

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ  
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА ТЕХНОЛОГІЙ

Кафедра \_\_\_\_\_ Комп'ютерних інформаційних технологій \_\_\_\_\_

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач кафедри

\_\_\_\_\_ Аліна САВЧЕНКО

«\_\_\_\_\_» \_\_\_\_\_ 2023 р.

**КВАЛІФІКАЦІЙНА РОБОТА**  
**(ДИПЛОМНА РОБОТА, ПОЯСНЮВАЛЬНА ЗАПИСКА)**  
**ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ**  
**“МАГІСТРА”**  
ЗА ОСВІТНЬО-ПРОФЕСІЙНОЮ ПРОГРАМОЮ “ІНФОРМАЦІЙНІ  
УПРАВЛЯЮЧІ СИСТЕМИ ТА ТЕХНОЛОГІЇ”

**Тема:** «Інформаційна система управління персоналом»

**Виконав:** Першко Руслан Анатолійович

**Керівник:** доцент кафедри КІТ Колісник Олена Василівна

**Нормоконтролер** \_\_\_\_\_ Ігор РАЙЧЕВ

**Київ – 2023**

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет Комп'ютерних наук та технологій

Кафедра Комп'ютерних інформаційних технологій

Галузь знань, спеціальність, освітньо-професійна програма: 12 “Інформаційні технології”, 122 “Комп'ютерні науки”, “Інформаційні управляючі системи та технології”

**ЗАТВЕРДЖУЮ**

Завідувач кафедри

Аліна САВЧЕНКО

"\_\_" \_\_\_\_\_ 2023 р.

**ЗАВДАННЯ**

**на виконання кваліфікаційної роботи студента**

**Першко Руслана Анатолійовича**

(прізвище, ім'я, по батькові)

- 1. Тема роботи:** «Інформаційна система управління персоналом», затверджена наказом ректора від “29” вересня 2023р. за № 1976/ст
- 2. Термін виконання роботи:** : з 02 жовтня 2023р. по 31 грудня 2023 р.
- 3. Вихідні дані до роботи:** методи та засоби проектування та реалізації інформаційної системи управління персоналом.
- 4. Зміст пояснювальної записки (перелік питань, що підлягають розробці):** теоретичні основи розробки інформаційної системи управління персоналом; розробка інформаційної системи управління персоналом; впровадження та експлуатація інформаційної системи управління персоналом.
- 5. Перелік обов'язкового графічного матеріалу:** інформативні рисунки, графічні скріншоти роботи системи, слайди презентації в MS PowerPoint.

## 6. Календарний план-графік

№ з/п	Завдання	Термін виконання	Підпис керівника
1.	Пошук і дослідження наукових джерел.	02.10.2023 – 08.10.2023	
2.	Розроблення та затвердження календарного плану виконання дипломної роботи.	09.10.2023 – 10.10.2023	
3.	Проведення консультацій з науковим керівником.	11.10.2023 – 13.10.2023	
4.	Написання Розділу 1. Теоретичні основи розробки інформаційної системи управління персоналом.	14.10.2023 – 27.10.2023	
5.	Написання Розділу 2. Розробка інформаційної системи управління персоналом.	28.10.2023 – 16.11.2023	
6.	Написання Розділу 3. Впровадження та експлуатація інформаційної системи управління персоналом.	17.11.2022 – 30.11.2022	
7.	Оформлення пояснювальної записки дипломної роботи.	01.12.2023 – 04.12.2023	
8.	Написання, друк та підписання Рецензії у рецензента та Відгуку керівника у встановленому порядку.	05.12.2022 – 10.12.2023	
9.	Створення Презентації та доповіді.	11.12.2023 – 12.12.2023	
10.	Підготовка до захисту та попередній захист дипломної роботи на випусковій кафедрі.	13.11.2023 – 17.12.2023	

7. Дата видачі завдання: «02» жовтня 2023 р.

Керівник дипломної роботи \_\_\_\_\_ Олена КОЛІСНИК  
(підпис керівника) (П.І.Б.)

Завдання прийняв до виконання \_\_\_\_\_ Руслан ПЕРШКО  
(підпис випускника) (П.І.Б.)

## РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи «Інформаційна система управління персоналом» викладена на 95 сторінках і містить 30 рисунків, 3 таблиці та два додатки. Список бібліографічних посилань складається з 30 найменувань.

**Ключові слова:** УПРАВЛІННЯ, СИСТЕМА, ПЕРСОНАЛ, БАЗА ДАНИХ, МОДУЛЬ, АВТОРИЗАЦІЯ, ТЕСТУВАННЯ.

**Об'єкт дослідження** – процеси управління персоналом в організації.

**Предмет дослідження** – інформаційна система для автоматизації процесів управління персоналом.

**Мета роботи.** Розробка інформаційної системи управління персоналом для підвищення ефективності кадрових процесів в організації.

**Актуальність теми.** Ефективне управління персоналом є запорукою успішної діяльності будь-якої організації. Адже саме працівники, їх знання, вміння та мотивація визначають результативність роботи компанії. У сучасних умовах гострої конкуренції компанії змушені постійно оптимізувати бізнес-процеси для підвищення ефективності. Одним з напрямків оптимізації є впровадження інформаційних технологій в управління персоналом.

Автоматизація HR-процесів за допомогою інформаційних систем дозволяє суттєво підвищити продуктивність праці в цій сфері. Рутинні та трудомісткі операції (ведення кадрових документів, розрахунок зарплати, пошук кандидатів тощо) виконуються швидше і з меншою кількістю помилок. Вивільняється час фахівців з персоналу для вирішення більш складних стратегічних завдань. Системи дозволяють формувати різноманітну аналітику в розрізі показників ефективності персоналу, що є основою для прийняття управлінських рішень.

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ**

ID	Identity Document
HRM	Human Resources Management
ERP	Enterprise Resource Planning
CRM	Customer Relationship Management
CSS	Cascading Style Sheets
SQL	Structured Query Language
UX	User Experience
UI	User Interface
PHP	Personal Home Page
SAP	System Analysis Program
СУБД	Система управління базами даних

## ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ПЕРСОНАЛОМ.....	10
1.1. Аналіз існуючих інформаційних систем управління персоналом.....	10
1.2. Склад та функції інформаційних системи управління персоналом.....	15
1.3. Обґрунтування доцільності розробки власної системи управління персоналом.....	17
1.4. Вибір інструментів розробки системи (мова програмування, СКБД).....	20
Висновки до розділу 1.....	25
РОЗДІЛ 2. РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ПЕРСОНАЛОМ.....	26
2.1. Вимоги до системи та функціональні можливості.....	26
2.2. Проектування структури бази даних.....	30
2.3. Розробка архітектури та інтерфейсу системи.....	34
2.4. Модуль інформації про співробітника.....	47
2.5. Тестування розробленої системи.....	54
Висновки до розділу 2.....	58
РОЗДІЛ 3. ВПРОВАДЖЕННЯ ТА ЕКСПЛУАТАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ПЕРСОНАЛОМ.....	59
3.1. Рекомендації щодо впровадження системи.....	59
3.2. Інструкція користувача системи.....	62
3.3. Переваги розробленої системи у порівнянні з аналогами.....	67
Висновки до розділу 3.....	70
ВИСНОВКИ.....	71
СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ.....	73
Додаток А.....	76
Додаток Б.....	94

## ВСТУП

**Актуальність теми.** Ефективне управління персоналом є запорукою успішної діяльності будь-якої організації. Адже саме працівники, їх знання, вміння та мотивація визначають результативність роботи компанії. У сучасних умовах гострої конкуренції компанії змушені постійно оптимізувати бізнес-процеси для підвищення ефективності. Одним з напрямків оптимізації є впровадження інформаційних технологій в управління персоналом.

Автоматизація HR-процесів за допомогою інформаційних систем дозволяє суттєво підвищити продуктивність праці в цій сфері. Рутинні та трудомісткі операції (ведення кадрових документів, розрахунків зарплати, пошук кандидатів тощо) виконуються швидше і з меншою кількістю помилок. Вивільняється час фахівців з персоналу для вирішення більш складних стратегічних завдань. Системи дозволяють формувати різноманітну аналітику в розрізі показників ефективності персоналу, що є основою для прийняття управлінських рішень.

Разом з тим, більшість готових систем не враховують специфіку конкретної організації, її структуру, бізнес-процеси, корпоративну культуру. Це ускладнює їх впровадження і знижує ефективність. Тому на сьогодні є актуальною розробка власних корпоративних систем управління персоналом, які максимально відповідають потребам і особливостям організації.

Отже, тема дослідження є надзвичайно актуальною в умовах інформатизації та оптимізації бізнес-процесів в організаціях. Розробка інформаційної системи управління персоналом дозволить автоматизувати рутинні операції, підвищити продуктивність HR-фахівців, отримати якісну аналітику для прийняття управлінських рішень.

**Мета і завдання дослідження.** Метою роботи є розробка інформаційної системи управління персоналом для підвищення ефективності кадрових процесів в організації.

**Завдання:**

- проаналізувати існуючі системи управління персоналом;
- визначити функціональні вимоги до системи;
- обрати засоби розробки;
- спроектувати структуру бази даних;
- розробити систему та інтерфейс користувача;
- протестувати роботу системи;
- скласти рекомендації щодо впровадження.

**Методи дослідження:**

– аналіз наукової літератури та інших джерел за тематикою дослідження. Використовувався для вивчення теоретичних основ проектування та розробки корпоративних систем управління персоналом, визначення їх функціональних можливостей;

– порівняльний аналіз існуючих систем управління персоналом. Дозволив виявити їх переваги та недоліки, сформулювати вимоги до проектованої системи;

– моделювання – побудова функціональної, інформаційної та об'єктної моделей системи управління персоналом. Використовувалось для проектування логічної структури та архітектури системи;

– програмування – розробка системи мовою PHP, СУБД MySQL. Дозволило створити програмний продукт, що реалізує запроектовані моделі та виконує необхідні функції;

– тестування – перевірка працездатності розробленої системи, її відповідності поставленим вимогам. Проводилось на основі тестових сценаріїв та даних;

– аналіз та узагальнення результатів дослідження. Дозволило оцінити ефективність розробленої системи, сформулювати висновки та рекомендації.

Таким чином, застосовані методи дослідження дозволили комплексно вирішити поставлені в роботі завдання.



**Наукова новизна** полягає в комплексному підході до проектування та розробки інформаційної системи управління персоналом з урахуванням специфіки конкретної організації. Вперше запропоновано модель системи, яка враховує структуру, штатний розпис, посадові інструкції, бізнес-процеси, корпоративну культуру компанії. Розроблено унікальну архітектуру та структуру бази даних системи, що забезпечує гнучкість налаштування під конкретного замовника. Запропоновано нові алгоритми розрахунку показників ефективності персоналу та формування HR-аналітики. У комплексі це дозволяє створити персоналізовану систему управління персоналом, максимально адаптовану до потреб організації.

**Практичне значення** роботи полягає в можливості впровадження розробленої системи для комплексної автоматизації процесів управління персоналом. Запропоновані моделі, алгоритми, архітектура та структура бази даних системи дозволяють мінімізувати витрати часу та ресурсів на її налаштування під потреби конкретної компанії. Впровадження системи сприятиме:

- підвищенню продуктивності фахівців HR-відділу;
- прискоренню та спрощенню рутинних операцій;
- мінімізації паперового документообігу;
- покращенню якості аналітики та звітності з персоналу;
- прийняттю обґрунтованих управлінських рішень на основі даних;
- формуванню кадрового резерву та плануванню кар'єрного росту працівників.

У цілому впровадження системи дозволить оптимізувати процеси управління персоналом та підвищити загальну результативність HR-відділу і організації в цілому.

**Кваліфікаційна робота** складається зі вступу, трьох розділів, висновків, списку використаної літератури (30 найменувань) та 2 додатки.

# РОЗДІЛ 1

## ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ПЕРСОНАЛОМ

### 1.1. Аналіз існуючих інформаційних систем управління персоналом

Ефективне управління сучасною організацією неможливе без використання інформаційних технологій та систем, що дозволяють автоматизувати бізнес-процеси. Одним з ключових напрямків автоматизації є управління людськими ресурсами, або інформаційний менеджмент персоналу.

Інформаційний менеджмент персоналу полягає в раціональній організації всієї сукупності інформаційних процесів, пов'язаних з персоналом: збір, накопичення, обробка, аналіз даних про працівників з метою прийняття ефективних кадрових рішень.

Однією з основних методів інформаційного менеджменту є використання інформаційних систем, що дозволяє ефективно збирати, обробляти та зберігати інформацію. До складу інформаційної системи можуть входити різноманітні засоби збирання та обробки інформації, такі як бази даних, програми для аналізу даних, системи управління документами та інші.

Інший метод – це розробка та використання інформаційних стратегій. Інформаційна стратегія – це план дій, який визначає, як організація буде збирати, обробляти та використовувати інформацію для досягнення своїх цілей. Вона має включати аналіз потреб організації в інформації, визначення джерел інформації, розробку методів збору та обробки інформації, оцінку ефективності інформаційної діяльності та інші етапи [3].

<i>Кафедра КІТ (47)</i>				<i>НАУ 23.16.20.000 ПЗ</i>			
<i>Виконав</i>	<i>Першко Р.А.</i>			ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ПЕРСОНАЛОМ	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Керівник</i>	<i>Колісник О.В.</i>				Д	10	16
<i>Консульт.</i>					<i>УС-211Мз</i>		<i>122</i>
<i>Н. Контр.</i>	<i>Райчев І.Е.</i>						

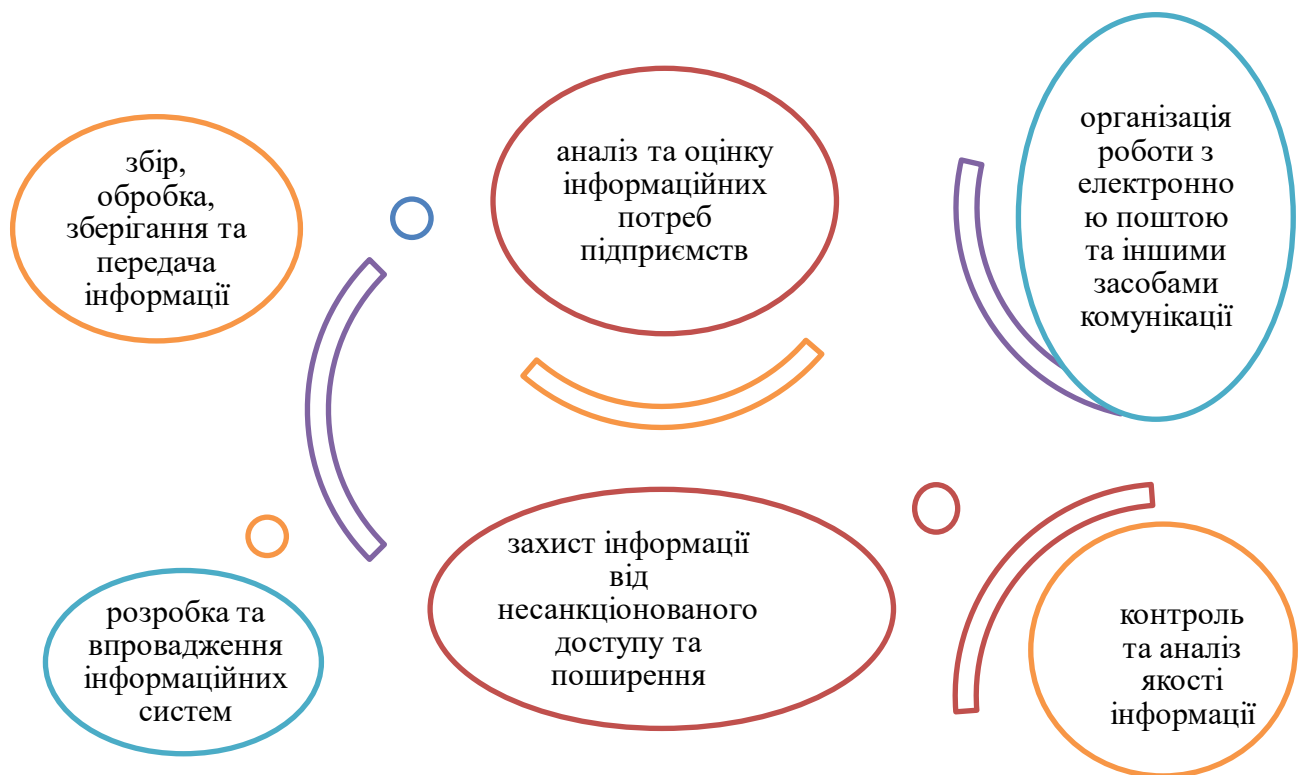


Рис. 1.1. Основні методи інформаційного менеджменту [12]

Для досягнення ефективних результатів у роботі інформаційного менеджменту необхідно використовувати сучасні інформаційні технології та програмне забезпечення, а також мати професійний колектив спеціалістів у галузі інформаційних технологій та менеджменту.

Основним інструментом інформаційного менеджменту персоналу виступають електронні системи управління персоналом (ЕСУП), призначені для комплексної автоматизації HR-процесів в організації.

Електронна система управління персоналом (ЕСУП) – це програмно-апаратний комплекс, який дозволяє автоматизувати та оптимізувати процеси управління людськими ресурсами в організації. ЕСУП охоплює всі ключові HR-процеси: від підбору персоналу до вивільнення, включаючи облік кадрів,

розрахунок зарплати, оцінку персоналу, навчання та розвиток. Переваги ЕСУП – підвищення продуктивності HR-відділу, зниження витрат часу та коштів на управління персоналом, поліпшення якості та швидкості прийняття кадрових рішень.

На ринку представлено багато ЕСУП як вітчизняних, так і зарубіжних виробників (табл. 1.1).

Таблиця 1.1.

