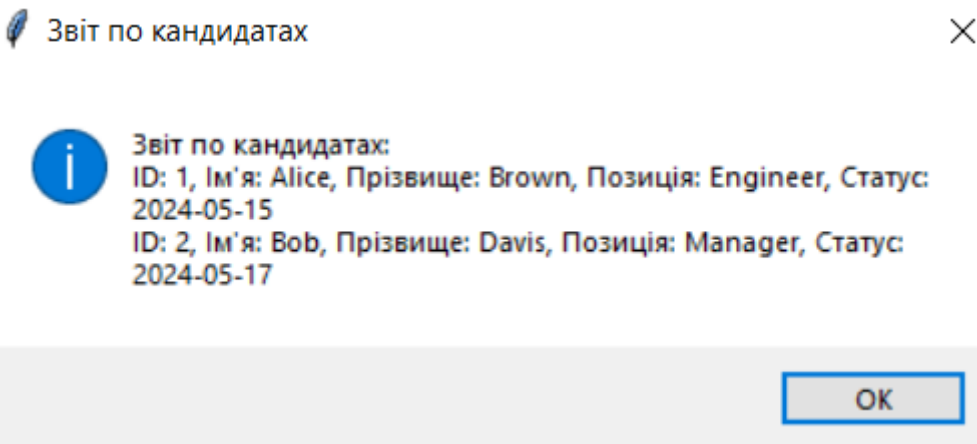


Рис. 4.18. Вікно «Звіт по кандидатах»



Рис. 4.19. Вікно «Звіт по тренінгах»

## ВИСНОВОК

У межах виконання роботи проведено комплексний аналіз предметної галузі, сучасних тенденцій розвитку HR-технологій та існуючих рішень для автоматизації HR-процесів. Виявлено основні проблеми управління персоналом, зокрема, високу трудомісткість рутинних завдань, труднощі інтеграції існуючих систем і недостатній рівень автоматизації.

Розроблено концептуальну та математичну моделі системи автоматизації управління кадровими ресурсами, які забезпечують інтеграцію ключових функцій управління персоналом, таких як облік співробітників, управління відпустками, розрахунок заробітної плати та генерація звітів.

Обґрунтовано вибір методології управління проектом, яка базується на сучасних підходах (Agile, Scrum). Це дозволило забезпечити гнучкість у процесі розробки системи, швидке реагування на зміни вимог та оптимальне використання ресурсів.

Розроблено організаційну структуру команди проекту, що включала ключові ролі для успішної реалізації завдань. Виконано декомпозицію задач за допомогою WBS, ідентифіковано ризики на всіх етапах проекту та розроблено ефективні стратегії їхнього управління.

Створено концептуальну модель бази даних, яка забезпечує централізоване зберігання та обробку кадрових даних, а також реалізовано інтуїтивно зрозумілі інтерфейси для HR-спеціалістів і співробітників.

Оцінено ефективність запропонованого рішення. Впровадження системи дозволяє суттєво скоротити адміністративне навантаження, мінімізувати кількість помилок і підвищити швидкість обробки даних. Розроблено рекомендації щодо інтеграції системи в організаційні процеси, включаючи технічну підтримку, масштабування та подальше вдосконалення.

Особлива увага приділена забезпеченню відповідності системи сучасним вимогам безпеки даних, включаючи дотримання стандартів GDPR. Реалізовано механізми шифрування даних, багаторівневу авторизацію користувачів і контроль доступу до конфіденційної інформації.

