

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ
ПРОГНОЗУВАННЯ ПЛИННОСТІ КАДРІВ ПІДПРИЄМСТВА
RESEARCH AND SOFTWARE IMPLEMENTATION OF
METHODS FOR FORECASTING EMPLOYEE TURNOVER IN
AN ENTERPRISE

Виконав(ла): студент(ка) 2 курсу, групи КНТ-214м

Спеціальності 122 Комп'ютерні науки

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Системи штучного інтелекту

ГОРОБЕЦЬ В. І.

(ПРИЗВИЩЕ та ініціали)

Керівник ЛЬОВКІН В. М.

(ПРИЗВИЩЕ та ініціали)

Рецензент ЗЕЛІК О.В.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
(повне найменування закладу вищої освіти)

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти магістр
Спеціальність 122 Комп'ютерні науки
(код і найменування)
Освітня програма (спеціалізація) Системи штучного інтелекту
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ
Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2025 року

ЗАВДАННЯ
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ГОРОБЦЯ Владислава Ігоровича
(ПРИЗВИЩЕ, ім'я, по батькові)

- Тема проєкту (роботи) Дослідження та програмна реалізація методів прогнозування плинності кадрів підприємства
Research and Software Implementation of Methods for Forecasting Employee Turnover in an Enterprise
керівник проєкту (роботи) к.т.н., доцент ЛЬОВКІН Валерій Миколайович,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)
затверджені наказом закладу вищої освіти від “30” вересня 2025 року № 447
- Строк подання студентом проєкту (роботи) 01 грудня 2025 року
- Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання
- Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз проблеми та постановка завдань дослідження.
2. Матеріали та методи. 3. Проєктування програмного забезпечення.
4. Основні рішення щодо реалізації програмного забезпечення.
5. Експлуатація, тестування та експериментальне дослідження програми.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)
Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-5 Основна частина	ЛЬОВКІН В.М., доцент		
Нормоконтроль	БЄЛОВА А.В., асистент		

7. Дата видачі завдання « 30 » вересня 2025 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області	2 тиждень	Розділ 1
3	Дослідження проблеми	3-5 тижні	Розділ 2
4	Проектування та розробка архітектури програми	6 тиждень	Розділ 3
5	Розробка програми	7-8 тижні	Розділ 4
6	Тестування та експериментальне дослідження	9-10 тижні	Розділ 5
7	Оформлення пояснювальної записки та документів до неї. Нормоконтроль та рецензування	11 тиждень	Додатки
8	Захист роботи	12 тиждень	

Студент(ка)

_____ Владислав ГОРОБЕЦЬ
(підпис) (Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

_____ Валерій ЛЬОВКІН
(підпис) (Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
133 с., 38 рис., 7 табл., 3 дод., 45 джерел.

ПЛИННІСТЬ КАДРІВ, ПРОГНОЗУВАННЯ, ЛЮДСЬКІ РЕСУРСИ,
ШТУЧНИЙ ІНТЕЛЕКТ, HR-МЕНЕДЖЕР, ПРАЦІВНИК, DJANGO,
RANDOM FOREST, MYSQL, HTML, CSS, PYTHON, DOCKER.

Об'єкт дослідження – процес обчислення плинності кадрів підприємства.

Предметом дослідження є комп'ютерні алгоритми та програмні засоби прогнозування плинності кадрів підприємства.

Метою кваліфікаційної роботи є зменшення витрат на управління персоналом підприємства шляхом розробки програмних засобів прогнозування плинності кадрів підприємства.

Матеріали, методи та технічні засоби: мови програмування Python, JavaScript, HTML, CSS, вебфреймворк Django, метод Random Forest, MySQL.

Результати. Створено програмний застосунок для прогнозування плинності кадрів, який на основі алгоритмів машинного навчання аналізує великі обсяги даних про працівників, виявляє приховані закономірності та оцінює ймовірність звільнення.

Висновки. Наукова новизна роботи полягає в методі класифікації працівників підприємства за ймовірністю звільнення, що характеризується використанням моделі, побудованої методом Random Forest, виконанням автоматичного відбору ознак для побудови моделі, що дозволяє, сукупно за підприємством, виконувати прогнозування плинності кадрів.

Галузь використання – бізнес і корпоративний менеджмент.

ABSTRACT

Explanatory note to the diploma qualifying work of the master: 133 pages, 38 figures, 7 tables, 3 appendixes, 45 sources.

EMPLOYEE TURNOVER, FORECASTING, HUMAN RESOURCES, ARTIFICIAL INTELLIGENCE, HR MANAGER, EMPLOYEE, DJANGO, RANDOM FOREST, MYSQL, HTML, CSS, PYTHON, DOCKER.

The object of the study is the process of calculating employee turnover in an enterprise.

The subject of the study is computer algorithms and software tools for predicting employee turnover in an enterprise.

The objective of the thesis is to reduce the costs of personnel management in the enterprise through the development of software tools for predicting employee turnover.

Resources, methods, and tools: programming languages Python, JavaScript, HTML, CSS, Django web framework, Random Forest algorithm, MySQL.

Outcomes. A software application for predicting employee turnover has been created that uses machine learning algorithms to analyze large amounts of employee data, identify hidden patterns, and estimate the likelihood of dismissal.

Conclusions. The scientific novelty of the work lies in the method of classifying employees of an enterprise by the probability of dismissal, characterized by the use of a model built using the Random Forest method, which automatically selects features to construct a model that enables enterprise-wide forecasting of staff turnover.