Програми у сфері інформаційного менеджменту для системи управління персоналом (HRM)

Назва програми	Опис програми	Функціонал
Система управління персоналом (HRM)		
SAP SuccessFactors	Система управління персоналом, що дозволяє керувати процесами найму, оцінки та розвитку персоналу, забезпечувати доступ до даних про співробітників та аналізувати результати.	Керування процесами найму, оцінки та розвитку персоналу, забезпечення доступу до даних про співробітників, аналітика даних.
Oracle HCM Cloud	Система управління персоналом, що дозволяє керувати процесами найму, оцінки та розвитку персоналу, забезпечувати доступ до даних про співробітників та аналізувати результати.	Керування процесами найму, оцінки та розвитку персоналу, забезпечення доступу до даних про співробітників, аналітика даних.
Workday Human Capital Management	Система управління персоналом, що дозволяє керувати процесами найму, оцінки та розвитку персоналу, забезпечувати доступ до даних про співробітників та аналізувати результати.	Керування процесами найму, оцінки та розвитку персоналу, забезпечення доступу до даних про співробітників, аналітика даних.

Ці програми допомагають управляти різними аспектами персоналу, такими як найм, оцінка працівників, навчання та розвиток, забезпечення безпеки та здоров'я на робочому місці та багато іншого. Вони також надають можливість зберігати та аналізувати дані про персонал, що дозволяє компаніям приймати кращі управлінські рішення щодо персоналу.

Програми у сфері інформаційного менеджменту для системи управління персоналом (HRM) дозволяють компаніям зберігати та керувати інформацією про своїх співробітників, а також забезпечувати ефективний процес управління персоналом. За допомогою цих програм компанії можуть здійснювати ряд функцій, таких як розрахунок заробітної плати, ведення електронних дос'є співробітників, моніторинг продуктивності та виконання завдань, навчання та розвиток персоналу, підвищення мотивації та задоволеності співробітників тощо [14].

Одним з найважливіших аспектів програм у сфері інформаційного менеджменту для HRM є забезпечення точності та надійності інформації про співробітників. Це дозволяє компаніям більш ефективно планувати та керувати своїм персоналом, забезпечувати правильний розрахунок заробітної плати та інших виплат, а також забезпечувати виконання законодавчих вимог та норм.

Інформаційні технології дозволяють підтримувати різні аспекти управління бізнесом, забезпечувати доступ до потрібної інформації, підвищувати продуктивність та ефективність, полегшувати комунікацію та співпрацю між колегами та дозволяти приймати кращі управлінські рішення. Використання таких програм може допомогти компаніям зберегти час та зусилля та підвищити свою конкурентоспроможність на ринку.

SAP SuccessFactors - потужна хмарна ЕСУП від компанії SAP, одного з лідерів на ринку бізнес-додатків. Система охоплює весь спектр задач управління персоналом: підбір, адаптація, навчання, оцінка, планування кар'єри, мотивація персоналу.

Ключові можливості:

- автоматизація процесів підбору, оцінки, адаптації персоналу;

- ведення особових справ і самообслуговування для співробітників;
- оцінка компетенцій, планування навчання та розвитку персоналу;
- багатовимірний HR-аналіз і звітність;
- модульна структура і гнучкість налаштувань.

Хмарний додаток від Oracle для комплексного управління персоналом. Включає функції планування штатного розпису, найму, відстеження продуктивності, компенсацій і пільг, навчання та розвитку персоналу.

Ключові можливості:

- ведення кадрового обліку і організаційної структури;
- управління процесами найму і адаптації;
- модуль компенсацій, пільг і мотивації персоналу;
- планування кар'єрного шляху і навчання;
- HR-аналітика і створення звітів;
- інтеграція з хмарними ERP і CRM рішеннями Oracle.

Зручне хмарне рішення від Workday для автоматизації процесів управління персоналом, поєднує функції кадрового обліку, навчання, оцінки і мотивації співробітників.

Ключові можливості:

- комплексне ведення кадрової інформації;
- реєстрація робочого часу і управління відпустками;
- планування витрат на персонал і контроль бюджету;
- оцінка персоналу і управління талантами;
- модуль аналітики і звітності;
- інтуїтивний інтерфейс і мобільний додаток.

На основі проведеного аналізу можна зробити висновок, що на ринку представлено широкий спектр електронних систем управління персоналом як вітчизняних, так і зарубіжних виробників. Серед розглянутих ЕСУП найбільш функціональними та зручними у використанні є SAP SuccessFactors, Oracle HCM Cloud та Workday HCM.

Ці системи дозволяють комплексно автоматизувати основні HR-процеси: ведення кадрового обліку, організаційної структури, підбір персоналу, оцінка і навчання, розрахунок зарплати. Вони мають гнучкі можливості налаштування, зручний інтерфейс, потужні засоби HR-аналітики.

Для нашої компанії оптимальним рішенням може бути аналог, власна розробка подібної програми завдяки простоті впровадження, інтуїтивному інтерфейсу та широким аналітичним можливостям. Її використання дозволить автоматизувати HR-процеси, підвищити продуктивність праці та якість прийняття управлінських рішень в галузі персоналу.

## **1.2. Склад та функції інформаційних системи управління персоналом**

Електронна система управління персоналом (ЕСУП) являє собою комплексну інформаційну систему, призначену для автоматизації та підвищення ефективності процесів управління людськими ресурсами в організації. Вона дозволяє вести в електронному вигляді кадрову інформацію, оптимізувати виконання HR-функцій, підвищити оперативність та обґрунтованість прийняття управлінських рішень в галузі персоналу.

З точки зору складу ЕСУП являє собою інформаційну систему, що включає наступні компоненти:

- програмне забезпечення на базі прикладних програм і баз даних, яке реалізує всі необхідні функції і процедури обробки інформації про персонал. Може містити десятки спеціалізованих модулів для автоматизації окремих напрямків кадрового менеджменту;
- інформаційне забезпечення у вигляді баз і банків даних з інформацією про працівників організації, їх кваліфікацію, посади, результати роботи тощо;
- технічне забезпечення, що включає сервери, персональні комп'ютери, мережеве обладнання, засоби зв'язку і передачі даних, на яких функціонує система;

– організаційно-методичне забезпечення у вигляді нормативно-правової документації, що регламентує процеси управління персоналом та роботу з ЕСУП.

З точки зору функціональності ключовими можливостями ЕСУП є (рис. 1.2):



Рис. 1.2. Функції електронної системи управління персоналом [14]

Ведення особових справ працівників передбачає можливість зберігати в системі в електронному вигляді всю необхідну інформацію про кожного співробітника: особисті дані, освіта, досвід роботи, сімейний стан, медичні дані, результати атестацій, нагороди, знання мов, навички роботи з програмами тощо. Це дозволяє мати повну картину про персонал організації.



Облік робочого часу дає можливість відстежувати відпрацьований час співробітників, контролювати своєчасність приходу та виходу з роботи, нараховувати зарплату відповідно до відпрацьованих годин. Можливий облік переробок, нічних змін, роботи у вихідні.

Розрахунок заробітної плати включає нарахування окладів, премій, доплат, утримань, податків згідно з заданими формулами розрахунку, з урахуванням усіх особливостей оплати праці в організації.

Управління процесами найму і звільнення дозволяє автоматизувати оформлення прийому та звільнення співробітників, формування наказів, внесення змін в особові справи.

Облік відпусток і лікарняних включає планування та моніторинг використання відпусток персоналом, автоматичний облік лікарняних за даними листків непрацездатності.

Організація навчання персоналу охоплює функції планування курсів, тренінгів, семінарів, бюджетів на навчання, направлення співробітників, обліку результатів. Оцінка персоналу передбачає проведення комплексної оцінки, фіксацію результатів, формування планів розвитку.

HR-аналітика надає широкі можливості аналізу даних про персонал, побудови звітів, виявлення тенденцій для прийняття управлінських рішень.

Самообслуговування дозволяє співробітникам та менеджерам взаємодіяти з кадровою системою через особистий кабінет для виконання ряду завдань.

Отже, ЕСУП забезпечує комплексну підтримку всіх ключових HR-процесів на новому якісному рівні за рахунок їх автоматизації. Це дозволяє суттєво підвищити ефективність управління персоналом в організації.

### **1.3. Обґрунтування доцільності розробки власної системи управління персоналом**

Незважаючи на наявність широкого спектру готових рішень, для нашої компанії може виявитися доцільним створення власної системи управління персоналом.

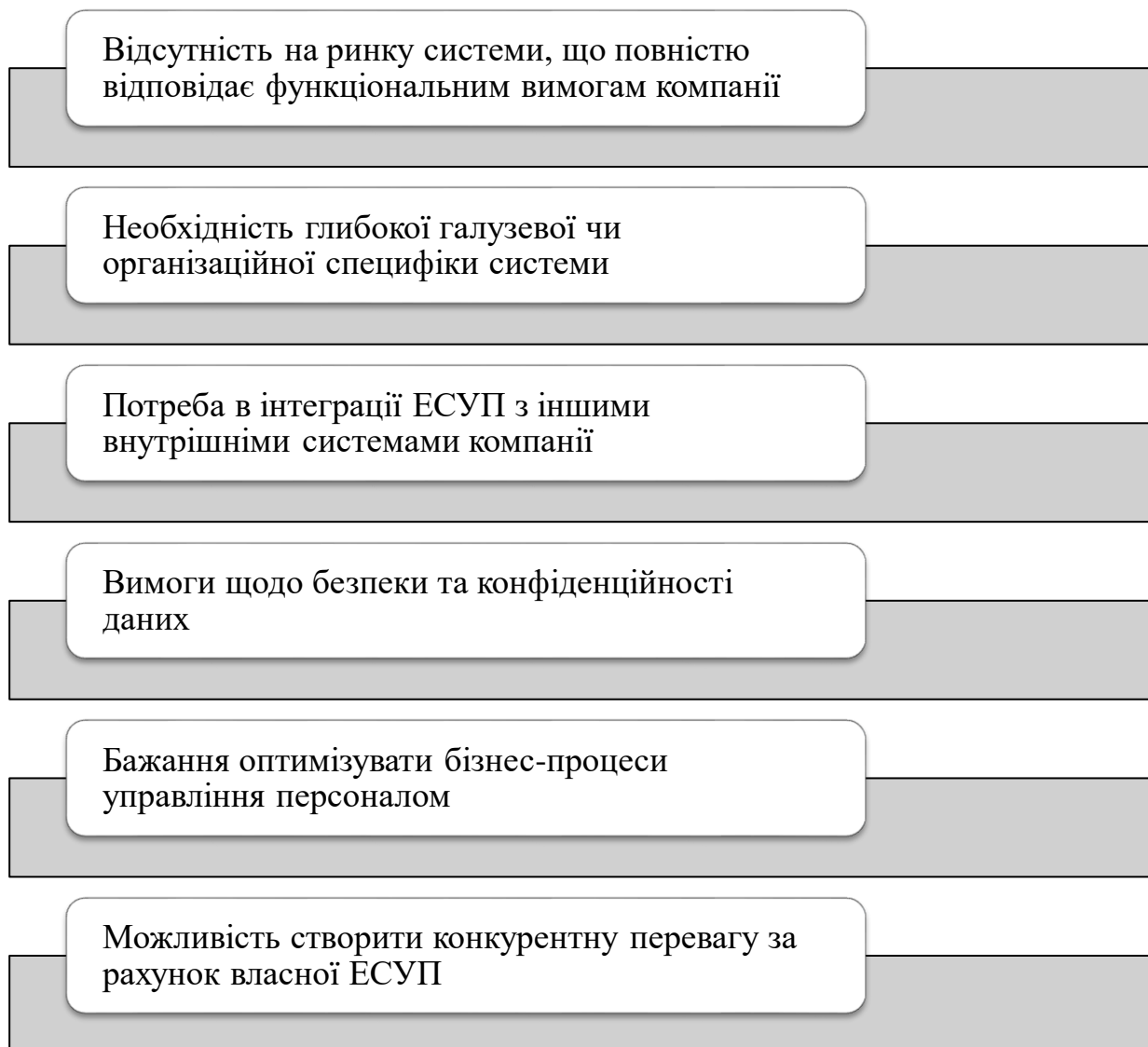


Рис. 1.3. Обґрунтування доцільності розробки власної системи управління персоналом [14-15]

По-перше, існуючі на ринку системи орієнтовані на задоволення потреб пересічної компанії, тоді як наша організація має глибоку галузеву специфіку діяльності і потребує забезпечення унікальних бізнес-процесів в сфері управління персоналом. Готові стандартні системи можуть не підходити для оптимальної підтримки наших задач.

По-друге, нам необхідна тісна інтеграція модулів управління персоналом з іншими внутрішніми інформаційними системами для реалізації складних

крос-функціональних бізнес-процесів. Це може бути складно реалізувати при використанні стороннього програмного забезпечення.

По-третє, у нас є специфічні і жорсткі вимоги в сфері безпеки та конфіденційності даних про персонал, які легше забезпечити при використанні власної системи. Ми зможемо мати повний контроль над даними.

По-четверте, власна розробка ЕСУП дозволить максимально точно оптимізувати HR-процеси під потреби нашого бізнесу.

І нарешті, наявність сучасної ефективної системи управління персоналом власної розробки може стати нашою конкурентною перевагою на ринку.

Безумовно, розробка власної ЕСУП вимагає істотних початкових інвестицій. Однак з огляду на зазначені фактори, це має сенс як стратегічне рішення, що окупиться за рахунок оптимізації витрат на персонал. Доцільно розглянути комбінований підхід з використанням як власних модулів, так і готових рішень для стандартних функцій.

Отже, на основі проведеного аналізу можна зробити висновок, що для нашої компанії доцільно розглянути можливість розробки власної системи управління персоналом, а не впровадження типового готового рішення. Головними аргументами на користь створення власної ЕСУП є потреба в рахуванні галузевої специфіки нашої діяльності, необхідність інтеграції з іншими внутрішніми інформаційними системами, особливі вимоги до безпеки та конфіденційності даних, а також можливість оптимізувати HR-процеси під цілі бізнесу та отримати стратегічні конкурентні переваги. Безсумнівно, розробка ЕСУП власними силами потребує значних початкових інвестицій. Проте на перспективу це окупиться за рахунок підвищення ефективності управління персоналом. Доцільно розглянути комбінований підхід з використанням готових рішень для стандартних функцій. В цілому, створення сучасної ЕСУП відповідає стратегічним інтересам компанії.

#### 1.4. Вибір інструментів розробки системи (мова програмування, СКБД тощо)

Система управління персоналом відіграє критично важливу роль в організації. Вона впливає на ефективність роботи команди, мотивацію співробітників та загалом на успішність підприємства. Отже, підбір оптимальних технологій для її розробки вимагає серйозного підходу.

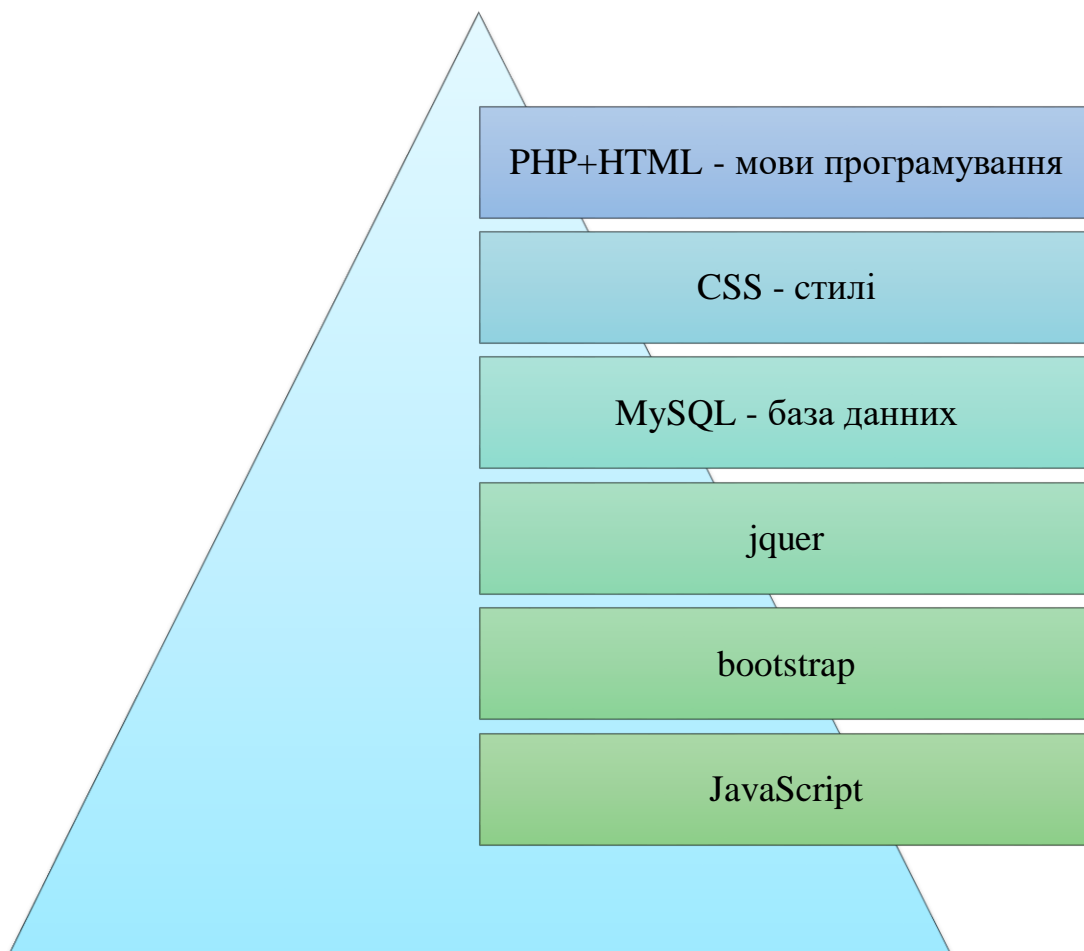


Рис. 1.4. Вибір інструментів розробки системи

PHP вибрано як мову програмування завдяки її високій продуктивності, популярності серед розробників та наявності великої кількості додаткових інструментів, які спрощують розробку веб-додатків.