Загалом, виконана робота демонструє можливості автоматизованої системи «iHRs» для оптимізації кадрових процесів, підвищення ефективності управління персоналом і забезпечення безпеки даних, що є важливим для досягнення стратегічних цілей організацій.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Азбука управління проєктами. Планування / О.В.Єгорченков, Н.Ю.Єгорченкова, Є.Ю.Катаєва. — Київ, 2016. — 226 стр
2. Стаття, скорочена версія вебінару з приводу Управління проєктами [Електронний ресурс] – 2021 – Режим доступу до ресурсу: <https://www.manage.gov.in/studymaterial/PM.pdf>
3. "Проєктне управління в інформаційних технологіях" - Гребенніков Д.В. - Київ, 2017.
4. "Ефективний проєкт: від ідеї до реалізації" - Коваль О.В., Мельник О.П. - Львів, 2016.
5. "Інформаційні технології в управлінні проєктами" - Мельничук О.М. - Київ, 2015.
6. Морозов В.В., Чумаченко І.В., Доценко Н.В., Чередніченко А.М. Управління проєктами: планування проєктних дій: Підручник, – К.: Університет економіки та права «КРОК», 2014.
7. "Проєктний менеджмент в сучасних умовах" - Сидоренко В.П. - Харків, 2018.
8. "Моделі та методи управління ризиками проєктів" - Григор'єва І.В. - Київ, 2014.
9. "ІТ-проєкти: організація та управління" - Коваленко І.І. - Харків, 2016.
10. Ю. Ю. Нестюк АНАЛІЗ НЕДОЛІКІВ ТА ПЕРЕВАГ ІСНУЮЧИХ АВТОМАТИЗОВАНИХ СИСТЕМ УПРАВЛІННЯ ПЕРСОНАЛОМ Вінницький національний технічний університет.
11. BambooHR — <https://www.bamboohr.com>
12. Human Resource Information Systems: Basics, Applications, and Future Directions (Michael J. Kavanagh, Mohan Thite) — <https://dokumen.pub/human-resource-information-systems-basics-applications-and-future-directions-4nbsped-9781506351452-150635145x-9781506386539-1506386539.html>
13. Терещенко О. О. *Інформаційні системи в управлінні підприємством*. Київ: КНЕУ, 2019.

14. Smith J., Brown L. *Trends in HR Software Automation. Journal of Management Information Systems*, 2020.
15. Ivanova I. *Security in HR Automation Systems. Cybersecurity Journal*, 2021.
16. HCM Cloud від Oracle: <https://www.oracle.com/hcm>
17. Deloitte. *Global Human Capital Trends*. 2022.
18. Gartner. *Magic Quadrant for HCM Suites for Enterprises*. 2023.
19. Дьогтева І.О., Рудь Л.І. *Автоматизація HR-процесів в HRM-системах*. Репозитарій Вінницького національного технічного університету.— <https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/33964/89103.pdf?sequence=2>
20. *Цифровізація бізнес-процесів менеджменту персоналу: можливості HRM-систем*. Галицький економічний вісник, №1 (2022). — [https://elartu.tntu.edu.ua/bitstream/lib/38690/2/GEB\\_2022v74n1\\_Varis\\_I-Hr\\_management\\_business\\_processes\\_90-102.pdf](https://elartu.tntu.edu.ua/bitstream/lib/38690/2/GEB_2022v74n1_Varis_I-Hr_management_business_processes_90-102.pdf)
21. Рудакова С.Г., Щетініна Л.В. *Digital HR: Майбутнє кадрового адміністрування*. Бізнес Інформ, 2020.
22. Куйбіда В., Петроє О.М., Федулова Л.І. *Цифрові компетенції як умова формування якості людського капіталу*. Київ: НАДУ, 2019.
23. Ситник Й. С. *Інтелектуалізація систем менеджменту підприємств*. Львів: Видавництво Львівської політехніки, 2017.
24. Agile Manifesto (<https://agilemanifesto.org>) — для опису методології Agile.
25. Scrum.org — для специфіки методології Scrum.
26. Ernst & Young (EY). *The New Age: Artificial Intelligence for HR Functions*. — [https://assets.ey.com/content/dam/ey-sites/ey-com/en\\_gl/topics/alliances/ey-the-new-age-artificial-intelligence-for-human-resources-010978-18gbl.pdf](https://assets.ey.com/content/dam/ey-sites/ey-com/en_gl/topics/alliances/ey-the-new-age-artificial-intelligence-for-human-resources-010978-18gbl.pdf)
27. Піддубна Л., Чуєва І. Міжнародний досвід використання цифрових технологій в управлінні персоналом ІТ компаній. Журнал № 55(2023): Економіка та суспільство.
28. Worklviv. [Work.lviv.ua](http://Work.lviv.ua) - Топ-6 HRM-систем: як правильно автоматизувати процес HR.