Application area – business and corporate management.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок	8
Вступ	9
1 Аналіз проблеми та постановка завдань дослідження	12
1.1 Огляд вебзастосунків з управління персоналом	12
1.2 Порівняння аналогів розробки вебзастосунку з управління персоналом	16
1.3 Висновки за розділом 1	17
2 Матеріали та методи	19
2.1 Задача прогнозування плинності кадрів на підприємстві для виявлення ризиків відтоку персоналу	19
2.2 Вибір оптимального метода машинного навчання.....	21
2.3 Метод класифікації працівників підприємства за ймовірністю звільнення	23
2.4 Висновки за розділом 2	26
3 Проектування програмного забезпечення	28
3.1 Вибір мови та засобів розробки.....	28
3.2 Моделювання роботи з програмою за функціональними вимогами	31
3.3 Вибір оптимальної системи керування базою даних та обґрунтування її застосування в рамках розробки програмного продукту	36
3.4 Проектування структури бази даних	38
3.5 Висновки за розділом 3	47
4 Основні рішення щодо реалізації програмного забезпечення	48
4.1 Порядок взаємодії користувача з програмою	48
4.2 Структура застосунку.....	48
4.3 Опис реалізованих класів моделей.....	51
4.4 Опис реалізованих класів представлень.....	59
4.5 Опис реалізованих класів ШІ.....	65
4.6 Висновки за розділом 4	65

5	Експлуатація, тестування та експериментальне дослідження програми	67
5.1	Опис програми.....	67
5.2	Експлуатація програми	68
5.3	Експериментальне дослідження	74
5.4	Висновки за розділом 5	78
	Висновки.....	79
	Перелік джерел посилання	82
	Додаток А Технічне завдання.....	87
	Додаток Б Текст програми.....	93
	Додаток В Слайди презентації	123

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

- CBV – Class-Based View;
- CSV – Comma Separated Values;
- ESS – Employee Self-Service;
- HR – Human Resources;
- HRM – Human Resources Management;
- MSS – Manager Self-Service;
- URL – Uniform Resource Locator;
- БД – база даних;
- ЛР – людські ресурси;
- ОСП – особова справа працівника;
- ПЗ – програмний засіб;
- СКБД – система управління базами даних;
- ШІ – штучний інтелект.

ВСТУП

В сучасних умовах проблема плинності кадрів набуває особливої актуальності. Це комплексне явище, яке формується під впливом багатьох внутрішніх і зовнішніх чинників: динамічний розвиток ринку праці, економічна нестабільність, зміна професійних пріоритетів, кар'єрні перспективи та зростаюча доступність цифрових ресурсів. Усі вони стимулюють працівників до частішої зміни місця роботи. Пошук нової зайнятості, який раніше вимагав значних часових і фізичних ресурсів, сьогодні значно спрощено завдяки широкому розповсюдженню онлайн-платформ та сервісів працевлаштування [1].

Після початку повномасштабного вторгнення в Україні ситуація в країні суттєво ускладнилася. Війна спричинила глибокі соціально-економічні зміни, які безпосередньо вплинули на ринок праці та діяльність підприємств. Масова зовнішня та внутрішня міграція, мобілізація працездатного населення, знищення інфраструктури, економічна нестабільність – усе це створює додаткові навантаження для українського бізнесу. Підприємства змушені працювати в умовах постійної невизначеності, кадрового дефіциту та зростаючої плинності співробітників. Даний негативний фактор позначається на стабільності функціонування виробництва: знижується продуктивність праці, втрачаються важливі корпоративні знання та навички, зростають витрати на підбір, навчання та адаптацію нових працівників. У таких умовах дієве управління персоналом та зменшення небажаної плинності стають критично важливими факторами збереження конкурентоспроможності та життєздатності підприємств [2].

В останні роки важливе місце у вирішенні цих проблем займають технології з використанням функцій штучного інтелекту (ШІ), трансформуючи різноманітні процеси та підвищуючи їхню результативність. Їх інтеграція у системи управління кадрами відкриває нові обрії для створення прозорих, об'єктивних та персоналізованих систем мотивації.

Завдяки аналізу великих обсягів даних про продуктивність, поведінку та індивідуальні вподобання працівників, стає можливим формувати адаптовані мотиваційні стратегії, що суттєво підвищують продуктивність персоналу.

Своєчасне включення до процесу мотиваційних важелів – економічних, соціальних, соціально-психологічних, морально-виховних, впливає на успішність та стабільність бізнесу [3]. Оптимальне поєднання цих інструментів створює умови для підвищення залученості персоналу, формування позитивного організаційного клімату та зміцнення корпоративної культури, а також підвищенню продуктивності праці та сталому розвитку підприємства. Автоматизація кадрових процесів із застосуванням передових технологій дозволяє підприємствам значно скоротити час і фінансові витрати на управління персоналом, підвищити залученість співробітників та знизити рівень плинності кадрів. Такі системи мінімізують вплив суб'єктивних факторів, зокрема упереджень керівництва, «кумівства» та корупційних ризиків, що сприяє більш об'єктивному та справедливому відбору й оцінці працівників. Впровадження інноваційних технологій у сфері керування людськими ресурсами (ЛР) є важливим кроком для подолання труднощів, пов'язаних із втратою кадрового складу, і сприяє формуванню стабільного та конкурентоспроможного кадрового потенціалу підприємств. Ефективне управління кадрами приваблює талановитих співробітників, сприяє їх розвитку, що створює умови для довгострокового зростання компанії [4], [5].