MySQL в якості СУБД використовується для забезпечення надійного зберігання даних та гарантування швидкого доступу до інформації. Завдяки своїй стабільності та високій продуктивності, MySQL є однією з найбільш популярних реляційних баз даних у світі.

HTML та CSS становлять основу для створення користувацького інтерфейсу. Вони дозволяють створювати інтуїтивно зрозумілі, адаптивні та сучасні веб-інтерфейси.

Bootstrap – це фреймворк, який спрощує і прискорює розробку інтерфейсу, дозволяючи розробникам створювати консистентний та адаптивний дизайн.

jQuery – це бібліотека, яка значно спрощує роботу з JavaScript, дозволяючи легше інтегрувати різноманітні інтерактивні елементи на сторінці.

Laravel в якості PHP-фреймворка вибрано через його високу продуктивність, безпеку та гнучкість. Завдяки Laravel можна швидко і ефективно створювати складні веб-додатки, отримуючи при цьому всі переваги сучасних підходів до розробки.

В цілому, обраний стек технологій дозволить нам створити сучасну, надійну та легко масштабовану систему управління персоналом, яка відповідатиме всім сучасним вимогам і стандартам.

Основою будь-якого додатку є мова розмітки HTML. Вона виконує критично важливу функцію структурування контенту та визначення його семантики. За допомогою HTML розробники організують текст, зображення, мультимедіа та інші елементи у логічному порядку, використовуючи спеціальні теги. Ці теги слугують своєрідним каркасом майбутньої веб-сторінки, забезпечуючи базову розмітку.

Вбудовуючи різноманітний контент у структуру HTML, розробники створюють фундамент для подальшого оформлення та надання інтерактивності за допомогою каскадних таблиць стилів (CSS) та мов програмування, таких як JavaScript. Саме завдяки універсальності HTML, веб-сторінки можуть коректно

відображатися у більшості існуючих браузерів, а отже контент стає доступним широкому колу користувачів.

Організуючи інформацію за допомогою HTML, розробники веб-додатків також оптимізують користувацький досвід, роблячи навігацію сайтом більш зручною та інтуїтивно зрозумілою. Логічна структура з використанням відповідних тегів та атрибутів спрощує пошук потрібного контенту та взаємодію з інтерфейсом. Таким чином, грамотне застосування HTML є запорукою не лише технічної досконалості, але й зручності роботи кінцевих користувачів з веб-додатком.

Каскадні таблиці стилів CSS відіграють вирішальну роль у створенні привабливого та функціонального інтерфейсу додатків. Вони дозволяють трансформувати суху HTML-розмітку в естетично приємний та візуально організований дизайн. За допомогою CSS розробники визначають стилі шрифтів, кольори, розташування елементів, анімації та багато інших аспектів подання контенту.

Перевагою використання CSS є можливість чіткого розділення структури та дизайну. HTML відповідає за логічну організацію контенту, а CSS – за його подання. Це дає значну гнучкість, адже змінюючи лише CSS можна миттєво оновити зовнішній вигляд усього сайту.

CSS базується на каскадному принципі успадкування стилів від батьківських елементів до дочірніх. Це дозволяє створювати ієрархію класів та селекторів для структурованого управління дизайном складних інтерфейсів. Також CSS забезпечує адаптивність, даючи змогу налаштовувати відображення під різні пристрої та розміри екранів.

Застосування CSS є обов'язковим елементом сучасної розробки веб-додатків для досягнення привабливості, зручності та візуальної цілісності інтерфейсу. Це критично важливо для користувацького досвіду та залучення аудиторії.

Фреймворк Bootstrap є де-факто стандартом в індустрії розробки сучасних адаптивних веб-інтерфейсів. Завдяки його інструментам на основі

HTML, CSS та JavaScript, створення респонсивного та мобільно-орієнтованого дизайну значно спрощується.

Основною перевагою Bootstrap є набір готових UI-компонентів та стилів, що дозволяють швидко збирати професійно виглядаючі інтерфейси. Ця екосистема включає кнопки, навігаційні панелі, модальні вікна, акордеони, пагінацію та інші поширені елементи. Їх можна легко інтегрувати та налаштовувати.

Ще однією ключовою перевагою є система сітки Bootstrap з адаптивними контейнерами та рядками. Вона дозволяє будувати відгуківі макети під різні розміри екранів від мобільних до десктопних пристроїв. Такий підхід гарантує оптимальне відображення контенту на будь-яких пристроях.

Засвоєння Bootstrap не потребує глибоких технічних знань завдяки ґрунтовній документації та чисельним прикладам. Навіть початківці можуть швидко опанувати основи та почати застосовувати його в проектах. Підтримка різних JavaScript бібліотек та розширень також спрощує розробку складних інтерактивних веб-додатків.

Загалом, завдяки Bootstrap можна заощадити багато часу та ресурсів на вирішенні типових задач створення адаптивного та мобільного UI/UX. Це дає змогу фокусуватися на бізнес-логіці та інтеграції. Bootstrap є оптимальним вибором для швидкого створення якісних респонсивних інтерфейсів з мінімальними витратами.

Система управління базами даних MySQL є фундаментальним компонентом сучасних веб-додатків, забезпечуючи надійне зберігання та ефективну обробку інформації. Завдяки вражаючій продуктивності та масштабованості, MySQL може впоратися з величезними обсягами даних, що циркулюють у популярних веб-сервісах та онлайн-платформах.

MySQL демонструє стабільно високу швидкодію як на невеликих, так і на вкрай великих базах, що налічують мільйони записів та тисячі транзакцій за секунду. Це досягається завдяки ефективній реалізації механізмів кешування,

індексації, паралельної обробки запитів та оптимізації складних операцій з даними.

MySQL також відрізняється розвинутими засобами адміністрування, що включають потужні інструменти резервного копіювання та відновлення, реплікації для створення відмовостійких конфігурацій, а також моніторингу продуктивності в реальному часі.

Ще однією ключовою перевагою MySQL є підтримка широкого кола мов програмування – від PHP до Python та Java. Це спрощує інтеграцію функцій бази даних безпосередньо у веб-додаток за допомогою SQL-запитів. Розробники також можуть використовувати зручні інтерфейси (API) для маніпулювання даними.

Загалом, поєднання високої продуктивності, надійності, масштабованості та гнучкості робить MySQL ідеальним вибором СУБД для сучасних веб-проектів, що прагнуть ефективно обробляти великі обсяги запитів та даних від тисяч користувачів. Саме тому ця система набула такої популярності серед веб-розробників по всьому світу.

Обрані інструменти для розробки системи управління персоналом відображають стратегічний підхід до оптимізації робочих процесів в організації. Використання PHP як мови програмування, MySQL для збереження даних, а також інших технологій, таких як HTML, CSS, Bootstrap, jQuery та Laravel, вказує на прагнення до створення надійної, гнучкої та масштабованої системи. Ці технологічні рішення підкреслюють важливість гарантованого зберігання даних, інтуїтивно зрозумілого інтерфейсу та можливості адаптації до майбутніх потреб організації. Таким чином, підбір цих інструментів відображає стратегічне бачення необхідності створення сучасної системи управління персоналом, яка буде відповідати вимогам сучасного бізнес-середовища та сприятиме зростанню та розвитку компанії.



## **Висновки до розділу 1**

У першому розділі було проведено глибоке дослідження теоретичних основ розробки інформаційної системи управління персоналом. В ході аналізу існуючих інформаційних систем було виявлено їх ключові особливості, переваги та недоліки. Розділ, присвячений складу та функціям системи, дав зрозуміти структурні компоненти та основні задачі, які виконує система управління персоналом.

Обґрунтовуючи доцільність розробки власної системи, було наголошено на важливості індивідуального підходу до специфіки кожного підприємства, а також на можливості адаптації системи під конкретні бізнес-потреби.

Останній підрозділ розкрив аспекти вибору технологічного стеку для розробки системи. Обравши сучасний набір інструментів, таких як PHP, MySQL, Laravel та ін., ми закладаємо основу для створення надійної, масштабованої та гнучкої системи управління персоналом.

Загалом, розділ надає чітке розуміння теоретичних засад створення системи управління персоналом та вказує на ключові аспекти, які слід враховувати під час її розробки для досягнення максимальної ефективності та відповідності бізнес-вимогам.

## РОЗДІЛ 2

### РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ПЕРСОНАЛОМ

#### 2.1. Вимоги до системи та функціональні можливості

В сучасних умовах, коли велика кількість інформації потребує ефективного управління, особливого значення набуває впровадження автоматизованих систем управління персоналом. Розробка даної системи базується на вимогах до функціональності та можливостей, які визначаються як відповідь на потреби компанії в обліку та управлінні співробітниками.



Рис. 2.1. Основні вимоги до системи

<b>Кафедра КІТ (47)</b>				<b>НАУ 23.16.20.000 ПЗ</b>			
<b>Виконав</b>	Першко К.В.			ВПРОВАДЖЕННЯ ТА ЕКСПЛУАТАЦІЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ УПРАВЛІННЯ БУХГАЛТЕРСЬКИМИ РОЗРАХУНКАМИ ТА ОБЛІКУ ПРОДАЖІВ	<b>Літ.</b>	<b>Арк.</b>	<b>Аркушів</b>
<b>Керівник</b>	Колісник О.В.				Д	26	33
<b>Консульт.</b>					<b>УС-211Мз</b>		<b>122</b>
<b>Н. Контр.</b>	Райчев І.Е.						

Перш за все, безпека та конфіденційність є важливими аспектами управління даними в системі. Система повинна гарантувати, що доступ до інформації мають лише авторизовані користувачі з відповідними правами доступу. Це допомагає уникнути неправомірного доступу та забезпечує інтегральність та конфіденційність даних.

Далі, інтуїтивно зрозумілий інтерфейс є ключовим для ефективності використання системи. Користувачі повинні мати змогу легко та швидко засвоїти основні функції та можливості системи без необхідності проходження складного та тривалого навчання. Це сприяє продуктивності та ефективності роботи користувачів.

Масштабованість також є важливою вимогою, яка забезпечує можливість адаптації системи до росту та розвитку компанії. Система повинна бути гнучкою, щоб дозволити додавання нових функцій та опцій відповідно до змінюваних потреб бізнесу.

Нарешті, інтеграція з іншими корпоративними системами дозволяє створити єдине інформаційне середовище, яке сприяє обміну даними та автоматизації бізнес-процесів. Це полегшує управління ресурсами компанії та підвищує ефективність операційних процесів. Врахування цих вимог є ключовим для розробки та впровадження успішної інформаційної системи, яка буде відповідати актуальним потребам та викликам бізнесу.

Система, що розглядається, представляє собою комплексний інструмент для ефективного управління персоналом та ресурсами компанії, який включає в себе ряд ключових функціональних можливостей.

По-перше, система забезпечує облік співробітників, дозволяючи реєструвати нових працівників, зберігати актуальну інформацію про кожного співробітника (таку як ПІБ, посада, контактні дані), а також редагувати та видаляти записи за потреби. Це сприяє організації та систематизації даних про персонал.

Далі, функціонал авторизації користувачів дозволяє забезпечити безпеку доступу до системи, використовуючи логіни та паролі, а також встановлювати різні рівні доступу, що відповідають ролі кожного користувача в системі.

Облік підрозділів компанії також є важливою частиною системи, дозволяючи створювати, редагувати та видаляти підрозділи, визначати керівників підрозділів та будувати ієрархічну структуру організації.

Система також включає в себе облік відпусток, де можна реєструвати запити на відпустку, переглядати списки запланованих та вже здійснених відпусток, що сприяє плануванню робочого часу та ресурсів.

Календар відпусток візуалізує інформацію, дозволяючи легко орієнтуватися в запланованих відпустках, а також фільтрувати записи за різними критеріями, такими як дати або співробітники.

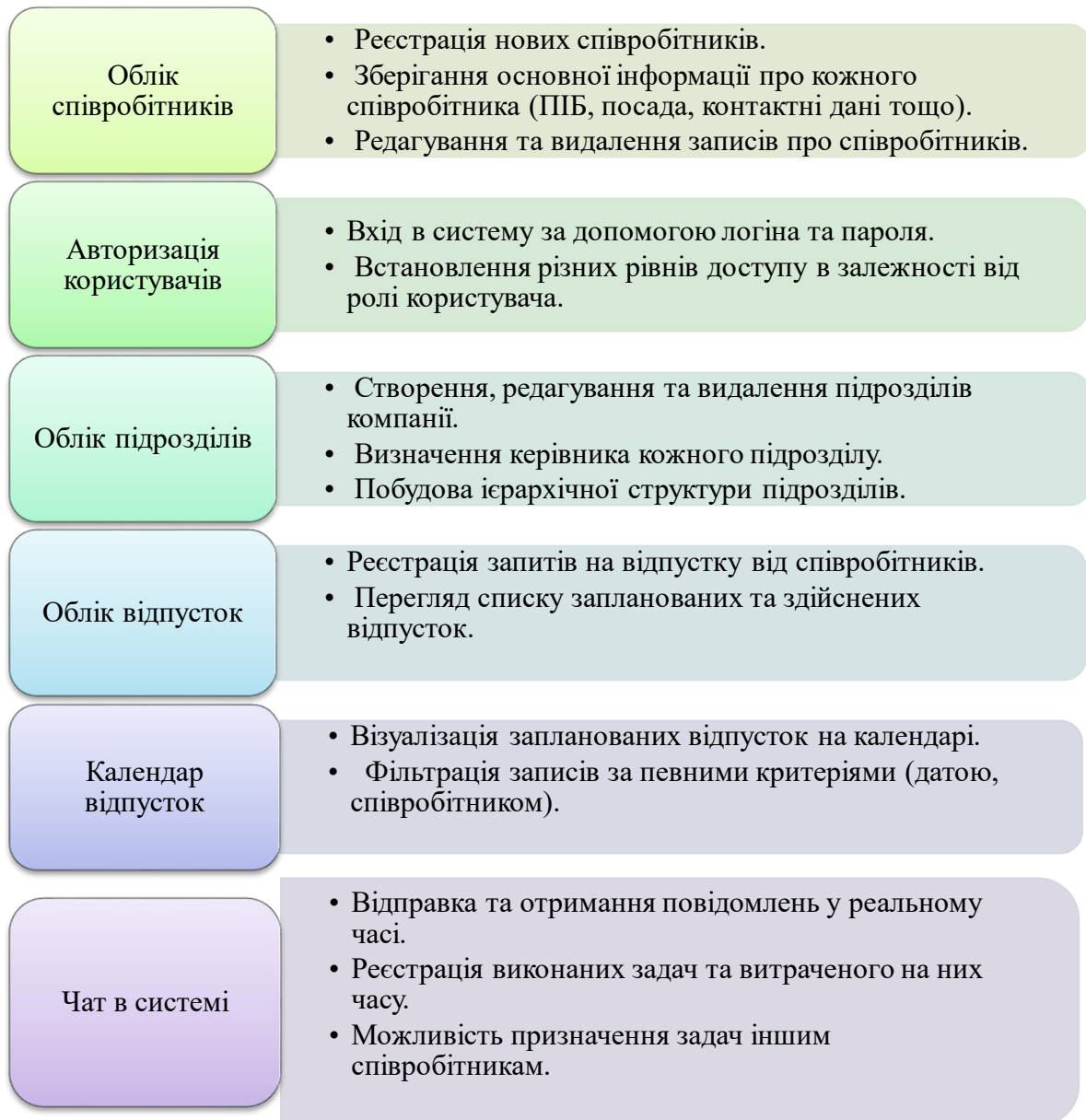


Рис. 2.2. Функціональні можливості системи

Функція чату в системі дозволяє співробітникам спілкуватися в реальному часі, реєструвати виконані задачі та витрачений на них час, а також призначати задачі іншим співробітникам, що сприяє координації та ефективності робочих процесів.

Завдяки всім цим функціональним можливостям, система дозволяє компанії ефективно управляти своїм персоналом, забезпечуючи високу продуктивність роботи та задоволеність співробітників.

## **2.2. Проектування структури бази даних**

Для забезпечення функціональності розробленої системи критично важливим є наявність ефективної структури бази даних, яка б стала надійним фундаментом для усіх операцій з даними. Проектування структури бази даних - це процес, що вимагає детального планування та аналізу, оскільки від його якості залежить швидкість, надійність та зручність майбутньої системи.

Проектована структура бази даних передбачає створення різноманітних таблиць для зберігання інформації, які мають бути чітко організовані та оптимізовані. Кожна таблиця має містити специфічну для себе інформацію, наприклад, дані про співробітників, підрозділи, відпустки тощо.

Окрім створення таблиць, важливим етапом проектування є визначення зв'язків між ними. Зв'язки допомагають уникнути дублювання даних, забезпечують цілісність інформації та спрощують процеси пошуку та аналізу даних. Вони можуть бути один-до-одного, один-до-багатьох або багато-до-багатьох, в залежності від специфіки зберігаємої інформації та бізнес-процесів компанії.