29. Писаревська Г.І. Тенденції розвитку використання Digital технологій в управлінні персоналом №41(2021): Вісник ХДУ Серія Економічні науки.
30. Кобець Д. HR-аналітика і Big Data як інструменти цифрового менеджменту людських ресурсів № 3(2024): Вісник Хмельницького національного університету.
31. GO IT ( goit.global) – IT статті – Хто такий HR-менеджер і чим він займається.(27.04.2024)
32. Федорова Ю. Інноваційні інформаційні технології у підготовці та управлінні персоналом. DOI: [https://doi.org/10.33296/2707-0654-11\(22\)-11](https://doi.org/10.33296/2707-0654-11(22)-11) (accessed September 6, 2023).
33. Електронний ресурс. Django — <https://www.djangoproject.com>
34. Електронний ресурс Python — <https://avada-media.ua/services/python-plyusy-i-minusy-yazyka-kakiye-zadachi-reshayet-i-stoit-li-izuchat/>
35. Електронний ресурс. PyCharm — <https://www.jetbrains.com/pycharm/>
36. Електронний ресурс — <https://foxminded.ua/angular-vs-react/>
37. Електронний ресурс PostgreSQL — <https://www.postgresql.org>
38. Електронний ресурс AWS / Microsoft Azure — <https://www.softwareone.com/uk-ua/blog/articles/2024/05/30/microsoft-azure-vs-amazon-web-services>
39. Електронний ресурс. REST API — <https://foxminded.ua/shcho-take-rest-api/>
40. Електронний ресурс. OAuth 2.0 — <https://stfalcon.com/uk/blog/post/oauth-2.0>
41. Електронний ресурс. JWT (JSON Web Tokens) — <https://devzone.org.ua/post/iak-vykorystovuvaty-json-web-tokens-jwt-dlia-avtentyfikatsiyi>
42. "Проектний підхід у розробці програмного забезпечення" - Шевченко В.М. - Київ, 2015.

## ДОДАТОК А

Скрипт створення БД:

```
import sqlite3

def create_db():
    conn = sqlite3.connect('hr_system.db')
    cursor = conn.cursor()

    cursor.execute("""
CREATE TABLE IF NOT EXISTS employees (
    id INTEGER PRIMARY KEY,
    first_name TEXT,
    last_name TEXT,
    birth_date TEXT,
    address TEXT,
    marital_status TEXT,
    position TEXT,
    salary REAL
)
""")

    cursor.execute("""
CREATE TABLE IF NOT EXISTS candidates (
    id INTEGER PRIMARY KEY,
    first_name TEXT,
    last_name TEXT,
    position_applied TEXT,
    application_date TEXT,
    interview_date TEXT,
    status TEXT
)
""")

    cursor.execute("""
CREATE TABLE IF NOT EXISTS positions (
    id INTEGER PRIMARY KEY,
    position_name TEXT,
    department TEXT,
    requirements TEXT,
    status TEXT
)
""")

    cursor.execute("""
CREATE TABLE IF NOT EXISTS interviews (
    id INTEGER PRIMARY KEY,
    candidate_id INTEGER,
    interview_date TEXT,
    interviewer TEXT,
    feedback TEXT,
    decision TEXT,
    FOREIGN KEY(candidate_id) REFERENCES candidates(id)
)
""")
```

```
"""
```

```
cursor.execute("""
CREATE TABLE IF NOT EXISTS trainings (
    id INTEGER PRIMARY KEY,
    training_name TEXT,
    description TEXT,
    scheduled_date TEXT,
    duration TEXT
)
""")
```

```
cursor.execute("""
CREATE TABLE IF NOT EXISTS employee_trainings (
    id INTEGER PRIMARY KEY,
    employee_id INTEGER,
    training_id INTEGER,
    completion_date TEXT,
    FOREIGN KEY(employee_id) REFERENCES employees(id),
    FOREIGN KEY(training_id) REFERENCES trainings(id)
)
""")
```

```
conn.commit()
conn.close()
```

```
create_db()
```