У процесі дослідження застосовано комплекс взаємодоповнюючих алгоритмів, який забезпечив цілісний та системний підхід до аналізу теоретичних засад, практичних підходів і сучасних технологічних рішень у сфері дослідження плинності персоналу із використанням ШІ [6].

За результатом аналізу проблеми звільнення працівників, а також огляду сучасних можливостей інформаційних систем управління персоналом, було визначено мету роботи. Вона полягає у зменшенні витрат на управління

персоналом підприємства шляхом розробки програмних засобів прогнозування плинності кадрів підприємства.

1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАВДАНЬ ДОСЛІДЖЕННЯ

1.1 Огляд вебзастосунків з управління персоналом

У сучасному світі широкого поширення набули спеціалізовані програми для управління персоналом, які дозволяють ефективно вести облік працівників, контролювати їхню продуктивність, формувати повну звітність щодо кожного співробітника та автоматизувати рутинні кадрові процеси. Такі системи, як правило, охоплюють функціональність від прийому на підприємство до оцінювання результатів роботи та аналізу причин звільнення [7].

Особливої популярності набули аналітичні human resources (HR) платформи, які інтегрують ШІ для створення персоналізованих стратегій мотивації персоналу, прогнозування ризиків звільнення співробітників, а також оптимізації витрат на утримання штату.

В Україні також спостерігається активне впровадження подібних технологій у сфері управління персоналом. З кожним роком дедалі більше підприємств – як у приватному, так і державному секторі – переходять на використання сучасних цифрових рішень. Особливу увагу приділяють системам, здатним забезпечити гнучкий підхід до управління персоналом, адаптації нових працівників, а також стимулювання їхньої залученості та відданості компанії [8].

Завдяки таким технологіям підприємства отримують змогу не лише зменшити рівень плинності, а й підвищити загальну ефективність кадрової політики, що у свою чергу, позитивно впливає на конкурентоспроможність бізнесу в цілому.

"BambooHR" – це універсальний ПЗ для управління ЛР, розроблений для спрощення та оптимізації роботи з персоналом для сучасних компаній [9]. Сервіс централізує інформацію про співробітників, зарплату, робочі години та пільги в одному місці, забезпечуючи координацію даних, необхідних для безперебійної роботи організацій (рис. 1.1).

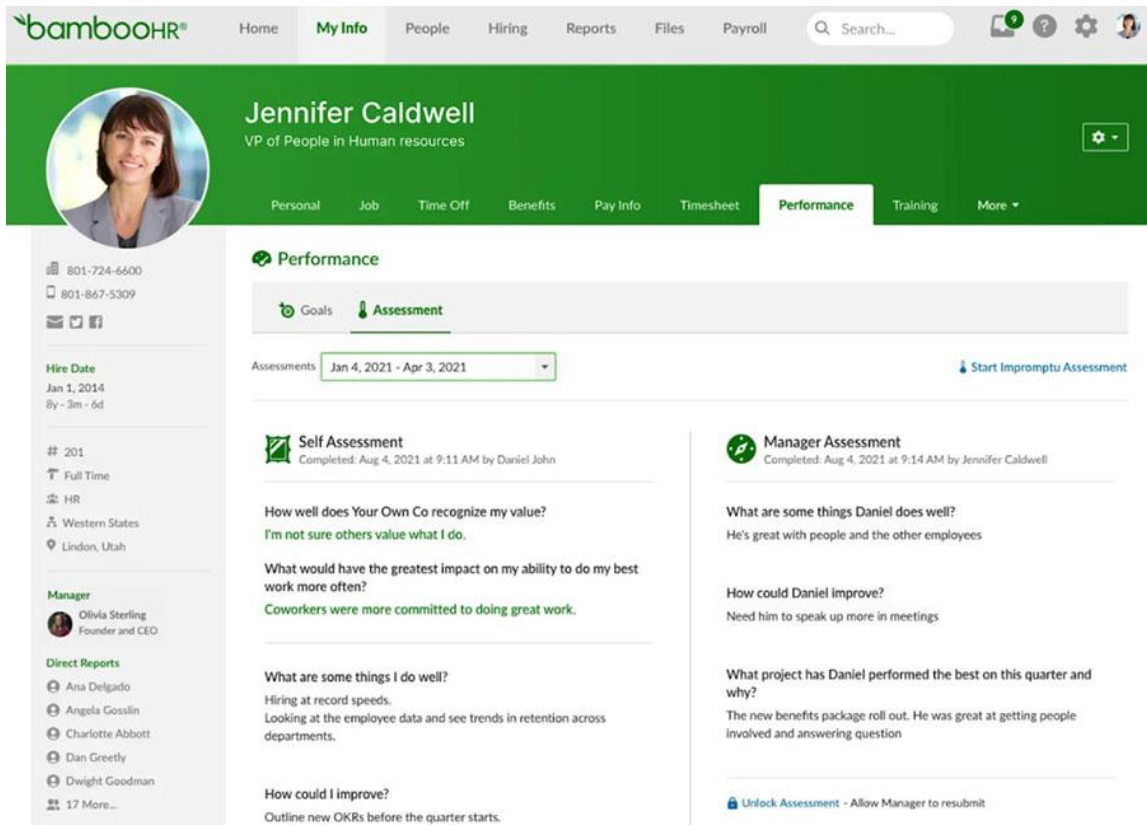


Рисунок 1.1 – BambooHR

На різних HR-платформах широко застосовують концепції Manager Self-Service та Employee Self-Service. Manager Self-Service (MSS) – це тип платформи управління ЛР, яка надає керівникам негайний доступ до інформації про співробітників та спрощує дії, пов'язані з працівниками [10].