Нижче представлена проектована структура бази даних (рис. 2.3):

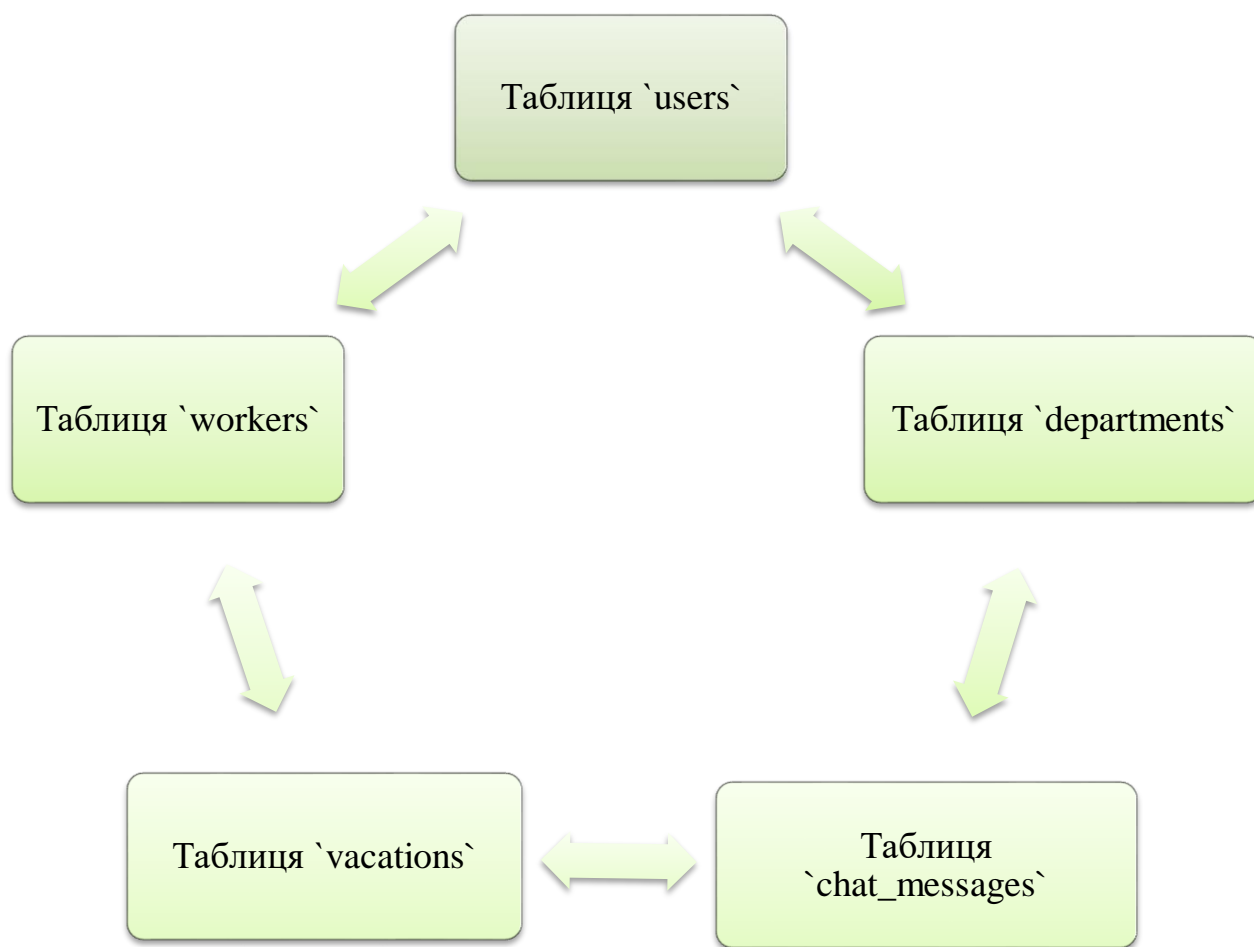


Рис. 2.3. Проектована структура бази даних

Таблиця **chat\_messages** призначена для зберігання повідомлень у чаті, включаючи унікальний ідентифікатор повідомлення, ідентифікатор користувача, текст повідомлення, час, затрачений на виконання завдання, та мітку часу створення повідомлення. Ця таблиця дозволяє тримати запис всіх комунікацій в системі, а також відслідковувати продуктивність співробітників.

- `id` - унікальний ідентифікатор повідомлення.

- `user\_id` - ідентифікатор користувача, який відправив повідомлення;

- `message` - текст повідомлення;

- `task\_time` - час, затрачений на виконання завдання;

- ``timestamp`` - мітка часу створення повідомлення.

Таблиця **departments** містить інформацію про підрозділи компанії, включаючи унікальний ідентифікатор, назву підрозділу, ідентифікатор керівника та ідентифікатор батьківського підрозділу. Це дозволяє створити чітку ієрархічну структуру організації.

- ``department_id`` - унікальний ідентифікатор підрозділу;
- ``department_name`` - назва підрозділу;
- ``head_id`` - ідентифікатор керівника підрозділу;
- ``parent_department_id`` - ідентифікатор батьківського підрозділу.

Таблиця **users** зосереджена на даних користувачів системи, зберігаючи інформацію, таку як унікальний ідентифікатор, ім'я користувача, пароль та погодинну ставку. Ця таблиця є основою для авторизації та аутентифікації користувачів в системі.

- ``id`` - унікальний ідентифікатор користувача;
- ``username`` - ім'я користувача;
- ``password`` - пароль користувача;
- ``hourly_rate`` - погодинна ставка користувача.

Таблиця **vacations** відповідає за облік відпусток співробітників, зберігаючи інформацію про унікальний ідентифікатор відпустки, ідентифікатор співробітника, тип відпустки та дати початку та закінчення відпустки. Це дозволяє ефективно планувати робочий час та ресурси.

- ``vacation_id`` - унікальний ідентифікатор відпустки;
- ``worker_id`` - ідентифікатор співробітника;
- ``vacation_type`` - тип відпустки (наприклад, щорічна, медична тощо);
- ``start_date`` - дата початку відпустки;
- ``end_date`` - дата закінчення відпустки.

Таблиця **workers** зберігає основну інформацію про співробітників, таку як унікальний ідентифікатор, повне ім'я та інші релевантні дані, а також ідентифікатор підрозділу, до якого належить співробітник.



- `id` - унікальний ідентифікатор співробітника;
- `full\_name` - повне ім'я співробітника;
- ... (інші поля для зберігання інформації про співробітника;
- `department\_id` - ідентифікатор підрозділу, до якого належить співробітник.

Відносини між таблицями базуються на принципах відносинних баз даних, зокрема на використанні зовнішніх ключів для забезпечення цілісності даних.

Ця структура бази даних дозволяє ефективно зберігати, обробляти та відображати інформацію в системі, а також забезпечує гнучкість для подальшого розвитку та масштабування.

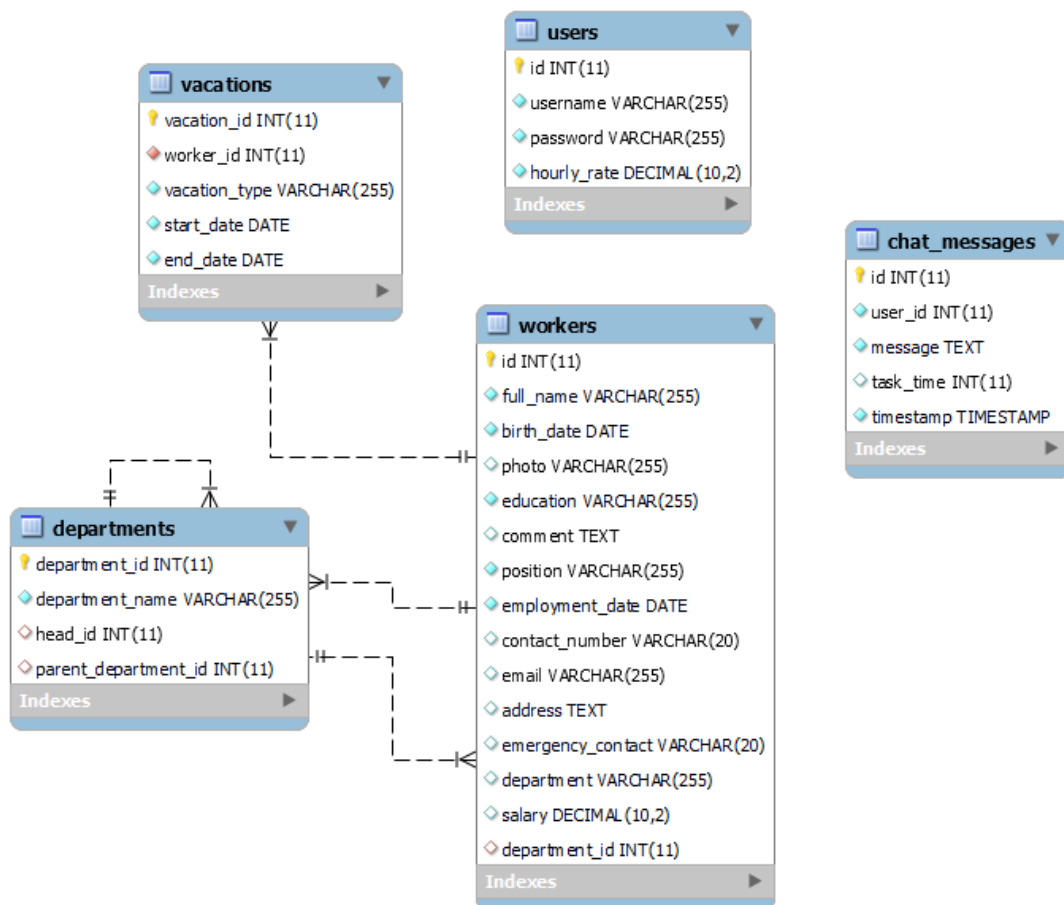


Рис. 2.4. Діаграма зв'язків між таблицями бази даних

Висновок з аналізу проекрованої структури бази даних свідчить про її комплексність та гнучкість, що дозволяє ефективно управляти великим обсягом інформації в системі. Кожна таблиця чітко спроектована для зберігання специфічних даних, таких як інформація про співробітників, підрозділи, відпустки та комунікації в чаті, забезпечуючи організованість та систематизацію даних.

Використання зовнішніх ключів та встановлення відносин між таблицями дозволяє забезпечити цілісність даних, уникнути дублювання інформації та спростити процеси пошуку та аналізу даних. Така структура також сприяє гнучкості системи, дозволяючи легко адаптуватися до можливих змін або розширень у майбутньому.

Загалом, проектована структура бази даних виглядає добре організованою та продуманою, готовою ефективно відповідати на вимоги системи та забезпечити надійну основу для її подальшого розвитку та оптимізації.

### **2.3. Розробка архітектури та інтерфейсу системи**

Підрозділ 2.3, присвячений розробці архітектури та інтерфейсу системи, акцентує увагу на ключових принципах конструкції системи, які спрямовані на забезпечення її гнучкості, масштабованості та надійності. Обравши веб-орієнтовану архітектуру, розробники використовують стандартні технології індустрії, що є важливим кроком для інтеграції сучасних технологічних рішень та підтримки актуальності системи в довготривалій перспективі.

Веб-орієнтована архітектура дозволяє забезпечити універсальний доступ до системи, роблячи її доступною з будь-якого пристрою, який підтримує веб-браузер. Це означає, що користувачі можуть легко і швидко отримати доступ до необхідної інформації або функціональних можливостей системи, не залежачи від конкретних технічних характеристик або обмежень їх пристроїв.

Такий підхід також сприяє масштабованості системи, оскільки дозволяє легко адаптувати архітектуру до змінюваних вимог та обсягів даних, а також до інтеграції нових функціональних можливостей у майбутньому.

Сучасний світ неможливо уявити без інформаційних систем, які спрощують, оптимізують та автоматизують багато аспектів нашого життя та роботи. У контексті організацій та бізнесу, ефективні інформаційні системи стають ключовим фактором успіху, допомагаючи управляти ресурсами, оптимізувати процеси та підвищувати продуктивність. Далі зосередимося на архітектурі однієї з таких систем. Ця частина документації детально розглядає технічні аспекти системи, включаючи клієнтську та серверну частини, та технології, які були використані для їх реалізації.

Архітектура системи розроблена з урахуванням найкращих практик та сучасних технологічних рішень, що має на меті забезпечити її надійність, гнучкість та масштабованість. Використання веб-орієнтованої архітектури сприяє універсальності доступу та забезпечує широкі можливості для подальшого розвитку та інтеграції з іншими системами та сервісами.

Клієнтська частина системи базується на використанні мов HTML та CSS, які є стандартними технологіями для розробки веб-інтерфейсів. Для підвищення продуктивності та спрощення процесу розробки використовується фреймворк Bootstrap. Цей фреймворк є потужним інструментом, який надає розробникам готові компоненти стилізації та інші інструменти для швидкого та ефективного створення адаптивних та мобільно-орієнтованих веб-сайтів.

Серверна частина системи реалізована на мові програмування PHP, яка відома своєю надійністю та гнучкістю у веб-розробці. Для зберігання та управління даними використовується система управління базами даних MySQL. PHP та MySQL взаємодіють між собою, забезпечуючи обробку запитів до бази даних та відображення результатів користувачеві. Така взаємодія дозволяє системі ефективно обробляти, зберігати та представляти необхідну інформацію, забезпечуючи високу продуктивність та надійність роботи системи.

Для досягнення високої якості користувацького досвіду та інтуїтивно зрозумілого інтерфейсу у системі було вирішено використовувати фреймворк Bootstrap. Цей вибір обумовлений бажанням створити сучасний, адаптивний

дизайн, який автоматично налаштовується під різні типи пристроїв, від мобільних телефонів до настільних комп'ютерів, забезпечуючи зручний та ефективний доступ до функціональності системи.

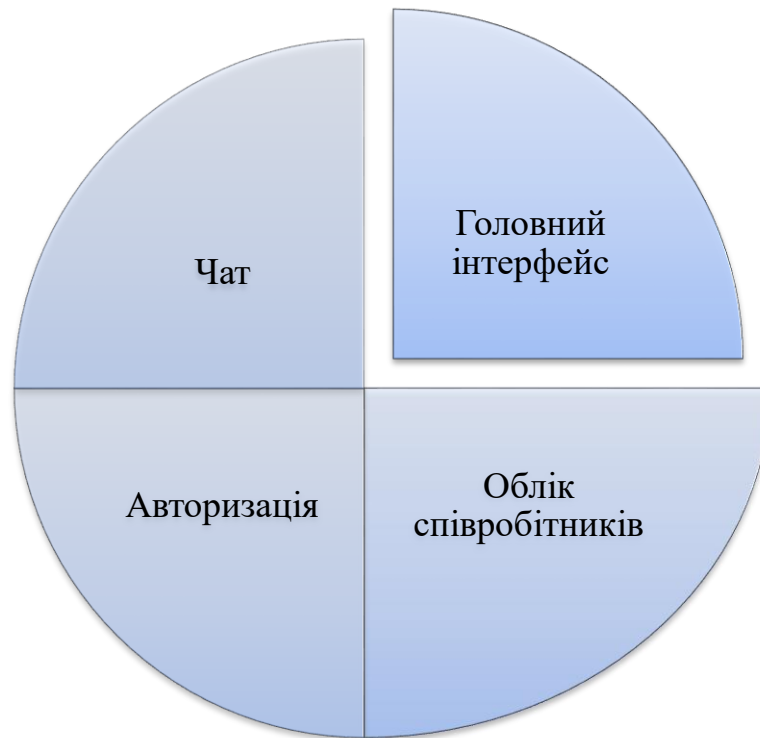


Рис. 2.5. Складові інтерфейсу користувача

Система має декілька ключових компонентів інтерфейсу. По-перше, сторінка авторизації, яка є входом до системи, де користувачі можуть ввести свої облікові дані для доступу до функціональних можливостей системи. По-друге, головний інтерфейс, який відображається після успішної авторизації, і який надає доступ до основних функцій системи, таких як облік співробітників, облік підрозділів, календар відпусток та інші.

Окрема увага приділена розробці інтерфейсу для обліку співробітників, який дозволяє з легкістю переглядати, додавати, редагувати та видаляти інформацію про співробітників. Також в системі присутній чат, який забезпечує

можливість для користувачів спілкуватися у реальному часі, обмінюватися інформацією, ділитися статусом виконання задач.

Bootstrap відіграє ключову роль у досягненні цілей, пов'язаних з удобством та інтуїтивністю інтерфейсу, допомагаючи розробникам створити візуально привабливий та зручний для користувача дизайн, який сприяє ефективній взаємодії користувачів з системою.

#### **2.4. Реалізація основних модулів системи**

Система управління персоналом є невід'ємною частиною сучасного бізнес-процесу, що допомагає організаціям оптимізувати роботу своїх команд, забезпечуючи ефективне використання людських ресурсів. Вона включає в себе ряд модулів, які адресують конкретні завдання і потреби: від слідкування за відпустками співробітників до управління структурою департаментів компанії. В цьому підрозділі ми докладно розглянемо кожний з цих модулів, їх функціональність та особливості реалізації.

Для забезпечення безпеки доступу до системи та захисту конфіденційних даних користувачів було реалізовано модуль авторизації.