Скрипт наповнення БД:

```
def populate_db():
```

```
    conn = sqlite3.connect('hr_system.db')
```

```
    cursor = conn.cursor()
```

```
    employees = [
```

```
        ('John', 'Doe', '1980-01-01', '123 Main St', 'Married', 'Manager', 60000),
```

```
        ('Jane', 'Smith', '1990-02-14', '456 Elm St', 'Single', 'Engineer', 50000),
```

```
        ('Mike', 'Johnson', '1985-03-20', '789 Oak St', 'Married', 'Technician', 45000)
```

```
    ]
```

```
    candidates = [
```

```
        ('Alice', 'Brown', 'Engineer', '2024-05-01', '2024-05-15', 'Interview Scheduled'),
```

```
        ('Bob', 'Davis', 'Manager', '2024-05-03', '2024-05-17', 'Application Received')
```

```
    ]
```

```
    positions = [
```

```
        ('Engineer', 'Engineering', 'BSc in Engineering, 3 years experience', 'Open'),
```

```
        ('Manager', 'Management', 'MBA, 5 years experience', 'Open')
```

```
    ]
```

```
    interviews = [
```

```
        (1, '2024-05-15', 'Jane Smith', 'Strong technical skills', 'Pending'),
```

```
        (2, '2024-05-17', 'John Doe', 'Excellent leadership qualities', 'Pending')
```

```
    ]
```

```

trainings = [
    ('Python Basics', 'Introduction to Python programming', '2024-06-01', '2 days'),
    ('Advanced Management', 'Advanced skills for management', '2024-06-15', '3 days')
]

employee_trainings = [
    (1, 1, '2024-06-03'),
    (2, 2, '2024-06-18')
]

cursor.executemany('INSERT INTO employees (first_name, last_name, birth_date, address, marital_status,
position, salary) VALUES (?, ?, ?, ?, ?, ?, ?)', employees)
cursor.executemany('INSERT INTO candidates (first_name, last_name, position_applied, application_date,
interview_date, status) VALUES (?, ?, ?, ?, ?, ?)', candidates)
cursor.executemany('INSERT INTO positions (position_name, department, requirements, status) VALUES
(?, ?, ?, ?)', positions)
cursor.executemany('INSERT INTO interviews (candidate_id, interview_date, interviewer, feedback,
decision) VALUES (?, ?, ?, ?, ?)', interviews)
cursor.executemany('INSERT INTO trainings (training_name, description, scheduled_date, duration)
VALUES (?, ?, ?, ?)', trainings)
cursor.executemany('INSERT INTO employee_trainings (employee_id, training_id, completion_date)
VALUES (?, ?, ?)', employee_trainings)

conn.commit()
conn.close()

populate_db()

```

Загальний код для створення графічного інтерфейсу:

```

import tkinter as tk
from tkinter import ttk
from tkinter import messagebox
import sqlite3

class HRSystemApp:
    def __init__(self, root):
        self.root = root
        self.root.title("HR Management System")

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

        self.create_home_tab()
        self.create_employee_tab()
        self.create_candidate_tab()
        self.create_training_tab()
        self.create_reports_tab()

    def create_home_tab(self):
        self.home_tab = ttk.Frame(self.notebook)
        self.notebook.add(self.home_tab, text="Головна")

```

```

label = ttk.Label(self.home_tab, text="Ласкаво просимо до системи управління персоналом",
font=("Helvetica", 16))
label.pack(pady=20)

btn_employees = ttk.Button(self.home_tab, text="Співробітники", command=lambda:
self.notebook.select(self.employee_tab))
btn_employees.pack(pady=5)

btn_candidates = ttk.Button(self.home_tab, text="Кандидати", command=lambda:
self.notebook.select(self.candidate_tab))
btn_candidates.pack(pady=5)

btn_trainings = ttk.Button(self.home_tab, text="Тренінги", command=lambda:
self.notebook.select(self.training_tab))
btn_trainings.pack(pady=5)

btn_reports = ttk.Button(self.home_tab, text="Звітності", command=lambda:
self.notebook.select(self.reports_tab))
btn_reports.pack(pady=5)

def create_employee_tab(self):
self.employee_tab = ttk.Frame(self.notebook)
self.notebook.add(self.employee_tab, text="Співробітники")

self.employee_tree = ttk.Treeview(self.employee_tab, columns=("ID", "First Name", "Last Name",
"Birth Date", "Address", "Marital Status", "Position", "Salary"), show="headings")
self.employee_tree.heading("ID", text="ID")
self.employee_tree.heading("First Name", text="First Name")
self.employee_tree.heading("Last Name", text="Last Name")
self.employee_tree.heading("Birth Date", text="Birth Date")
self.employee_tree.heading("Address", text="Address")
self.employee_tree.heading("Marital Status", text="Marital Status")
self.employee_tree.heading("Position", text="Position")
self.employee_tree.heading("Salary", text="Salary")
self.employee_tree.pack(expand=True, fill="both")

self.load_employees()

def create_candidate_tab(self):
self.candidate_tab = ttk.Frame(self.notebook)
self.notebook.add(self.candidate_tab, text="Кандидати")

self.candidate_tree = ttk.Treeview(self.candidate_tab, columns=("ID", "First Name", "Last Name",
"Position Applied", "Application Date", "Interview Date", "Status"), show="headings")
self.candidate_tree.heading("ID", text="ID")
self.candidate_tree.heading("First Name", text="First Name")
self.candidate_tree.heading("Last Name", text="Last Name")
self.candidate_tree.heading("Position Applied", text="Position Applied")
self.candidate_tree.heading("Application Date", text="Application Date")
self.candidate_tree.heading("Interview Date", text="Interview Date")
self.candidate_tree.heading("Status", text="Status")
self.candidate_tree.pack(expand=True, fill="both")

self.load_candidates()