Employee Self-Service (ESS) – це система, яка дозволяє працівникам самостійно вирішувати свої питання та отримувати доступ до інформації без допомоги інших співробітників компанії [11].

Вебзастосунок "BambooHR" має таку функціональність:

- наявність MSS;
- наявність ESS;
- необмежене сховище документів;
- автоматизація робочих процесів;
- аналітика та звітність.

До недоліків данного вебресурсу потрібно віднести:

- платна система та додаткова ціна на окремі модулі;

- обмежена кастомізація;
- підходить не для всіх масштабів бізнесу;
- не адаптовано на український ринок;
- відсутність функцій з використанням ШІ.

"Zoho People" – це ПЗ [12], яке призначене для оптимізації всіх аспектів управління ЛР протягом усього виробничого циклу співробітника (рис. 1.2).

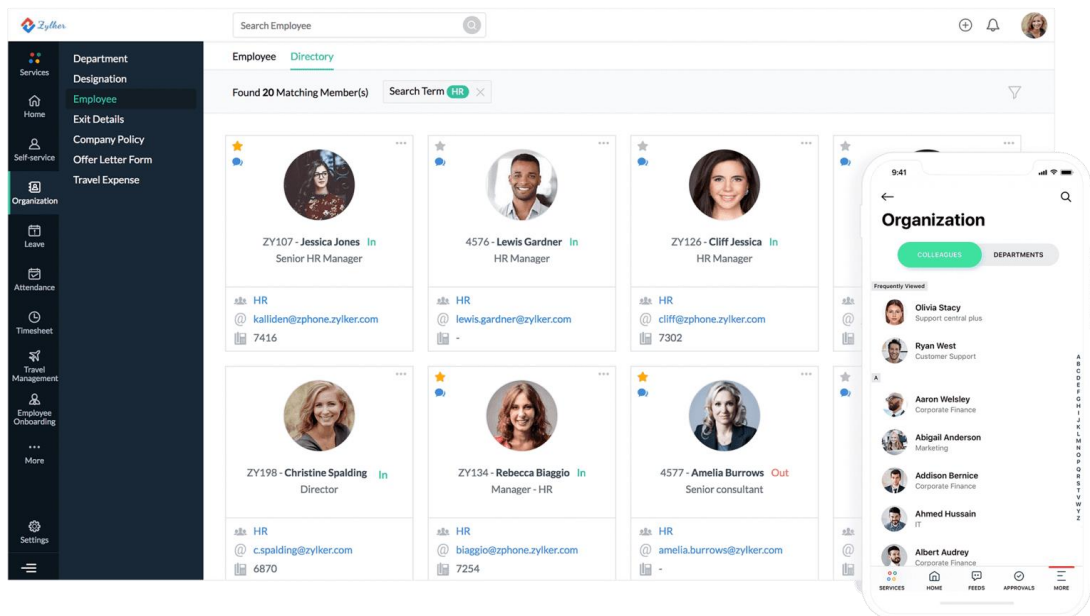


Рисунок 1.2 – Zoho People

До позитивних сторін сайту можна віднести наступне:

- широка функціональність;
- аналітика та звітність;
- модульна структура;
- наявність MSS;
- наявність ESS;
- має орієнтацію на український ринок.

До недоліків даного вебресурсу потрібно віднести:

- платна система;
- складність впровадження;

- складний інтерфейс для новачків;
- обмежені можливості кастомізації;
- відсутність функцій з використанням ШІ.

"КАДРОВИК.UA/КАДРИ.UA" (Україна) – це сучасна українська система, створена для кадровиків [13], HR-спеціалістів, бухгалтерів. (рис. 1.3).