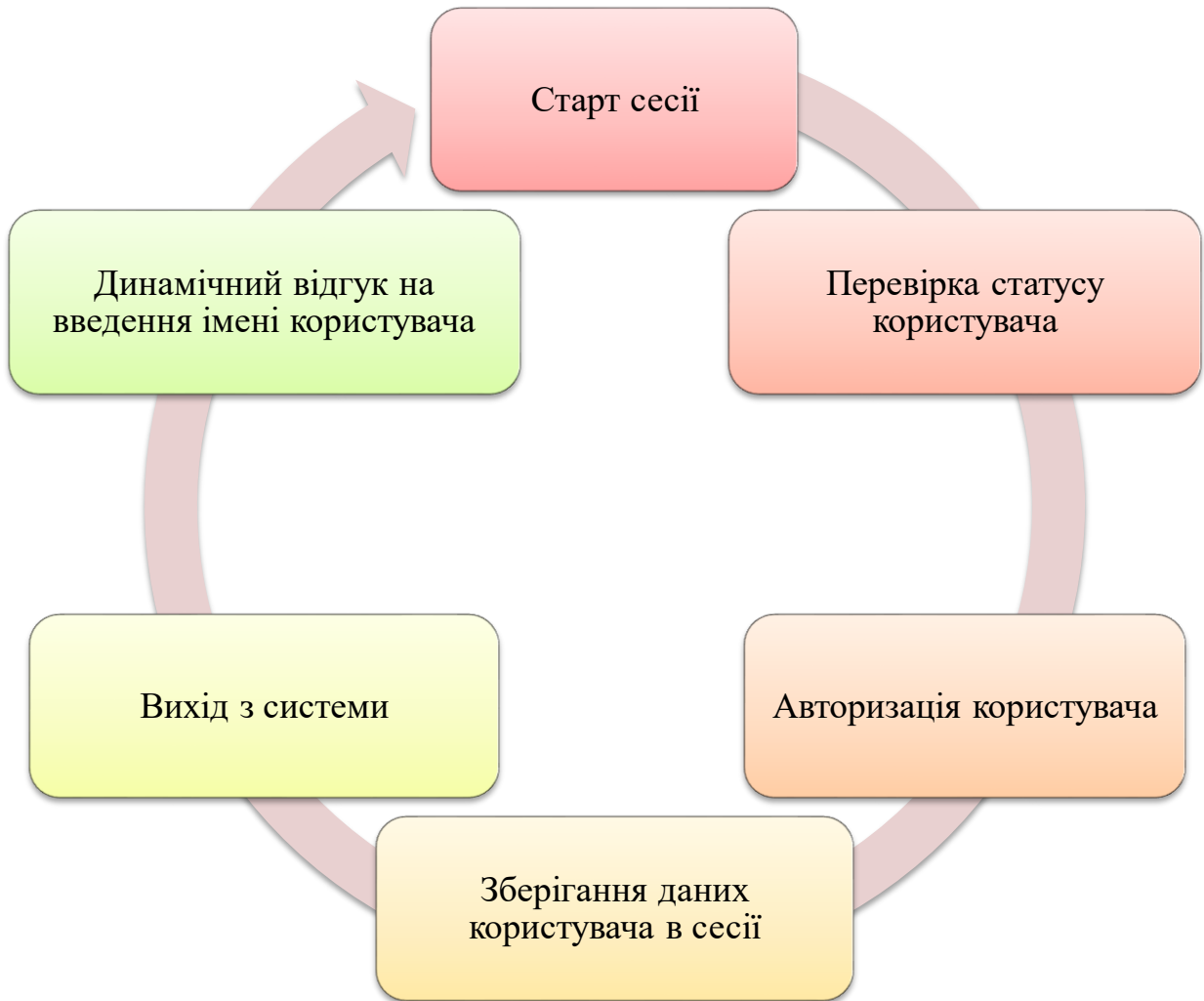


Рис. 2.6. Складові модуля авторизації

Сесія користувача ініціалізується при кожному відкритті веб-сторінки, створюючи унікальний ідентифікатор сесії, який дозволяє системі відслідковувати активність користувача та зберігати інформацію про нього протягом всього часу його перебування на сайті. Це необхідно для забезпечення безперервності роботи користувача з системою та збереження даних, пов'язаних з конкретним користувачем, між різними сторінками сайту.

Функція перевірки статусу користувача використовується для визначення, чи є користувач залогіненим. Це дозволяє системі визначити, чи має користувач права доступу до певних ресурсів системи, та відповідно регулювати доступ до функціональних можливостей.

Процес авторизації включає в себе введення користувачем своїх облікових даних (логін та пароль), які потім перевіряються системою. Паролі користувачів зберігаються в базі даних у зашифрованому вигляді, що забезпечує додатковий рівень безпеки.

# Вхід

Ім'я користувача

Пароль

Увійти

Зареєструватися

Рис. 2.7. Форма входу користувача

При спробі входу до системи дані форми відправляються на сервер, де відбувається перевірка імені користувача та пароля. Це здійснюється за допомогою функцій **authenticateUser** та **authenticateAuthor**.

Пароль користувача зберігається в зашифрованому вигляді, і перевірка автентичності виконується за допомогою функції **password\_verify**.

Після успішної авторизації дані користувача зберігаються в сесії, що дає можливість доступу до цих даних на інших сторінках системи без необхідності повторної авторизації.

Для виходу з системи використовується функція **logoutUser()**, яка очищує дані користувача в сесії.

За допомогою AJAX запиту до **check\_user.php** система динамічно реагує на введене ім'я користувача, показуючи або приховуючи поле вводу пароля в залежності від того, чи є користувач у базі даних.

Реалізація модуля авторизації забезпечує надійний захист від несанкціонованого доступу до системи та дозволяє легко та швидко ідентифікувати користувачів.

Однією з основних функцій будь-якої системи, що вимагає авторизації, є можливість реєстрації нових користувачів. Наша система надає адміністратору можливість додавання нових користувачів через спеціалізований інтерфейс.

## Додати користувача

Ім'я користувача

Ставка (грн./година)

Пароль

Додати

Рис. 2.8. Додавання нового користувача

Для забезпечення безпеки та ефективності процесу реєстрації користувачів в системі, було реалізовано ряд ключових функцій та обмежень.



1. **Обмеження доступу до модуля реєстрації.** Перед наданням доступу до форми реєстрації система перевіряє, чи є користувач залогіненим, а також чи має він відповідні права на додавання нових користувачів (в даному випадку, ID користувача повинен бути менше або рівним 10).

2. **Форма реєстрації.** Інтерфейс реєстрації є простим і інтуїтивно зрозумілим. Адміністратору потрібно ввести ім'я користувача, його ставку в годину та пароль. Після введення даних, вони передаються на сервер для обробки.

3. **Обробка форми реєстрації.** При натисканні на кнопку «Додати», дані форми відправляються на сервер методом POST. Сервер отримує дані, обробляє їх та додає нового користувача до бази даних.

4. **Хешування паролю.** Однією з важливих функцій, яку виконує серверна частина під час обробки форми реєстрації, є хешування паролів. Це робиться для забезпечення безпеки даних користувачів. У нашій системі використовується PHP-функція **password\_hash** для генерації безпечного хешу паролю.

5. **Додавання користувача до бази даних.** Після хешування паролю, система виконує SQL-запит до бази даних, щоб додати нового користувача. Це здійснюється за допомогою функції **registerUser**, яка формує SQL-запит та передає його до бази даних.

Завдяки реалізації цього модуля, адміністратор системи має можливість швидко та просто реєструвати нових користувачів, забезпечуючи при цьому найвищий рівень безпеки їхніх особистих даних.

Модуль обліку робітників призначений для адміністративного управління інформацією про робітників в системі. Завдяки цьому модулю, адміністратор може переглядати, додавати, редагувати та видаляти дані робітників.

## Список співробітників

Додати робітника

ID	ФІО	Посада	Дата прийому	Дії
1	Щербина Олександр Петрович	Змішувальник	2023-08-24	<a href="#">Деталі</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
2	Пікалов Олександр Борисович	Змішувальник	2021-05-04	<a href="#">Деталі</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
3	Шевченко Тарас Григорович	Автор	1994-08-01	<a href="#">Деталі</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
4	Іванов Іван Іванович	Менеджер	2020-01-01	<a href="#">Деталі</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
5	Петров Петро Петрович	Маркетолог	2019-02-15	<a href="#">Деталі</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
6	Сидоров Сидір Сидорович	Розробник	2018-03-20	<a href="#">Деталі</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>
7	Козлов Олександр Олександрович	Бухгалтер	2017-04-05	<a href="#">Деталі</a> <a href="#">Редагувати</a> <a href="#">Видалити</a>

Рис. 2.9. Модуль список співробітників

Кожен робітник відображається в таблиці, де вказано його ID, ПІБ, посаду та дату прийому на роботу. Для зручності адміністрації, зроблено кнопки дій біля кожного робітника: деталі, редагування та видалення.

Таблиця 2.1.

### Зміст модуля списку співробітників

№	Категорія	Деталі
1	Додавання нового робітника	Зверху таблиці розміщена кнопка «Додати робітника», яка веде на форму додавання нового робітника.
2	Перегляд деталей робітника	Кнопка «Деталі» веде на окрему сторінку, де відображена детальна інформація про вибраного робітника.
3	Редагування даних робітника	Кнопка «Редагувати» веде на сторінку редагування даних робітника, де можна змінити інформацію про нього.

№	Категорія	Деталі
4	Видалення робітника	Кнопка «Видалити» видаляє інформацію про робітника з бази даних після підтвердження дії.
5	Отримання списку робітників з бази даних	Функція <code>getWorkers</code> відповідає за отримання списку робітників з бази даних. Використовуючи SQL-запит, ця функція вибирає ідентифікатор, ПІБ, посаду та дату прийому кожного робітника з таблиці <code>workers</code> .

Ефективне управління персоналом є запорукою успішної діяльності будь-якої компанії. Для автоматизації процесів кадрового обліку використовується спеціальний модуль, який дозволяє адміністратору легко керувати даними про робітників компанії.

Цей модуль реалізує такі основні функції:

1. Додавання нових робітників в систему. За допомогою спеціальної форми адміністратор вводить повні відомості про нового співробітника – ПІБ, дату народження, посаду, дату прийняття на роботу тощо. Ці дані зберігаються в базі даних;
2. Перегляд детальної інформації про робітника. Адміністратор може в будь-який момент подивитися повний профіль працівника, щоб отримати вичерпні відомості про нього;
3. Редагування даних робітника в разі їх зміни. За потреби адміністратор може внести коригування в інформацію про співробітника, наприклад змінити посаду, дату народження тощо;
4. Видалення робітника з системи при звільненні. Коли працівник припиняє трудові відносини з компанією, його дані можуть бути видалені з бази даних;

5. Отримання загального списку робітників шляхом вибірки з бази даних. Ця функція дозволяє швидко сформувати актуальний перелік усіх працівників компанії.

Завдяки цьому модулю керівництво компанії має постійний доступ до повної та достовірної інформації про персонал. Автоматизація кадрового обліку значно полегшує управління робітниками та підвищує ефективність роботи компанії в цілому.

Модуль створення та редагування даних про робітників є важливою складовою системи управління персоналом на підприємстві. Він дозволяє автоматизувати процеси обліку та контролю кадрів, підвищуючи ефективність роботи кадрової служби. цей модуль є базовим інструментом для ведення кадрового обліку, контролю та аналізу персоналу, що спрощує роботу HR-відділу та підвищує ефективність управління кадрами.

# Додати співробітника

П.І.Б.

Дата народження

Посада

Підрозділ

Дата прийому на роботу

Освіта

Коментар

Фото

 файл не вибран

Рис. 2.10. Додавання робітника

Представлена форма для вводу основної інформації про робітника, зокрема П.І.Б., дата народження, посада, дата прийому на роботу, освіта, коментар, фото та підрозділ.



Рис. 2.11. Складові модуля створення та редагування даних про робітників

Фотографія робітника завантажується на сервер в папку 'uploads/workers/' з іменем, яке формується на основі імені робітника, і потім додається до бази даних.

Функція **addWorker** вставляє запис про робітника в таблицю 'workers'. Після успішного додавання користувач перенаправляється на сторінку зі списком робітників.

Сторінка з формою редагування завантажує існуючу інформацію про робітника з бази даних. Користувач може змінити будь-яку інформацію, включаючи фотографію.

Якщо користувач вирішив змінити фото, нове фото завантажується на сервер і замінює старе.

Функція **updateWorker** вносить зміни до існуючого запису робітника в таблиці 'workers'. Після успішного редагування користувач перенаправляється на сторінку зі списком робітників.

Обидва модулі мають перевірку дозволів. Якщо користувач не авторизований, він буде перенаправлений на головну сторінку.

## 2.4. Модуль інформації про співробітника

Модуль інформації про співробітника дає можливість користувачам системи отримувати детальну інформацію про конкретного співробітника, а також відслідковувати його відпустки.

### Деталі про співробітника

ІД	1
П.І.Б.	Щербина Олександр Петрович
Дата народження	1989-05-20

### Заплановані відпустки

Заплановані відпустки відсутні.

Запланувати відпустку

Повернутись

Редагувати

Рис. 2.12. Модуль інформація про співробітника

Виводиться таблиця із основною інформацією про співробітника (такою як ІД, П.І.Б., дата народження та інші). Якщо у співробітника немає запланованих відпусток, виводиться відповідне повідомлення. В іншому випадку відображається список його відпусток.



Рис. 2.13. Складові модуля інформації про співробітника

Біля кожної відпустки є кнопки для редагування або видалення відпустки. Редагування відпустки перенаправляє користувача на відповідну сторінку з формою, де можна змінити деталі відпустки.

Видалення відпустки передбачає підтвердження видалення, щоб уникнути помилкового видалення.

Кнопка «Запланувати відпустку» відкриває модальне вікно з формою для додавання нової відпустки. У формі вказуються тип відпустки, дата початку та дата закінчення відпустки.

Функція **addVacation** додає нову відпустку в базу даних.



Кнопка «Повернутись» дозволяє користувачеві повернутися до списку всіх співробітників. Кнопка «Редагувати» перенаправляє користувача на сторінку редагування інформації про співробітника.

**getVacationsByWorker** – повертає список всіх відпусток конкретного співробітника. **getWorker** – отримує інформацію про конкретного співробітника за його ID.

Модуль інформації про співробітника є корисним для HR-менеджерів або керівників, яким потрібно швидко отримати детальну інформацію про співробітника та його відпустки. Завдяки цьому модулю можна легко додавати, редагувати або видаляти відпустки конкретного співробітника.

Модуль департаментів дає можливість переглядати, додавати та редагувати інформацію про різні підрозділи в організації. Система підтримує ієрархічну структуру департаментів, де кожен підрозділ може мати підлеглі підрозділи.

## Підрозділи

- Адміністрація (Керівник: Пугачов Роман Романович) [Редагувати](#) [Додати підрозділ](#)
  - Маркетологи (Керівник: Щербина Олександр Петрович) [Редагувати](#) [Додати підрозділ](#)
    - Щербина Олександр Петрович (Посада: Змішувальник)
    - Іванов Іван Іванович (Посада: Менеджер)
    - Петров Петро Петрович (Посада: Маркетолог)
    - Сазонов Євген Євгенович (Посада: Маркетолог)
  - Відділ продажів (Керівник: Пікалов Олександр Борисович) [Редагувати](#) [Додати підрозділ](#)
    - Шевченко Тарас Григорович (Посада: Автор)
    - Горбунов Юрій Юрійович (Посада: Менеджер з продажу)
    - Гудков Олесь Олесьович (Посада: Менеджер з продажу)
  - Відділ маркетингу (Керівника не назначено) [Редагувати](#) [Додати підрозділ](#)
    - Сидоров Сидір Сидорович (Посада: Розробник)
    - Борисов Артем Артемович (Посада: Розробник)
    - Миронов Олексій Олексійович (Посада: Розробник)
  - Відділ розробки (Керівника не назначено) [Редагувати](#) [Додати підрозділ](#)
    - Король Олег Олегович (Посада: Спеціаліст з закупівель)
    - Ткачов Вадим Вадимович (Посада: Спеціаліст з закупівель)
  - Відділ фінансів (Керівника не назначено) [Редагувати](#) [Додати підрозділ](#)
    - Павлов Дмитро Дмитрович (Посада: HR-спеціаліст)
    - Шаров Андрій Андрійович (Посада: HR-спеціаліст)
  - Відділ закупівель (Керівника не назначено) [Редагувати](#) [Додати підрозділ](#)

Рис. 2.14. Модуль департаментів

Використовуючи функцію **displayDepartments**, департаменти відображаються у вигляді ієрархічного списку. Для кожного підрозділу виводиться його назва, інформація про керівника (якщо є) та списком співробітників, які працюють у цьому підрозділі. Користувачі можуть вибрати підрозділ для редагування, натиснувши на посилання «Редагувати» поряд із назвою підрозділу.

Для додавання нового підрозділу, користувач може використовувати посилання «Додати підрозділ» поряд із назвою батьківського підрозділу або головне посилання «Додати підрозділ» для додавання кореневого підрозділу.

Для кожного підрозділу відображається список співробітників, що працюють у цьому підрозділі. Кожен запис містить повне ім'я співробітника та його посаду.

Модуль департаментів є важливою частиною системи обліку співробітників. Завдяки цьому модулю, керівництво може легко стежити за структурою компанії, виділяти підрозділи, призначати керівників та виводити списки співробітників кожного підрозділу. Додаткова можливість додавати підлеглі підрозділи забезпечує гнучкість при структуруванні організації.

«Модуль департаментів: створення та редагування» дає можливість адміністраторам або користувачам з вищими привілеями додавати нові підрозділи в систему або редагувати існуючі.

# Додати підрозділ

Назва підрозділу

Керівник підрозділу

Не назначено

Підрозділ якому підпорядковується

Основний підрозділ

Додати

Рис. 2.15. Модуль підрозділів

Форма для створення департаменту містить поля: назва підрозділу, керівник підрозділу (з випадаючого списку співробітників) та вищестоящий підрозділ (якщо є). Після заповнення форми та натиснення кнопки «Додати», інформація про новий підрозділ зберігається в базі даних за допомогою функції **addDepartment**.

Щоб редагувати існуючий підрозділ, користувачу потрібно вибрати його зі списку департаментів. Форма редагування департаменту схожа на форму створення, але додатково виводиться поточна інформація про підрозділ. Коли користувач завершує редагування та натискає «Зберегти зміни», інформація в базі даних оновлюється за допомогою функції **editDepartment**.

**getDepartment** – повертає інформацію про конкретний підрозділ на основі його ID. **getAllWorkers** – повертає список усіх співробітників, який використовується для формування випадаючого списку керівників підрозділу.

**getAllDepartments** – повертає список усіх підрозділів для формування випадального списку вищестоящих підрозділів.

Модуль департаментів забезпечує гнучке керування ієрархічною структурою організації. Завдяки можливості додавати нові підрозділи або редагувати існуючі, адміністратори можуть легко адаптувати систему до змін в організаційній структурі компанії.

Модуль відпусток надає користувачам системи можливість відслідковувати всі заплановані та здійснені відпустки співробітників організації.

## Всі відпустки

ID відпустки	ФІО	Дата початку	Дата закінчення	Дії
2	Шевченко Тарас Григорович	2023-09-22	2023-09-27	<a href="#">Редагувати</a> <a href="#">Видалити</a>

[На головну](#)

Рис. 2.16. Модуль відпусток

Список відпусток представлений у вигляді таблиці, де для кожної відпустки вказані її ID, П.І.Б. співробітника, дата початку та дата закінчення відпустки. Для кожної відпустки передбачено можливість редагування або видалення за допомогою відповідних кнопок.