```

```

def create_training_tab(self):
    self.training_tab = ttk.Frame(self.notebook)
    self.notebook.add(self.training_tab, text="Тренинги")

    self.training_tree = ttk.Treeview(self.training_tab, columns=("ID", "Training Name", "Description",
"Scheduled Date", "Duration"), show="headings")
    self.training_tree.heading("ID", text="ID")
    self.training_tree.heading("Training Name", text="Training Name")
    self.training_tree.heading("Description", text="Description")
    self.training_tree.heading("Scheduled Date", text="Scheduled Date")
    self.training_tree.heading("Duration", text="Duration")
    self.training_tree.pack(expand=True, fill="both")

    self.load_trainings()

def create_reports_tab(self):
    self.reports_tab = ttk.Frame(self.notebook)
    self.notebook.add(self.reports_tab, text="Звітності")

    label = ttk.Label(self.reports_tab, text="Звітності", font=("Helvetica", 16))
    label.pack(pady=20)

    btn_employee_report = ttk.Button(self.reports_tab, text="Звіт по співробітниках",
command=self.generate_employee_report)
    btn_employee_report.pack(pady=5)

    btn_candidate_report = ttk.Button(self.reports_tab, text="Звіт по кандидатах",
command=self.generate_candidate_report)
    btn_candidate_report.pack(pady=5)

    btn_training_report = ttk.Button(self.reports_tab, text="Звіт по тренінгах",
command=self.generate_training_report)
    btn_training_report.pack(pady=5)

def load_employees(self):
    conn = sqlite3.connect('hr_system.db')
    cursor = conn.cursor()
    cursor.execute("SELECT * FROM employees")
    rows = cursor.fetchall()
    for row in rows:
        self.employee_tree.insert("", "end", values=row)
    conn.close()

def load_candidates(self):
    conn = sqlite3.connect('hr_system.db')
    cursor = conn.cursor()
    cursor.execute("SELECT * FROM candidates")
    rows = cursor.fetchall()
    for row in rows:
        self.candidate_tree.insert("", "end", values=row)
    conn.close()

def load_trainings(self):
    conn = sqlite3.connect('hr_system.db')
    cursor = conn.cursor()

```

```

cursor.execute("SELECT * FROM trainings")
rows = cursor.fetchall()
for row in rows:
    self.training_tree.insert("", "end", values=row)
conn.close()

def generate_employee_report(self):
    conn = sqlite3.connect('hr_system.db')
    cursor = conn.cursor()
    cursor.execute("SELECT * FROM employees")
    rows = cursor.fetchall()
    report = "Звіт по співробітниках:\n"
    for row in rows:
        report += f"ID: {row[0]}, Ім'я: {row[1]}, Прізвище: {row[2]}, Посада: {row[6]}, Зарплата:
{row[7]}\n"
    conn.close()
    messagebox.showinfo("Звіт по співробітниках", report)

def generate_candidate_report(self):
    conn = sqlite3.connect('hr_system.db')
    cursor = conn.cursor()
    cursor.execute("SELECT * FROM candidates")
    rows = cursor.fetchall()
    report = "Звіт по кандидатах:\n"
    for row in rows:
        report += f"ID: {row[0]}, Ім'я: {row[1]}, Прізвище: {row[2]}, Позиція: {row[3]}, Статус:
{row[5]}\n"
    conn.close()
    messagebox.showinfo("Звіт по кандидатах", report)

def generate_training_report(self):
    conn = sqlite3.connect('hr_system.db')
    cursor = conn.cursor()
    cursor.execute("SELECT * FROM trainings")
    rows = cursor.fetchall()
    report = "Звіт по тренінгах:\n"
    for row in rows:
        report += f"ID: {row[0]}, Назва: {row[1]}, Опис: {row[2]}, Дата: {row[3]}, Тривалість: {row[4]}\n"
    conn.close()
    messagebox.showinfo("Звіт по тренінгах", report)

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

```