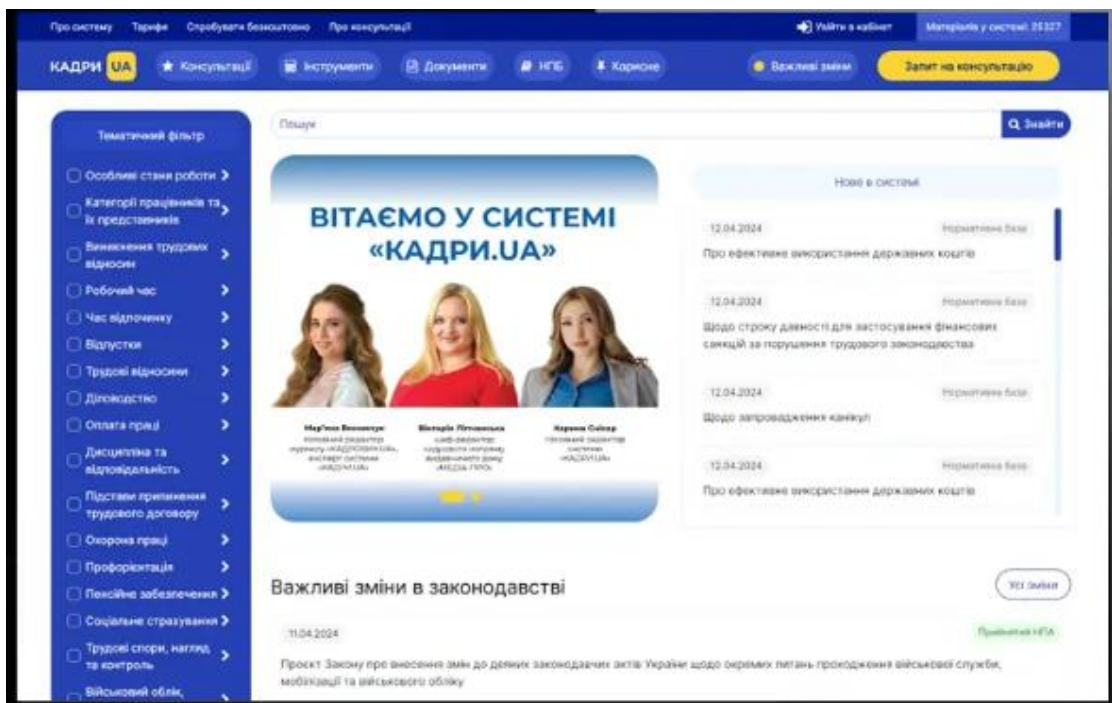


Рисунок 1.3 – КАДРОВИК.UA/КАДРИ.UA

До плюсів можна віднести:

- велика база документів та шаблонів;
- підтримка української мови;
- інтуїтивно зрозумілий інтерфейс;
- можливість масштабування;
- можливість індивідуального пакету.

До мінусів можна віднести:

- платна система;
- обмеження кастомізації;
- затримки з оновленнями;

- відсутній MSS;
- відсутній ESS;
- відсутність функцій з використанням ШІ.

1.2 Порівняння аналогів розробки вебзастосунку з управління персоналом

Однією з ключових особливостей розробки вебзастосунку для управління персоналом є інтеграція з ШІ, що дозволяє здійснювати аналітику та прогнозування плинності кадрів на підприємстві. Здійснюється точна ідентифікація співробітників із підвищеним ризиком звільнення з метою своєчасного застосування ефективних мотиваційних заходів.

На основі результатів було проведено порівняльний аналіз основних ПЗ із програмою, що розробляється в межах даного дослідження, – "HRFactory" (табл. 1.1). Як ключові характеристики для порівняння визначено такі параметри: категорія ресурсу, зручність користувацького інтерфейсу, можливості кастомізації, наявність платних функцій, масштабованість, інтеграція з технологіями ШІ, підтримка режимів MSS та ESS, а також орієнтовність на український ринок [14].

Таблиця 1.1 – Порівняння програмних аналогів з виконуваною розробкою

Критерій	VambooHR	Zoho People	КАДРОВИК.U А / КАДРИ.UA	HRFactory
1	2	3	4	5
Категорія застосунку	HR-система	HR-система	HR-система	HR-система
Зручний інтерфейс	+	–	+	+
Множина базових функцій	+	–	–	+

Кінець таблиці 1.1

1	2	3	4	5
Кастомізація	–	–	–	+
Масштабованість	–	+	–	+
Використання ШІ для прогнозування плинності кадрів	–	–	–	+
Безкоштовний сервіс	–	–	–	+
Наявність MSS	+	+	–	+
Наявність ESS	+	+	–	+
Орієнтовність на український ринок	–	+	+	+

1.3 Висновки за розділом 1

Перший розділ дипломної кваліфікаційної роботи присвячено аналізу предметної області, що стосується проблематики плинності кадрів на підприємстві. У межах дослідження було розглянуто особливості функціонування сучасних інформаційних HR-систем, які можуть бути використані для управління персоналом та прогнозування його відтоку [15]. Зокрема, здійснено порівняльний аналіз трьох популярних платформ: "BambooHR", "Zoho People" та "КАДРОВИК.UA/КАДРИ.UA" з розробляємим застосунком "HRFactory". Для кожної з них було проведено оцінку функціональних можливостей, орієнтованих на підтримку HR-процесів, таких як самообслуговування працівників, управління даними персоналу, автоматизація кадрового документообігу та його аналітика.

Окрему увагу приділено аналізу переваг та недоліків кожного з вебресурсів з урахуванням факторів, що впливають на ефективність управління кадрами та прийняття управлінських рішень у сфері HR. Такий підхід дозволив сформулювати обґрунтовану базу для вибору оптимального

інструментарію в межах подальшої програмної реалізації методів прогнозування плинності кадрів [16].

Аналіз виявив певні функціональні недоліки в частині розглянутих HR-платформ, серед яких варто відзначити:

- безкоштовна версія програмного забезпечення не передбачена, а доступ до окремих функціональних модулів потребує додаткової оплати;
- відсутнє використання ШІ для прогнозування плинності кадрів, що негативно впливає на точність та ефективність кадрового аналізу;
- обмежена здатність до індивідуалізації конфігурації відповідно до потреб організації;
- наявність значної кількості додаткових функціональних модулів призводить до підвищення складності програмних ресурсів;
- низький рівень адаптивності застосунків до зростання кількості користувачів або обсягів даних.

Для досягнення поставленої мети в даній роботі визначено низку наступних завдань:

- виконання аналізу предметної області;
- розробка методу класифікації працівників щодо ймовірності їх звільнення для планування кадрової політики підприємства;
- проектування архітектури ПЗ;
- реалізація ПЗ;
- виконання експериментального дослідження класифікації працівників щодо ймовірності їх звільнення.