Коли користувач клікає на кнопку «Редагувати», він переходить на сторінку редагування відпустки, де може змінити дати відпустки або вибрати іншого співробітника. При кліку на кнопку «Видалити», користувачу виводиться повідомлення про підтвердження дії. Якщо він підтверджує дію, відпустка видалюється з бази даних.

**getAllVacations** – функція повертає список всіх відпусток з інформацією про співробітника. Вона використовує JOIN для об'єднання таблиць **vacations**

та **workers** за допомогою поля **worker\_id**, що дозволяє отримати П.І.Б. співробітника для кожної відпустки.

Модуль відпусток є важливим для управління ресурсами в організації. Він дозволяє HR-менеджерам або керівникам відслідковувати періоди відсутності співробітників та планувати роботу команди відповідно до цього. За допомогою цього модуля можна легко додавати, редагувати або видаляти відпустки в системі.

Модуль календаря відпусток надає користувачам можливість візуалізації запланованих відпусток співробітників на місяць. Це полегшує планування роботи і дозволяє менеджерам зорієнтуватися в наявності персоналу.

### Календар відпусток

Пн	Вт	Ср	Чт	Пт	Сб	Нд
1	2	3	4	5	6 Шевченко Тарас Григорович (6 днів)	7 Шевченко Тарас Григорович (6 днів)
8 Шевченко Тарас Григорович (6 днів)	9 Шевченко Тарас Григорович (6 днів)	10 Шевченко Тарас Григорович (6 днів)	11 Шевченко Тарас Григорович (6 днів)	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Рис. 2.17. Модуль календаря відпусток

Користувач може переглядати відпустки, заплановані на конкретний місяць та рік. Якщо місяць і рік не вказані, система використовує поточний місяць і рік. Дні місяця відображаються у форматі тижневого календаря, де кожний день містить інформацію про заплановані відпустки.

Функція **isVacation** перевіряє, чи є заплановані відпустки на конкретний день. Якщо відпустка запланована на цей день, то відображається ім'я співробітника та тривалість відпустки. Використовується цикл, щоб перевірити

кожну відпустку та визначити, чи потрапляє вона в діапазон дат, вказаних користувачем.

Календар показує кожен день місяця. Якщо на даний день є заплановані відпустки, вони відображаються під числом дня. Для кожної відпустки вказано ім'я співробітника та тривалість відпустки.

**getAllVacations** – функція отримує всі відпустки з бази даних. Відпустки асоціюються з конкретними співробітниками завдяки з'єднанню таблиць **vacations** та **workers**.

Модуль календаря відпусток є незамінним інструментом для HR-менеджерів або керівників великих команд. Він дозволяє швидко і легко оцінити загальний графік відпусток співробітників і планувати роботу команди відповідно.

Аналізуючи різноманітні модулі системи управління персоналом, можна впевнено стверджувати про їх значущість у сучасному корпоративному управлінні. Кожен модуль, розглянутий у цьому розділі, спрямований на рішення конкретних задач, які стоять перед організацією, і забезпечує ефективне використання людських ресурсів.

Система управління персоналом не лише раціоналізує процеси у компанії, але й сприяє підвищенню загальної продуктивності роботи, створюючи комфортні умови для співробітників та відкриваючи керівництву чітке бачення діяльності своєї команди. Завдяки таким системам можна не лише оптимізувати роботу, а й адаптуватися до змін, відслідковуючи потреби співробітників та забезпечуючи гнучкість у прийнятті рішень.

Таким чином, інвестування в розробку та імплементацію ефективної системи управління персоналом є стратегічно важливим кроком для будь-якої організації, яка прагне до успіху в умовах сучасного бізнес-середовища.

## 2.5. Тестування розробленої системи

Після завершення процесу розробки системи необхідно провести тестування для переконання у її надійності, стабільності та правильності

роботи. Тестування дозволяє ідентифікувати та виправляти можливі дефекти програмного забезпечення перед його впровадженням у продуктивне середовище.

Функціональне тестування було спрямоване на всебічну перевірку відповідності функціоналу розробленої системи керування персоналом вимогам, визначеним у технічному завданні.

Першим етапом функціонального тестування була перевірка коректності авторизації користувачів в системі. Були розглянуті такі тестові випадки:

- введення вірного логіна та пароля користувача. Після вводу даних мав відбуватися вхід в особистий кабінет користувача. Цей сценарій був успішно протестований для двох типів користувачів – адміністратора та звичайного співробітника;

- спроба входу за допомогою невірного логіна чи пароля. Система мала виводити повідомлення про помилку авторизації. Цей тест продемонстрував коректну роботу механізму авторизації;

- спроба отримати доступ до захищених розділів системи без авторизації. Перевірено, що без авторизації користувач не має доступу до confidential даних.

Наступним кроком було тестування функціоналу перегляду та роботи зі списком співробітників:

- перевірка коректного відображення даних (ПІБ, посада, відділ, дата прийому на роботу тощо) для всіх співробітників компанії. Вибірково переглянуті профілі 15 співробітників;

- перевірка можливості сортування списку співробітників за різними полями (прізвище, дата прийому, зарплата тощо);

- тестування фільтрації списку за відділами, посадами, статусом та іншими критеріями. Фільтри працюють коректно.

Окремим блоком було протестовано функціонал створення та редагування департаментів компанії:

- створення нового департаменту в системі. Перевірено можливість додавання назви, опису, призначення керівника тощо;
- редагування існуючого департаменту. Вибірково відредаговані 3 департаменти, змінені їх назви, описи та керівники;
- видалення департаменту. Після видалення перевірено, що департамент більше не відображається у списку.

Аналогічним чином було протестовано інші функції системи: реєстрація нових співробітників, налаштування зарплат і бонусів, робота з вакансіями, звітність тощо.

Таке ретельне функціональне тестування дозволило пересвідчитися, що система керування персоналом повністю відповідає вимогам та коректно виконує всі необхідні функції.

Вантажне тестування є вітальним компонентом у процесі оцінки продуктивності та стабільності систем, що мають функціонувати в умовах високого навантаження та інтенсивних користувацьких запитів.

Цей вид тестування має на меті визначити, наскільки добре система здатна працювати під навантаженням, вимірюючи її продуктивність, надійність та стабільність при великій кількості одночасних запитів або активних користувачів. Основна ціль полягає у виявленні можливих «вузьких місць», проблем з продуктивністю та інших питань, які можуть виникнути при масовому використанні системи.

Розглядаються наступні тестові випадки:

1. Симуляція одночасної роботи 100 користувачів: цей тестовий випадок спрямований на оцінку роботи системи при множинних одночасних сесіях користувачів, що дозволяє визначити, як система відповідає на паралельні запити та взаємодії;
2. Симуляція великої кількості запитів до бази даних: цей сценарій фокусується на аналізі здатності системи ефективно обробляти та оптимізувати велику кількість запитів до бази даних, що допомагає оцінити роботу системи в



умовах інтенсивного використання ресурсів бази даних та великої кількості даних.

Інтеграційне тестування є важливим етапом у процесі розробки та впровадження програмних систем, який зосереджений на перевірці взаємодії та координації різних компонентів системи між собою. Цей процес включає аналіз та оцінку того, як окремі модулі або компоненти системи працюють разом, забезпечуючи цілісність, консистентність та надійність обміну даними та функціональних процесів.

У рамках інтеграційного тестування було розглянуто декілька тестових випадків. Перший випадок фокусується на взаємодії модуля управління співробітниками і модуля відпусток, де було оцінено, як ці два компоненти координуються та взаємодіють для забезпечення правильного управління даними про співробітників та їх відпустки. Другий випадок оцінює синхронізацію даних між базою даних та інтерфейсом користувача, переконуючись, що інформація консистентно та правильно відображається та обробляється в різних частинах системи.

Під час процесу інтеграційного тестування було виявлено та успішно усунуто декілька менших дефектів, що сприяло підвищенню стабільності та надійності системи. Загалом, результати тестування показали, що система демонструє стабільну роботу, відповідаючи всім вимогам та критеріям, визначеним на етапі проектування. Таким чином, інтеграційне тестування підтвердило готовність системи до успішного впровадження та ефективного використання в реальних умовах роботи.

Загалом, всі три види тестування показали, що система готова до впровадження та може ефективно функціонувати в реальних умовах, відповідаючи всім встановленим вимогам та стандартам. Тестування підтвердило високий рівень стабільності, продуктивності та надійності розробленої системи.

## **Висновки до розділу 2**

В розділі 2 детально розглядається процес розробки інформаційної системи управління персоналом, яка організована через послідовність логічно структурованих етапів.

На початковому етапі, при визначенні вимог, було проведено глибокий аналіз потреб користувачів, на основі якого були сформульовані ключові функціональні можливості системи. Це дозволило забезпечити, що система буде відповідати реальним потребам користувачів та ефективно вирішувати поставлені задачі.

На етапі проектування бази даних було розроблено оптимізовану схему зберігання даних, яка враховує всі необхідні аспекти взаємодії та зв'язків між різними елементами даних, що забезпечує цілісність, консистентність та надійність інформації.

Розробка архітектури та інтерфейсу системи була здійснена з урахуванням принципів зручності та інтуїтивності для користувача, що сприяє швидкому освоєнню та ефективному використанню системи.

Реалізація основних модулів системи була виконана з урахуванням всіх визначених вимог та специфікацій, що забезпечило високу функціональність та гнучкість системи у відповідності до бізнес-процесів організації.

Тестування системи було організовано таким чином, щоб перевірити її на стабільність, надійність та відповідність всім встановленим вимогам та специфікаціям. Результати тестування підтвердили готовність системи до ефективної роботи в реальних умовах.

Загалом, завершена інформаційна система управління персоналом представляє собою комплексне, технологічно завершене рішення, яке спроможне автоматизувати, оптимізувати та удосконалити процеси управління персоналом, сприяючи підвищенню ефективності та конкурентоспроможності організації на ринку.

## РОЗДІЛ 3

### ВПРОВАДЖЕННЯ ТА ЕКСПЛУАТАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ПЕРСОНАЛОМ

#### 3.1. Рекомендації щодо впровадження системи

Успішне впровадження інформаційної системи управління персоналом вимагає чіткої стратегії, методичного підходу та адекватного навчання користувачів. Ось основні етапи та рекомендації щодо впровадження системи (рис. 3.1):



Рис. 3.1. Основні етапи впровадження системи

<b>Кафедра КІТ (47)</b>				<b>НАУ 23.16.20.000 ПЗ</b>			
<b>Виконав</b>	Першко Р.А.			ВПРОВАДЖЕННЯ ТА ЕКСПЛУАТАЦІЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ УПРАВЛІННЯ БУХГАЛТЕРСЬКИМИ РОЗРАХУНКАМИ ТА ОБЛІКУ ПРОДАЖІВ	<b>Літ.</b>	<b>Арк.</b>	<b>Аркушів</b>
<b>Керівник</b>	Колісник О.В.				Д	59	12
<b>Консульт.</b>					УС-211Мз		
<b>Н. Контр.</b>	Райчев І.Е.				122		

Розглядається підготовчий етап впровадження системи, який є критично важливим для успішної реалізації проекту. Перш за все, проводиться глибокий аналіз потреб організації: вивчається її структура, основні бізнес-процеси та специфіка діяльності. Це необхідно для того, щоб система, яка буде впроваджена, максимально відповідала реальним потребам організації та була оптимізована під її конкретні задачі та цілі.

Далі здійснюється технічна підготовка. На цьому етапі забезпечується наявність всього необхідного апаратного та програмного забезпечення, проводиться налаштування мережеских з'єднань та іншої інфраструктури, необхідної для функціонування системи.

Останнім кроком підготовчого етапу є формування команди впровадження. Команда повинна бути багатофункціональною та включати представників різних відділів: ІТ-спеціалістів, HR-менеджерів та інших ключових учасників проекту. Відповідальність команди полягає у координації та реалізації процесу впровадження системи, забезпечуючи її адаптацію та оптимізацію відповідно до специфіки та потреб організації.

Наступним є етап впровадження системи, який є вирішальним у процесі її реалізації та запуску. Починається цей етап з налаштування системи відповідно до специфікацій та унікальних потреб організації. Це включає конфігурацію різних параметрів, опцій та функціональних можливостей системи, щоб вона максимально ефективно відповідала вимогам бізнесу.

Далі проводиться імпорт даних, під час якого всі необхідні дані про персонал, які раніше були збережені в інших базах даних або документах, переносяться в нову систему. Це дозволяє забезпечити цілісність, актуальність та доступність інформації в межах єдиної системи.

Також на цьому етапі проводиться тестування системи, під час якого виявляються та усуваються різноманітні помилки, а також перевіряється поведінка системи в непередбачуваних ситуаціях та умовах високого навантаження.

Обов'язково акцентується увага на навчанні персоналу. Для цього розробляється детальна програма навчання, яка базується на інструкціях та руководствах, і організуються семінари та тренінги для різних груп користувачів. Особлива увага приділяється ключовим функціям системи. Крім того, забезпечується постійна підтримка користувачів на початковому етапі використання системи, щоб сприяти швидкому та ефективному її освоєнню.

Етап постійного моніторингу та оптимізації системи після її впровадження є ключовим для забезпечення неперервної ефективності, актуальності та безпеки системи.

Першим кроком є збір відгуків від користувачів. Це дозволяє вчасно ідентифікувати аспекти, які потребують удосконалення, та забезпечити адаптацію системи до змінюваних умов та потреб користувачів. Регулярний збір відгуків є важливим інструментом для підтримки високої якості сервісу та задоволення користувачів.

Далі, на основі зібраних відгуків та аналізу потреб організації, проводяться оновлення системи. Це може включати в себе технічні покращення, додавання нового функціоналу, оптимізацію існуючих процесів та інші аспекти, які сприяють підвищенню ефективності системи.

Також важливим елементом є аудит безпеки, який забезпечує захист системи від потенційних загроз та вразливостей. Регулярні перевірки та оновлення механізмів безпеки є ключовими для забезпечення надійності та захищеності даних в системі.

Загалом, постійний моніторинг та оптимізація дозволяють забезпечити стабільну, надійну та ефективну роботу системи, адаптовану до актуальних потреб та вимог організації.

Підсумовуючи, успішне впровадження системи управління персоналом потребує чіткого планування, налаштувань, навчання користувачів та постійного моніторингу. За дотриманням цих кроків ваша організація зможе максимально ефективно використовувати всі можливості системи.

## 3.2. Інструкція користувача системи

Підрозділ присвячений інструкції користувача системи електронного управління персоналом, розробленої з метою оптимізації та автоматизації ключових процесів управління персоналом організації. Інструкція структурована таким чином, щоб надати користувачам зрозумілі та конкретні вказівки щодо роботи з різними модулями системи, починаючи від процесу входу в систему та закінчуючи виходом з неї.

### 1) Вхід в систему

Для початку роботи з системою необхідно пройти авторизацію.

1. Відкрийте головну сторінку системи.
2. У вікні авторизації введіть ваш логін та пароль.
3. Натисніть кнопку «Увійти».

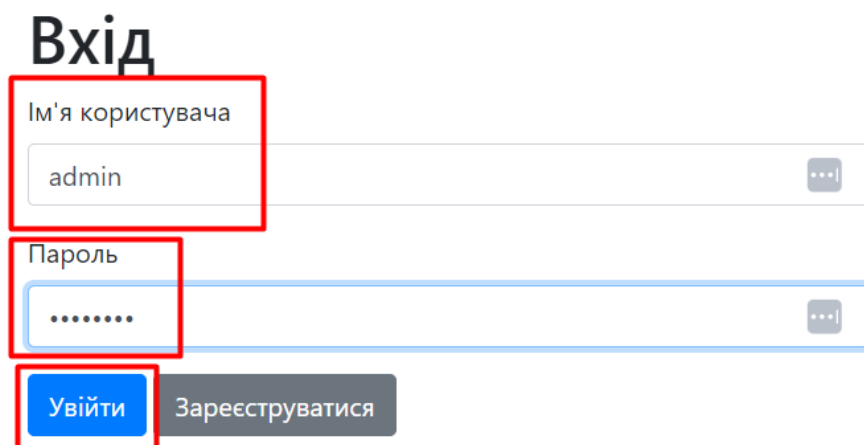


Рис. 3.2. Скріншот вікна авторизації

Якщо ви вперше в системі, зверніться до адміністратора для отримання даних для входу.

### 2) Робота з модулем «Підрозділи»

1. В головному меню оберіть пункт «Підрозділи».
2. Вам буде відображено список всіх підрозділів компанії.



## Підрозділи

- Адміністрація (Керівник: Пугачов Роман Романович) [Редагувати](#)  
[Додати підрозділ](#)
  - Маркетологи (Керівник: Щербина Олександр Петрович)  
[Редагувати](#) [Додати підрозділ](#)
    - Щербина Олександр Петрович (Посада: Змішувальник)
    - Іванов Іван Іванович (Посада: Менеджер)
    - Петров Петро Петрович (Посада: Маркетолог)
    - Сазонов Євген Євгенович (Посада: Маркетолог)
  - Відділ продажів (Керівник: Пікалов Олександр Борисович)  
[Редагувати](#) [Додати підрозділ](#)
    - Шевченко Тарас Григорович (Посада: Автор)

Рис. 3.3. Скріншот зі списком підрозділів

3. Для додавання нового підрозділу натисніть «Додати підрозділ», введіть необхідну інформацію та збережіть зміни.

Головна

## Додати підрозділ

Назва підрозділу  
Адміністрація

Керівник підрозділу  
Гудков Олесь Олесьович

Підрозділ якому підпорядковується  
Основний підрозділ

[Додати](#)

Рис. 3.4. Скріншот вікна додавання підрозділу

### 3) Робота з модулем «Відпустки»

1. В головному меню виберіть пункт «Відпустки».

2. Ви побачите календар відпусток, де відображені заплановані відпустки всіх співробітників.