2 МАТЕРІАЛИ ТА МЕТОДИ

2.1 Задача прогнозування плинності кадрів на підприємстві для виявлення ризиків відтоку персоналу

У сучасних умовах нестабільного ринку праці питання збереження ключового персоналу набуває стратегічного значення для будь-якої організації. Одним із дієвих підходів до управління трудовими ресурсами є аналіз плинності кадрів на основі історичних даних про працевлаштування та поведінкових характеристик працівників. Основна мета дослідження полягає у розв'язанні задачі бінарної класифікації, спрямованої на визначення ймовірності того, що конкретний працівник залишить організацію протягом найближчого періоду часу. Іншими словами, необхідно виявити залежності між набором вхідних ознак (вік, досвід роботи, стаж, рівень задоволеності та тощо) та фактом звільнення. Водночас важливим аспектом є встановлення часових меж, у рамках яких звільнення вважається проявом плинності кадрів, а не запланованою чи організаційною ротацією [16].

Працівники з високою ймовірністю звільнення зазвичай демонструють певні повторювані патерни: короткий період роботи, низьку продуктивність, знижену залученість, обмежені можливості професійного зростання, часті скарги або звернення до HR-служб. Виявлення таких ознак є основою для побудови ефективної класифікаційної моделі.

У межах цього дослідження кожен запис у вибірці працівників розглядається як окремий приклад задачі машинного навчання, що належить до одного з двох класів: "звільниться" або "не звільниться". Для ефективного розв'язання поставленої задачі необхідно здійснити обґрунтований вибір методу машинного навчання, який максимально відповідатиме визначеним критеріям оцінювання – таким як точність, стійкість до шуму, обчислювальна ефективність та інтерпретованість результатів [17].

Постановку задачі було конкретизовано за допомогою набору даних Employee/HR Dataset (ALL in One) [18], який містить детальну інформацію

про працівників, їх трудову історію, поведінкові патерни та організаційні параметри. Даний датасет повністю відповідає критеріям дослідження та дозволяє здійснити всебічний аналіз факторів, що впливають на ймовірність звільнення.

Вибірка даних складається з екземплярів, кожен з яких містить значення 26 ознак, які описані наступним чином:

- унікальний ідентифікатор працівника (Employee ID);
- ім'я працівника (First Name);
- прізвище працівника (Last Name);
- дата початку роботи в організації (Start Date);
- дата звільнення з організації (Exit Date);
- назва посади (Title);
- ім'я керівника (Supervisor);
- електронна адреса (Email);
- дільниця працівника (Business Unit);
- поточний статус працівника (Employee Status);
- тип зайнятості (Employee Type);
- діапазон заробітної плати (Pay Zone);
- класифікація типу оплати (Employee Classification Type);
- тип звільнення (Termination Type);
- додаткові деталі звільнення (Termination Description);
- цех/відділ працівника (Department Type);
- опис підрозділу працівника (Division Description);
- дата народження працівника (Date of Birth);
- область (State);
- опис функцій працівника (Job Function);
- стать (Gender);
- характер роботи (Location);
- етнічна приналежність (Race);
- сімейний стан (Marital Status);

- оцінка результативності (Performance Score);
- поточна оцінка працівника (Current Employee Rating).

2.2 Вибір оптимального метода машинного навчання

Вибір моделі машинного навчання становить один із визначальних кроків дослідження, оскільки коректний добір алгоритмів безпосередньо впливає на точність, узагальнювальну здатність та ефективність процесу прогнозування. У цьому підрозділі здійснено огляд і порівняльний аналіз різних моделей ШІ, потенційно ефективних для прогнозування плинності кадрів. Обґрунтовано доцільність їх використання з урахуванням специфіки поставленого завдання [19].

Linear regression – це один із базових методів машинного навчання, що застосовується для розв'язання задач регресійного типу, пов'язаних з прогнозуванням кількісних показників на основі одного або декількох незалежних змінних [20]. Для прогнозування цей метод дає змогу побудувати математичну модель, яка демонструє залежність між різними характеристиками та ймовірністю зміни цільової змінної під впливом цих характеристик.

Decision Tree – поширений алгоритм машинного навчання, який застосовується для розв'язання задач як класифікації, так і регресії [21]. Метод дерев рішень використовується для побудови моделі, що здійснює послідовне розділення даних, формуючи ієрархічну структуру прийняття рішень. Кожен внутрішній вузол дерева відповідає певному критерію або умові поділу, а листові вузли відображають кінцеві прогнози. Метою побудови дерева є пошук найкращих точок поділу для характеристик, які мінімізують різницю між передбачуваними та реальними значеннями [22].

Random Forest – це ансамблевий метод машинного навчання, який ґрунтується на побудові великої кількості дерев прийняття рішень для підвищення точності прогнозів та зменшення ризику перенавчання [23].

Кожне дерево в лісі навчається на випадково вибраних підмножинах даних та робить власне передбачення, а результат формується як середнє значення або як результат голосування більшості. Окремі дерева можуть давати неточні результати через перенавчання, проте при поєднанні всіх дерев випадкові помилки взаємно компенсуються, роблячи загальний прогноз більш точним.

Нижче наведено порівняння методів машинного навчання (табл. 2.1).

Таблиця 2.1 – Порівняння методів машинного навчання

Критерій	Linear Regression	Decision Tree	Random Forest
Тип задачі	Регресія	Класифікація, регресія	Класифікація, регресія
Тип моделі	Лінійна регресія	Одиночне дерево рішень	Ансамблевий метод (комбінація дерев рішень)
Точність передбачень	–	–	+
Стійкість до перенавчання	–	–	+
Підтримка нелінійних зв'язків	–	+	+
Обробка шумів/аномалій	–	–	+
Робота з великою кількістю змінних	–	–	+
Складність налаштування	+	+	–

На основі порівняльного аналізу обрано метод Random Forest, який вирізняється високою точністю прогнозування, стійкістю до перенавчання та

здатністю ефективно обробляти шумові дані. Результативний у роботі з великими обсягами даних та здатний оперувати як числовими, так і категоріальними ознаками, потребує мінімального попереднього опрацювання даних.