## Календар відпусток

Пн	Вт	Ср	Чт	Пт	Сб	Нд
1	2	3	4	5	6 <b>Шевченко Тарас Григорович</b> (6 днів)	7 <b>Шевченко Тарас Григорович</b> (6 днів)
8 <b>Шевченко Тарас Григорович</b> (6 днів)	9 <b>Шевченко Тарас Григорович</b> (6 днів)	10 <b>Шевченко Тарас Григорович</b> (6 днів)	11 <b>Шевченко Тарас Григорович</b> (6 днів)	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Рис. 3.5. Скріншот календаря відпусток

3. Для планування нової відпустки виберіть співробітника та період відпустки, а потім натисніть «Запланувати відпустку».



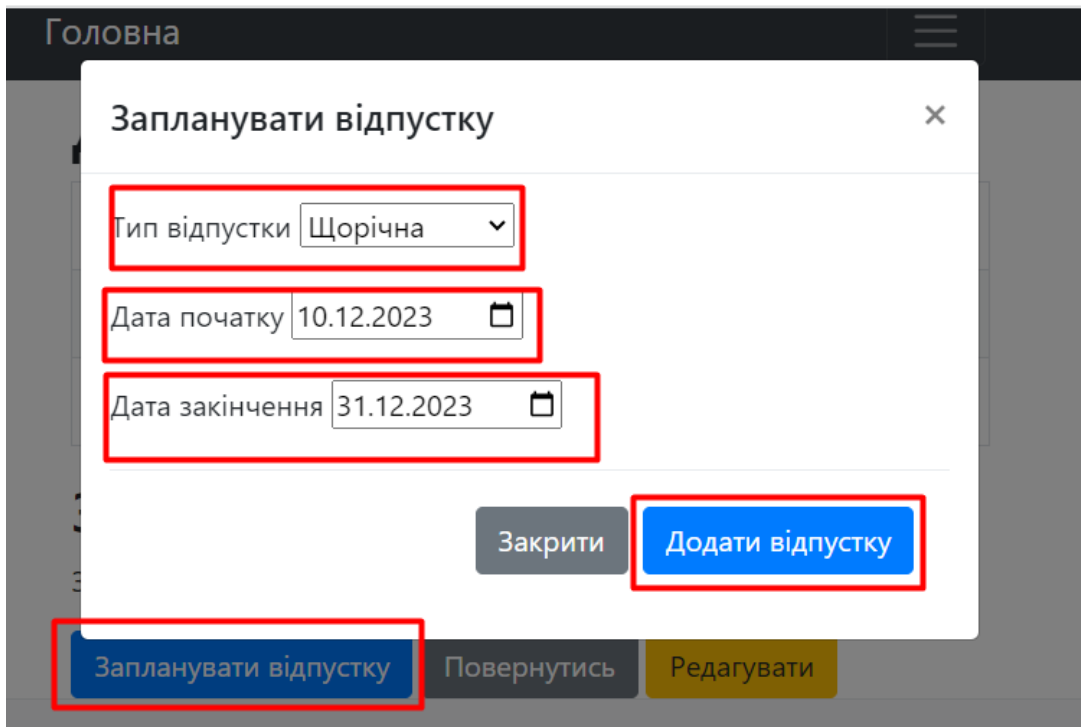


Рис. 3.6. Скріншот вікна планування відпустки

#### 4) Робота з модулем «Інформація про співробітника»

1. Виберіть в меню пункт «Список співробітників».
2. Знайдіть потрібного співробітника в списку та виберіть його.
3. Ви побачите детальну інформацію про співробітника та його заплановані відпустки.

### Деталі про співробітника

<b>ID</b>	2
<b>П.І.Б.</b>	Пікалов Олександр Борисович
<b>Дата народження</b>	1978-07-05

### Заплановані відпустки

Заплановані відпустки відсутні.

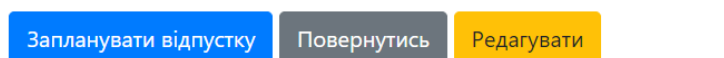
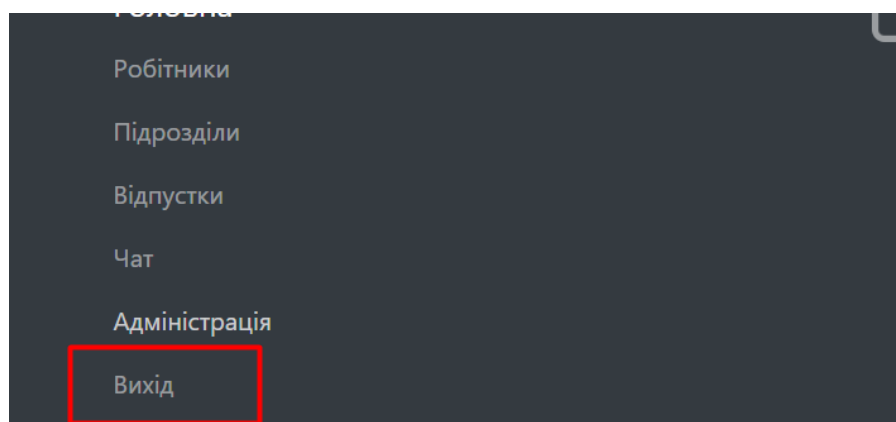


Рис. 3.7. Скріншот інформації про співробітника

#### 5) Вихід з системи

Для завершення роботи з системою натисніть на ваше ім'я користувача в верхньому правому куті екрана і оберіть «Вийти».



## Деталі про співробітника

Рис. 3.8. Скріншот вибору пункту «Вийти»

Дотримуючись цієї інструкції, ви зможете ефективно та без проблем працювати з електронною системою управління персоналом.

Таким чином, на початку інструкції користувачі отримують чіткі вказівки щодо процесу авторизації в системі, який є початковим етапом роботи з будь-якою електронною системою. Далі, в інструкції детально описано, як працювати з основними модулями системи, такими як «Підрозділи», «Відпустки» та «Інформація про співробітника». Кожен з цих розділів містить конкретні кроки та, за необхідності, супроводжується ілюстративними скріншотами для наглядності та кращого розуміння. Інструкція завершується розділом, присвяченим процесу виходу з системи, що є важливим для забезпечення безпеки користувацької інформації та даних системи.

### 3.3. Переваги розробленої системи у порівнянні з аналогами

Розроблена система управління персоналом має ряд переваг у порівнянні з аналогічними системами, які вже присутні на ринку. Наведемо декілька ключових переваг на рисунку 3.9.



Рис. 3.9. Переваги розробленої системи у порівнянні з аналогами

Таблиця 3.1 демонструє, що наша розроблена система має значні переваги у вигляді індивідуальності розробки та продуманості модулів. Це дозволяє системі бути більш адаптивною та ефективною у відповіді на унікальні потреби та виклики, з якими може зустрічатися організація у процесі управління персоналом.

Наша система відрізняється індивідуальністю підходу до розробки, що означає, що вона була спеціально розроблена з урахуванням конкретних потреб, вимог та особливостей організації-замовника. Це дозволяє системі бути

більш точно налаштованою під специфіку бізнес-процесів компанії, в той час як стандартні рішення часто є більш загальними та менш гнучкими щодо індивідуальних налаштувань.

Таблиця 3.1.

Порівняння нашої системи управління персоналом з існуючими аналогами  
(HRM)

Назва програми	Опис програми	Функціонал
<b>Система управління персоналом (HRM)</b>		
SAP SuccessFactors	Система управління персоналом, що дозволяє керувати процесами найму, оцінки та розвитку персоналу, забезпечувати доступ до даних про співробітників та аналізувати результати.	Керування процесами найму, оцінки та розвитку персоналу, забезпечення доступу до даних про співробітників, аналітика даних.
Oracle HCM Cloud	Система управління персоналом, що дозволяє керувати процесами найму, оцінки та розвитку персоналу, забезпечувати доступ до даних про співробітників та аналізувати результати.	Керування процесами найму, оцінки та розвитку персоналу, забезпечення доступу до даних про співробітників, аналітика даних.
Workday Human Capital Management	Система управління персоналом, що дозволяє керувати процесами найму, оцінки та розвитку персоналу, забезпечувати доступ до даних про співробітників та аналізувати результати.	Керування процесами найму, оцінки та розвитку персоналу, забезпечення доступу до даних про співробітників, аналітика даних.

Наша розроблена Система	Система управління персоналом, розроблена з урахуванням індивідуальних потреб організації, з фокусом на детальній проробці кожного модуля для максимальної ефективності та зручності користувачів.	Гнучкість у налаштуванні під конкретні потреби організації, детально пророблені модулі, що враховують специфіку HRM-процесів, висока адаптивність до змін у організаційних процесах.
-------------------------	--	--

Модулі нашої системи були детально розроблені та протестовані, щоб забезпечити їх високу ефективність, надійність та зручність використання. Кожен модуль розроблено з урахуванням найкращих практик та стандартів у сфері управління персоналом, що дозволяє досягти високої продуктивності та оптимізації робочих процесів.

У порівнянні з аналогами, наша система може виглядати більш пристосованою до конкретних потреб користувачів та організацій, оскільки вона надає більше можливостей для індивідуальних налаштувань та модифікацій.

Отже, розроблена нами система управління персоналом представляє собою високоякісне, індивідуалізоване рішення, яке відзначається глибокою проробкою кожного модуля та адаптацією під конкретні потреби організації. Ці переваги роблять нашу систему вельми конкурентоспроможною на ринку, надаючи клієнтам можливість оптимізувати та ефективно управляти своїми HRM-процесами з використанням сучасних, інноваційних технологій та підходів.

### **Висновки до розділу 3**

У третьому розділі дипломної роботи було детально розглянуто процес впровадження та експлуатації розробленої інформаційної системи управління персоналом.

Спершу були розроблені конкретні рекомендації щодо впровадження системи, які включають підготовчий етап, етап впровадження, навчання персоналу та постійний моніторинг та оптимізація системи. Ці рекомендації спрямовані на забезпечення плавного та ефективного процесу впровадження, адаптації користувачів та подальшої стабільної роботи системи.

Далі була розроблена детальна інструкція користувача системи, яка включає в себе поетапні вказівки з використання основних функціональних можливостей системи, що дозволяє користувачам швидко та ефективно освоїти систему.

Також було проведено аналіз переваг розробленої системи у порівнянні з аналогами. Розроблена система демонструє значні переваги завдяки індивідуальності розробки та продуманості модулів, що відповідають специфіці організаційних процесів та потреб користувачів.

Наша розроблена система управління персоналом володіє рядом ключових переваг, які відрізняють її на ринку HRM-систем. По-перше, індивідуальність розробки: система була створена з унікальним підходом, специфічно адаптованим під потреби та вимоги конкретної організації, що дозволяє максимально ефективно відповідати на її унікальні виклики та потреби. По-друге, продуманість модулів: кожен аспект системи був ретельно пророблений, щоб забезпечити гладкість, ефективність та зручність використання, від моменту впровадження до щоденного використання.

Загалом, третій розділ надає чітке розуміння процесу впровадження та експлуатації системи, підкреслюючи її переваги та готовність до ефективного використання в організаційних процесах управління персоналом.

## ВИСНОВКИ

У ході виконання дипломної роботи було успішно виконано ряд завдань, кожне з яких спрямоване на реалізацію ефективної системи управління персоналом.

1. Детальний аналіз ринку систем управління персоналом було проведено з метою вивчення сучасних тенденцій, можливостей та недоліків існуючих рішень. Це дослідження допомогло отримати цінні інсайти та визначити ключові аспекти для удосконалення в процесі розробки власної системи.

2. На основі проведеного аналізу були чітко визначені та сформульовані функціональні вимоги, які має відповідати нова система, забезпечуючи максимальну ефективність та зручність для кінцевого користувача.

3. Обрано сучасні технології та інструменти, які відповідають всім потребам проекту та забезпечують гнучкість, надійність та масштабованість розробленої системи.

4. Була розроблена логічна та фізична моделі бази даних, яка враховує всі необхідні зв'язки та взаємодії між елементами даних, забезпечуючи їх консистентність, інтеграцію та цілісність.

5. Система та її інтерфейс були розроблені з фокусом на зручності та інтуїтивності використання, враховуючи найкращі практики та стандарти UX/UI дизайну.

6. Проведено комплексне тестування системи, в ході якого були ідентифіковані та усунуті потенційні проблеми та помилки, що сприяло підвищенню стабільності та надійності системи.

7. На завершальному етапі були розроблені та сформульовані рекомендації щодо ефективного впровадження та оптимізації роботи системи в майбутньому.

Кожен з цих етапів був виконаний з максимальною відповідальністю та увагою до деталей, що дозволяє гарантувати високу якість та готовність розробленої системи до успішного впровадження та експлуатації.

Значення виконаної дипломної роботи велике і багатогранне, оскільки розроблена система управління персоналом відкриває нові можливості для організацій різного рівня та специфікації. Розроблена система є результатом глибокого аналізу існуючих рішень та тенденцій у сфері управління персоналом, що дозволяє їй бути актуальною, конкурентоспроможною та відповідати сучасним вимогам ринку.

Система розроблена з урахуванням індивідуальних потреб користувачів, що забезпечує високий рівень її адаптивності та гнучкості, дозволяючи організаціям оптимізувати та автоматизувати процеси управління персоналом з максимальною ефективністю. Розробка спрямована на підвищення продуктивності та ефективності роботи персоналу, поліпшення якості прийняття управлінських рішень та оптимізацію адміністративних процесів, що в свою чергу сприяє росту загальної ефективності організації.

Впровадження цієї системи може стати каталізатором для подальшого інноваційного розвитку організацій, стимулюючи пошук нових підходів, методів та інструментів управління персоналом.



## СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ

1. Байкарова О. О. Інформаційні технології – засіб оптимізації діяльності підприємств / О. О. Байкарова, Л. М. Тарасюк // Комп'ютерно-інтегровані технології : освіта, наука, виробництво. – 2013. – №11. – С. 177 – 182.
2. Вовчак І.С. Інформаційні системи та комп'ютерні технології в менеджменті : навч. посіб. / І. С. Вовчак. – Тернопіль : Карт-бланш, 2001. – 354 с.
3. Глівенко С. В. та ін. Інформаційні системи в менеджменті : навч. посіб. / С. В. Глівенко, С. В. Лапін, О. О. Павленко та ін. – Суми : ВТД «Університетська книга», 2005. – 407 с.
4. Знахур В. С. Інформаційний менеджмент та маркетинг : конспект лекцій / С.В. Знахур. – Харків : Вид. ХНЕУ. – 2009. – 131 с.
5. Жаворонкова Г. В. Інформаційне підприємництво: інновації, консалтинг, маркетинг. – К.: Національний авіаційний університет, 2003. – 366 с.
6. Жежнич П.І. Технології інформаційного менеджменту : навч. посіб. / П.І. Жежнич. — Львів : Львівська політехніка, 2010. — 260 с.
7. Куденко Н. В. Менеджмент – управління інформацією / Навч. посібн. – К: КДТЕУ, 1999. – 313 с.
8. Матвієнко О. В., Цивін М. Н. Інформаційний менеджмент: опорний конспект лекцій у схемах і таблицях. Київ, 2007. 200 с.
9. Матвієнко О. В. Основи інформаційного менеджменту: Навчальний посібник. Київ, 2004. 128 с.
10. Марусей Т. Впровадження CRM-систем у маркетингову діяльність підприємства / Т. Марусей // Економіка та держава. – 2016. – № 6. – С. 87–89.
11. Крупський А. Ю., Феоктистова Л. А. Інформаційний менеджмент : навч. посіб. /А. Ю. Крупський, Л. А. Феоктистова. – М. : Видавничоторгова корпорація «Дашков і К °», 2008. – 80 с.

12. Гужва В. М. Інформаційні системи і технології на підприємствах : навч. посіб. / В. М. Гужва. – К. : КНЕУ, 2001. – 400 с.
13. Вовчак І.С. Інформаційні системи та комп'ютерні технології в менеджменті : навч. посіб. / І. С. Вовчак. – Тернопіль : Карт-бланш, 2001. – 354 с.
14. Принципи вибору програмного забезпечення для вирішення різних завдань управління персоналом. URL: <https://studfile.net/preview/2398774/page:65/>.
15. Лисак В.М. Теоретичні аспекти автоматизації процесів збирання економічної інформації для управління підприємством // Вісник Хмельницького національного університету, - Хмельницький, 2008, № 5, Т. 2(119).
16. Фадєєва І.Г. Розвиток концептуальних засад автоматизованого аналітичного управління бізнес-процесами // Вісник Хмельницького національного університету, - Хмельницький, 2008, № 5, Т. 2(119).
17. Указ № 133/2017 – Указ Президента України "Про рішення Ради національної безпеки і оборони України від 28 квітня 2017 року "Про застосування персональних спеціальних економічних та інших обмежувальних заходів (санкцій)" від 15.05.2017 № 133/2017.
18. Майбутнє за технологіями: електронні програми управління фермою. URL: <https://kurkul.com/blog/74-maybutnye-za-tehnologiyami-elektronni-programi-upravlinnya-fermoyu>
19. Азарова А., Мороз О., Сторожа А. Моделювання системи підтримки прийняття рішень щодо покращення інноваційної діяльності підприємства. Вісник Хмельницького національного університету. 2012. № 6. Т. 1. С. 112–115.
20. Азарова А., Роїк О., Лобанкіна І. Впровадження та використання автоматизованих систем підвищення продуктивності праці на підприємстві за умов кризи. Економічний простір. 2010. № 42. С.125–132.

21. Гнатенко І. А. Логіка впровадження інноваційних заходів на регіональний ринок праці в умовах його циклічного розвитку / І. А. Гнатенко, В. О. Рубежанська // Бізнес Інформ. – 2017. – № 8. – С. 110–115.
22. Лук'яненко Д. І. Розвиток інформаційного менеджменту як наукової категорії / Д. І. Лук'яненко // Науковий вісник Полтавського університету економіки і торгівлі. Сер.: Економічні науки. – 2013. – № 1. – С. 183–187.
23. Ганущак-Єфіменко Л.М. CRM-система як ефективний інструмент розвитку готельного бізнесу в Україні. Вісник Київського національного університету технологій та дизайну. Серія : Економічні науки. 2017. № 4. С. 51–56.
24. Чайковська М.П. Перспективи гіпермедійної інтеграції CRM-систем. Економічний вісник університету. Збірник наукових робіт. 2012. № 18-2. С. 201–207.
25. Ліщинська Л.Б. Основні аспекти автоматизації роботи з клієнтами засобами CRM-систем. Вісник Хмельницького національного університету. Економічні науки. 2015. № 5(1). С. 206–209.
26. CRM-системи стали найбільшим сегментом ринку в 2017 році. URL: <https://news.finance.ua> (дата звернення 09.01.2019).
27. Результати дослідження ринку CRM в Україні. URL: <https://www.bitrix24.ua> (дата звернення 10.01.2019).
28. Сьомкіна Т.В., Литвинова О.В., Лобань О.О. Особливості моделей функціонування ІТ-компаній в Україні. Науковий вісник Ужгородського національного університету. Серія : Міжнародні економічні відносини та світове господарство. 2018. Вип. 19(3). С. 84–87.
29. Бутенко Н.В. Впровадження концепції CRM на промисловому ринку. Економіка та держава. 2011. № 3. С. 40–42.
30. Ушакова І.О. Соціальні мережі, як засіб впливу на взаємовідносини з клієнтами. Системи обробки інформації. 2012. Вип. 8. С. 54–58.

## Додаток А

### Функції HR CRM

```
<?php
require_once 'database.php';

if (session_status() == PHP_SESSION_NONE) {
    session_start();
}

function isLoggedIn() {
    return isset($_SESSION['user_id']);
}

function authenticateUser($username, $password) {
    global $db;
    $query = "SELECT * FROM users WHERE username = :username";
    $stmt = $db->prepare($query);
    $stmt->execute(['username' => $username]);
    $user = $stmt->fetch(PDO::FETCH_ASSOC);

    if ($user && password_verify($password, $user['password'])) {
        $_SESSION['user_id'] = $user['id'];
        $_SESSION['username'] = $user['username'];
        $_SESSION['user'] = $user;
        return true;
    }

    return false;
}
```

```
}
```

```
function userExists($username) {
```

```
    global $db;
```

```
    $query = "SELECT * FROM users WHERE username = :username";
```

```
    $stmt = $db->prepare($query);
```

```
    $stmt->execute(['username' => $username]);
```

```
    $user = $stmt->fetch(PDO::FETCH_ASSOC);
```

```
    return $user ? true : false;
```

```
}
```

```
function registerUser($username, $password) {
```

```
    global $db;
```

```
    $query = "INSERT INTO users (username, password) VALUES (:username,  
:password)";
```

```
    $stmt = $db->prepare($query);
```

```
    $result = $stmt->execute([
```

```
        'username' => $username,
```

```
        'password' => password_hash($password, PASSWORD_DEFAULT),
```

```
    ]);
```

```
    return $result;
```

```
}
```

```
function logoutUser() {
```

```
    unset($_SESSION['user_id']);
```

```
    unset($_SESSION['username']);
```

```
    unset($_SESSION['user']);
```

```

unset($_SESSION['author_id']);
unset($_SESSION['author_name']);
unset($_SESSION['author']);
}

function getUsers()
{
    global $db;

    $query = "SELECT * FROM users";
    $stmt = $db->prepare($query);
    $stmt->execute();

    return $stmt->fetchAll(PDO::FETCH_ASSOC);
}

function addUser($username, $password, $hourly_rate)
{
    global $db;

    $hashed_password = password_hash($password, PASSWORD_DEFAULT);
    $query = "INSERT INTO users (username, password, hourly_rate) VALUES
(:username, :password, :hourly_rate)";
    $stmt = $db->prepare($query);
    $stmt->execute([':username' => $username, ':password' => $hashed_password,
':hourly_rate' => $hourly_rate]);
}

function getUserById($id)
{
    global $db;

```

```

$query = "SELECT * FROM users WHERE id = :id";
$stmt = $db->prepare($query);
$stmt->execute([':id' => $id]);

return $stmt->fetch(PDO::FETCH_ASSOC);
}

function updateUser($id, $new_id, $username, $password, $hourly_rate)
{
    global $db;

    $hashed_password = password_hash($password, PASSWORD_DEFAULT);
    $query = "UPDATE users SET id = :new_id, username = :username, password =
:password, hourly_rate = :hourly_rate WHERE id = :id";
    $stmt = $db->prepare($query);
    $stmt->execute([':new_id' => $new_id, ':username' => $username, ':password' =>
$hashed_password, ':id' => $id, ':hourly_rate' => $hourly_rate]);
}

function deleteUser($id)
{
    global $db;

    $query = "DELETE FROM users WHERE id = :id";
    $stmt = $db->prepare($query);
    $stmt->execute([':id' => $id]);
}

/* Chat*/

```

```

function getChatMessages($userId = null) {
    global $db;
    $query = "SELECT chat_messages.*, users.username FROM chat_messages JOIN
users ON chat_messages.user_id = users.id ORDER BY chat_messages.timestamp";
    $stmt = $db->prepare($query);
    $stmt->execute();
    $messages = $stmt->fetchAll(PDO::FETCH_ASSOC);

    // ім'я користувача
    foreach ($messages as &$msg) {
        if (!empty($msg['username'])) {
            $msg['user'] = [
                'id' => $msg['user_id'],
                'username' => $msg['username']
            ];
        }
        unset($msg['username']);
    }

    if ($userId) {
        $messages = array_filter($messages, function ($msg) use ($userId) {
            return $msg['user']['id'] === $userId;
        });
    }

    return $messages;
}

```



```

function addChatMessage($user_id, $message, $task_time) {
    global $db;
    $query = "INSERT INTO chat_messages (user_id, message, task_time) VALUES
(:user_id, :message, :task_time)";
    $stmt = $db->prepare($query);
    $result = $stmt->execute([
        'user_id' => $user_id,
        'message' => $message,
        'task_time' => $task_time,
    ]);

    return $result;
}

```

```

function deleteChatMessage($id) {
    global $db;
    $query = "DELETE FROM chat_messages WHERE id = :id";
    $stmt = $db->prepare($query);
    $result = $stmt->execute(['id' => $id]);

    return $result;
}

```

```

function updateChatMessage($id, $message, $task_time) {
    global $db;
    $query = "UPDATE chat_messages SET message = :message, task_time =
:task_time WHERE id = :id";
    $stmt = $db->prepare($query);
    $result = $stmt->execute([
        'id' => $id,

```

```

        'message' => $message,
        'task_time' => $task_time,
    ];

    return $result;
}

/* Робітники */

function getWorkers() {
    global $db;

    $query = "SELECT id, full_name, position, employment_date FROM workers";
    $stmt = $db->prepare($query);
    $stmt->execute();

    return $stmt->fetchAll(PDO::FETCH_ASSOC);
}

function transliterate($text) {
    $translit = array(
        'А' => 'A', 'Б' => 'B', 'В' => 'V', 'Г' => 'G', 'Д' => 'D', 'Е' => 'E', 'Є' => 'Ye',
        'Ж' => 'Zh', 'З' => 'Z', 'И' => 'Y', 'І' => 'I', 'Ї' => 'Yi', 'Й' => 'Y', 'К' => 'K',
        'Л' => 'L', 'М' => 'M', 'Н' => 'N', 'О' => 'O', 'П' => 'P', 'Р' => 'R', 'С' => 'S',
        'Т' => 'T', 'У' => 'U', 'Ф' => 'F', 'Х' => 'Kh', 'Ц' => 'Ts', 'Ч' => 'Ch', 'Ш' => 'Sh',
        'Щ' => 'Shch', 'Ь' => "", 'Ю' => 'Yu', 'Я' => 'Ya'
    );
    return str_replace(' ', '_', strstr(mb_strtoupper($text), $translit));
}

```

```

function addWorker($full_name, $birth_date, $position, $employment_date,
$education, $comment, $photo_path, $department_id = null) {
    global $db;

    $query = "INSERT INTO workers (full_name, birth_date, position,
employment_date, education, comment, photo, department_id) VALUES
(:full_name, :birth_date, :position, :employment_date, :education, :comment, :photo,
:department_id)";

    $stmt = $db->prepare($query);

    return $stmt->execute([
        'full_name' => $full_name,
        'birth_date' => $birth_date,
        'position' => $position,
        'employment_date' => $employment_date,
        'education' => $education,
        'comment' => $comment,
        'photo' => $photo_path,
        'department_id' => $department_id
    ]);
}

function getWorker($id) {
    global $db;

    $query = "SELECT * FROM workers WHERE id = :id";
    $stmt = $db->prepare($query);
    $stmt->execute(['id' => $id]);

    return $stmt->fetch(PDO::FETCH_ASSOC);
}

```

```

function updateWorker($id, $full_name, $birth_date, $position, $employment_date,
$education, $comment, $department_id, $photo_path = null) {
    global $db;
    $query = "UPDATE workers SET full_name = :full_name, birth_date =
:birth_date, position = :position, employment_date = :employment_date, education =
:education, comment = :comment, department_id = :department_id";

    if($photo_path) {
        $query .= ", photo = :photo";
    }

    $query .= " WHERE id = :id";

    $stmt = $db->prepare($query);
    $params = [
        'id' => $id,
        'full_name' => $full_name,
        'birth_date' => $birth_date,
        'position' => $position,
        'employment_date' => $employment_date,
        'education' => $education,
        'comment' => $comment,
        'department_id' => $department_id
    ];

    if($photo_path) {
        $params['photo'] = $photo_path;
    }

    return $stmt->execute($params);
}

```

```
}
```

```
function deleteWorker($id) {
```

```
    global $db;
```

```
    $query = "DELETE FROM workers WHERE id = :id";
```

```
    $stmt = $db->prepare($query);
```

```
    return $stmt->execute(['id' => $id]);
```

```
}
```

```
/* Відпустки */
```

```
// Функція для додавання нової відпустки
```

```
function addVacation($worker_id, $vacation_type, $start_date, $end_date) {
```

```
    global $db;
```

```
    $query = "INSERT INTO vacations (worker_id, vacation_type, start_date, end_date) VALUES (:worker_id, :vacation_type, :start_date, :end_date)";
```

```
    $stmt = $db->prepare($query);
```

```
    $result = $stmt->execute([
```

```
        'worker_id' => $worker_id,
```

```
        'vacation_type' => $vacation_type,
```

```
        'start_date' => $start_date,
```

```
        'end_date' => $end_date
```

```
    ]);
```

```
    return $result;
```

```
}
```

```
// Функція для отримання усіх відпусток робітника
```

```
function getVacationsByWorker($worker_id) {
```

```
    global $db;
```

```
    $query = "SELECT * FROM vacations WHERE worker_id = :worker_id";
```

```

$stmt = $db->prepare($query);
$stmt->execute(['worker_id' => $worker_id]);
return $stmt->fetchAll(PDO::FETCH_ASSOC);
}

function getAllVacations() {
    global $db;
    $query = "SELECT vacations.vacation_id, workers.full_name,
vacations.start_date, vacations.end_date
FROM vacations
JOIN workers ON vacations.worker_id = workers.id";
    $stmt = $db->prepare($query);
    $stmt->execute();
    return $stmt->fetchAll(PDO::FETCH_ASSOC);
}

/* Департаменти */
function displayDepartments($parentId = NULL) {
    global $db;
    $query = "SELECT * FROM departments WHERE parent_department_id " .
($parentId ? "= :parentId" : "IS NULL");
    $stmt = $db->prepare($query);
    if($parentId) {
        $stmt->execute(['parentId' => $parentId]);
    } else {
        $stmt->execute();
    }
    $departments = $stmt->fetchAll(PDO::FETCH_ASSOC);
    if($departments) {
        echo "<ul>";
    }
}

```

```

foreach($departments as $department) {
    echo "<li>";
    echo $department['department_name'];

    // Отримання інформації про керівника
    $leaderInfo = "";
    if($department['head_id']) {
        $leader = getWorker($department['head_id']);
        if($leader) {
            $leaderInfo = " (Керівник: " . $leader['full_name'] . ")";
        } else {
            $leaderInfo = " (Керівника не назначено)";
        }
    } else {
        $leaderInfo = " (Керівника не назначено)";
    }
    echo $leaderInfo;

    echo "                "                <a
href='edit_department.php?id=".$department['department_id']."'>Редагувати</a>";
    echo "                "                <a
href='add_department.php?parent_id=".$department['department_id']."'>Додати
підрозділ</a>";

    // Отримання робітників цього підрозділу
    $workerQuery = "SELECT * FROM workers WHERE department_id =
:departmentId";
    $workerStmt = $db->prepare($workerQuery);
    $workerStmt->execute(['departmentId' => $department['department_id']]);
    $workers = $workerStmt->fetchAll(PDO::FETCH_ASSOC);

```

```

    if ($workers) {
        echo "<ul>";
        foreach ($workers as $worker) {
            echo "<li>" . $worker['full_name'] . " (Посада: " . $worker['position'] .
")</li>";
        }
        echo "</ul>";
    }

    displayDepartments($department['department_id']);
    echo "</li>";
}
echo "</ul>";
}
}
}

```

```

function displayDepartmentsStyle($parentId = NULL) {
    global $db;

    $query = "SELECT * FROM departments WHERE parent_department_id " .
($parentId ? "= :parentId" : "IS NULL");
    $stmt = $db->prepare($query);

    if($parentId) {
        $stmt->execute(['parentId' => $parentId]);
    } else {
        $stmt->execute();
    }
}

```



```

$departments = $stmt->fetchAll(PDO::FETCH_ASSOC);

if($departments) {
    echo '<div class="hv-item-children">';

    foreach($departments as $department) {
        echo '<div class="hv-item-child">';
        echo '<div class="hv-item">';
        echo '<div class="hv-item-parent">';
        echo '<p class="simple-card">' . $department['department_name'] . '</p>';
        echo "</div>"; // Close hv-item-parent

        displayDepartments($department['department_id']);

        echo '</div>'; // Close hv-item
        echo '</div>'; // Close hv-item-child
    }

    echo '</div>'; // Close hv-item-children
}

}

function addDepartment($department_name, $head_id = null, $parent_department_id
= null) {
    global $db;

```

```

$query = "INSERT INTO departments (department_name, head_id,
parent_department_id) VALUES (:department_name, :head_id,
:parent_department_id)";
$stmt = $db->prepare($query);
$stmt->execute([
    ':department_name' => $department_name,
    ':head_id' => $head_id,
    ':parent_department_id' => $parent_department_id
]);
}

```

```

function editDepartment($department_id, $department_name, $head_id = null,
$parent_department_id = null) {
    global $db;

```

```

    $query = "UPDATE departments SET department_name = :department_name,
head_id = :head_id, parent_department_id = :parent_department_id WHERE
department_id = :department_id";
    $stmt = $db->prepare($query);
    $stmt->execute([
        ':department_id' => $department_id,
        ':department_name' => $department_name,
        ':head_id' => $head_id,
        ':parent_department_id' => $parent_department_id
    ]);
}

```

```

function getDepartment($department_id) {

```

```

global $db;

$query = "SELECT * FROM departments WHERE department_id =
:department_id";
$stmt = $db->prepare($query);
$stmt->execute([':department_id' => $department_id]);

return $stmt->fetch(PDO::FETCH_ASSOC);
}

function getAllWorkers() {
    global $db;

    $query = "SELECT * FROM workers";
    $stmt = $db->prepare($query);
    $stmt->execute();

    return $stmt->fetchAll(PDO::FETCH_ASSOC);
}

function getAllDepartments() {
    global $db;

    $query = "SELECT * FROM departments";
    $stmt = $db->prepare($query);
    $stmt->execute();

    return $stmt->fetchAll(PDO::FETCH_ASSOC);
}

function getDepartmentChildren($departments, $parentId = null) {
    $children = [];
    foreach ($departments as $department) {

```

```

        if ($department['parent_department_id'] == $parentId) {
            $department['children'] = getDepartmentChildren($departments,
$department['department_id']);
            $children[] = $department;
        }
    }
    return $children;
}

function createDepartmentHierarchy($departments, $parentId = null) {
    $hierarchical = [];

    foreach ($departments as $key => $department) {
        if ($department['parent_department_id'] == $parentId) {
            unset($departments[$key]);
            $department['children'] = createDepartmentHierarchy($departments,
$department['department_id']);
            $hierarchical[] = $department;
        }
    }

    return $hierarchical;
}

function displayDepartmentHierarchy($departments) {
    foreach ($departments as $department) {
        echo '<div class="hv-item">';

        echo '<div class="hv-item-parent">';
        echo '<p class="simple-card">' . $department['department_name'] . '</p>';
        echo '</div>';
    }
}

```

```
if (!empty($department['children'])) {  
    echo '<div class="hv-item-children">';  
    displayDepartmentHierarchy($department['children']);  
    echo '</div>';  
}  
  
echo '</div>';  
}  
}
```

## Додаток Б

### Код головної сторінки

```
<?php
session_start();
require_once 'php/functions.php';

if (!isLoggedIn()) {
    header('Location: login.php');
    exit;
}

$departments = getAllDepartments();
$hierarchicalDepartments = getDepartmentChildren($departments);

include 'templates/header.php';
?>

<div class="container">
    <h1>Список співробітників</h1>
    <a href="workers_list.php" class="btn btn-primary mb-3">Список
робітників</a>
    <a href="all_vacation.php" class="btn btn-primary mb-3">Список відпусток</a>
    <a href="vacation_calendar.php" class="btn btn-primary mb-3">Календар
відпусток</a>
    <a href="departments.php" class="btn btn-primary mb-3">Підрозділи</a>

    <h1>Ієрархія підрозділів</h1>
<div class="hv-container">
    <div class="hv-wrapper">
```

```
<div class="hv-item">
  <div class="hv-item-parent">
    <p class="simple-card">Генеральный директор</p>
  </div>
  <div class="hv-item-children">
    <?php displayDepartmentHierarchy($hierarchicalDepartments); ?>
  </div>
</div>
</div>
</div>
</div>

<?php include 'templates/footer.php'; ?>
```