2.3 Метод класифікації працівників підприємства за ймовірністю звільнення

Для прогнозування процесу плинності кадрів на підприємстві необхідно пройти кілька етапів, розв'язуючи за кожним працівником фактично задачу класифікації працівників підприємства за ймовірністю звільнення, таким чином приводячи до сукупного розв'язання задачі прогнозування плинності кадрів.

На першому етапі необхідно отримати структуровану інформацію про працівників підприємства, яка включає їхні індивідуальні та професійні характеристики. Ці дані мають бути представлені у табличному форматі для подальшого аналізу. Важливо забезпечити репрезентативність вибірки, охопивши різні підрозділи підприємства, посади, вікові категорії та рівні кваліфікації. Це дозволить створити повну картину кадрового складу та підвищити точність аналітичних висновків. Структура вибірки має бути тією, що відповідає описанам у підрозділі 2.1 принципам.

На другому етапі отримані дані аналізуються на наявність помилок, пропусків або аномалій. На цьому етапі видаляються або коригуються неточності, узгоджуються формати та забезпечується однорідність даних для подальшої роботи. Необхідно застосувати автоматизовані методи очищення (алгоритми виявлення викидів), експертну перевірку, щоб мінімізувати втрату важливої інформації. Це забезпечує якість вхідних даних, яка є критичною для побудови достовірної моделі прогнозування.

На третьому етапі дані переводяться в форму, придатну для аналізу та прогнозування. Це включає нормалізацію числових показників, кодування

категоріальних ознак, формування нових узагальнених показників, що краще відображають закономірності плинності кадрів. Така попередня обробка дозволяє підвищити стабільність і надійність моделі.

На четвертому етапі визначаються ключові характеристики, які найбільше впливають на ймовірність звільнення працівника. Відбираються ті характеристики, які несуть найціннішу інформацію для прогнозу, зменшуючи шум та надлишковість даних. Для визначення інформативності кожної змінної застосовується критерій χ^2 (χ^2), який дозволяє оцінити ступінь залежності між ознаками та цільовою змінною. Характеристики, що демонструють вищі значення статистики $\chi^2 > 0.5$, виступають найбільш інформативними для побудови моделі, оскільки їхній зв'язок з цільовою змінною є значущим. Натомість змінні з низьким значенням $\chi^2 < 0.2$ можуть бути видалені, вони не забезпечують суттєвого внеску в прогнозування. Такий відбір підвищує інтерпретованість моделі та знижує ризик перенавчання.

На п'ятому етапі дані поділяються у відокремлені набори для навчання і тестування прогнозу на ефективність. Це дозволяє оцінити здатність методу робити точні передбачення на нових, раніше не використаних даних. Застосовується класичний розподіл вибірки 80 % на 20 %.

На шостому етапі формується модель прогнозування на основі методу Random Forest, яка описує залежність ймовірності плинності кадрів від ключових факторів. Модель адаптується до навчального масиву даних з метою оптимізації її параметрів. Застосування Random Forest дозволяє ефективно враховувати нелінійні залежності та взаємодії між різними змінними, що особливо важливо в кадровій аналітиці. Оптимізація гіперпараметрів моделі, таких як кількість дерев і глибина, здійснюється для досягнення максимальної точності прогнозу.

На сьомому етапі виконується оцінка здатності моделі до класифікації працівників за ймовірністю звільнення, а також аналізуються ділянки, де

точність передбачень є високою або потребує додаткового уточнення, щоб знайти закономірності, які неочевидні на перший погляд. Матриця помилок дає змогу оцінити точність класифікації, виявити типові помилки моделі та визначити, у яких групах працівників прогноз працює найменш ефективно. Узагальнений звіт повинен включати основні метрики оцінювання – точність, повнота, F1-міру, що дозволяє зробити висновки щодо здатності моделі до генералізації.

Нижче наведено метод класифікації працівників підприємства за ймовірністю звільнення (рис. 2.1).



Рисунок 2.1 – Метод класифікації працівників підприємства за ймовірністю звільнення

2.4 Висновки за розділом 2

У другому розділі було розглянуто підхід до вирішення задачі класифікації плинності кадрів на підприємстві з метою виявлення працівників, схильних до звільнення та подальшого впровадження превентивних заходів у рамках кадрової політики підприємства. Сучасні виклики на ринку праці зумовлюють необхідність глибокого аналізу поведінкових та професійних характеристик персоналу, що дозволяє формувати прогностичні моделі для підвищення стабільності кадрового складу [24].

У дослідженні використано набір даних Employee/HR Dataset (ALL in One), який містить повну інформацію про робітників підприємства та відповідає вимогам до якості й повноти даних для виконання прогнозування.

Було здійснено аналіз та порівняння основних моделей машинного навчання, що можуть бути застосовані для вирішення основної задачі. Розглянуто такі методи, як Linear Regression, Decision Tree та Random Forest, кожен із яких має власні переваги та недоліки. На основі порівняльного аналізу визначено, що Random Forest є найбільш доцільним методом для розв'язання задачі, сформульованої у роботі. Його перевагами є висока точність, стійкість до перенавчання, здатність ефективно працювати з великими та різнорідними наборами даних, а також мінімальні вимоги до попереднього опрацювання характеристик [25].

Докладно розкрито побудова методу класифікації працівників підприємства за ймовірністю звільнення, який передбачає виконання послідовних етапів обробки даних: збір даних, очищення та корекція даних, трансформація даних, вибір ознак, розподіл даних, побудова моделі прогнозування та навчання, оцінка моделі.

Наукова новизна роботи полягає в методі класифікації працівників підприємства за ймовірністю звільнення, що характеризується використанням моделі, побудованої методом Random Forest, виконанням

автоматичного відбору ознак для побудови моделі, що дозволяє сукупно за підприємством виконувати прогнозування плинності кадрів.

Завдяки поєднанню зазначених компонентів у єдиній аналітичній моделі створюється можливість виявлення нелінійних взаємозв'язків між факторами, що впливають на ймовірність звільнення та підвищується стійкість і точність загального прогнозування за всім кадровим складом підприємства. Такий підхід не лише забезпечує глибше розуміння причин звільнення працівників, але й сприяє розробці запобіжних стратегій управління персоналом, орієнтованих на збереження ключових фахівців і підвищення рівня кадрової стабільності на підприємстві [26].

3 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Вибір мови та засобів розробки

При розробці програмного засобу існує ряд особливостей та технічних аспектів, які потрібно врахувати на ранніх стадіях роботи над проектом. Професійна backend розробка забезпечує функціональність, продуктивність, безпеку та злагоджену роботу всіх складових продукту, зокрема серверної частини, бази даних, адмін-панелі, систем авторизації. Це важливо для стабільного функціонування програми та ефективної взаємодії з клієнтською частиною [27].

Вибір мови програмування є головним аспектом розробки вебзастосунку. Він залежить від цілей проєкту, цільової платформи, вимог до продуктивності, наявності ресурсів, а також від ринкової затребуваності. Із врахуванням специфіки веброзробки було проведено аналіз кількох альтернативних мов програмування, зокрема, Python, Java, C#.

Java – потужна та стабільна мова, що давно зарекомендувала себе у створенні великих корпоративних систем. Проте її громіздкий синтаксис і складність розуміння часто ускладнюють розробку, особливо на початкових етапах [28].

C# – сучасна мова з ефективною архітектурою та широкими можливостями. Однак її тісна прив'язаність до екосистеми Microsoft і складні інструменти розгортання можуть обмежувати гнучкість і швидкість реалізації проєктів [29].

Python суттєво вирізняється на фоні обох. Завдяки простому, інтуїтивному синтаксису, мінімальній конфігурації та великій екосистемі бібліотек для веброзробки, він дозволяє значно швидше переходити від ідеї до готового рішення. Дана мова сприяє продуктивній, гнучкій при цьому структурованій та методично обґрунтованій розробці [30].

Узагальнені результати порівняння представлено в таблиці (табл. 3.1).

Таблиця 3.1 – Порівняння мов програмування для створення ПЗ

Критерій	Python	Java	C#
Простота синтаксису та зрозумілість коду	+	–	–
Швидкість розробки backend-застосунків	+	–	–
Екосистема бібліотек для AI/ML	+	–	–
Продуктивність у високонавантажених системах	–	+	+
Розмір спільноти та підтримка	+	+	–
Масштабованість для великих проєктів	+	+	+
Кросплатформність	+	+	+
Збирач сміття	+	+	+
Підтримка ООП	+	+	+
Підтримка функціонального програмування	+	–	–
Інструменти налагодження	–	+	+
Різноманітність фреймворків для створення вебзастосунків	+	–	–

Одна з ключових переваг мови Python – наявність великої екосистеми бібліотек і фреймворків, що надають розробникам готові інструменти та рішення для прискорення процесу розробки. Популярні фреймворки для веброзробки – Django, Flask, FastAPI які допомагають розробляти високоякісні вебдодатки [31], [32], [33].

Нижче наведена таблиця (табл. 3.2), що порівнює основні характеристики та особливості фреймворків Django, Flask та FastAPI.

Таблиця 3.2 – Порівняння фреймворків Django, Flask та FastAPI

Критерій	Django	Flask	FastAPI
Мова програмування	Python	Python	Python
Повнота функціональності	+	–	–
Швидкість розробки застосунків	+	–	+
Документація та спільнота	+	+	+
Наявність адмін-панелі	+	–	–
Наявність аутентифікації та авторизації	+	–	–
Захист від CSRF атак	+	–	–
Управління статичними файлами	+	–	–
Можливість підключення до різних СКБД	+	+	+

Для даного проєкту вибір зроблено на користь вебфреймворку Django. Це високорівневий фреймворк мови програмування Python, призначений для швидкої розробки масштабованих вебзастосунків та є ефективним рішенням для створення як малих, так і великих корпоративних систем. Він надає готову архітектуру та інструменти для роботи з базами даних, автентифікації користувачів, управління сесіями та маршрутизації, велика увага приділяється безпеці [34].

3.2 Моделювання роботи з програмою за функціональними вимогами

Повний перелік функціональних вимог [35] до програми включає:

- авторизація;
- відправка заяви до HR-відділу на отримання облікових даних;
- визначення типу користувача;
- реєстрація нового працівника;
- створення особової справи працівника (ОСП);
- створення опитувань, щодо якості праці;
- перенесення акаунту звільненого працівника до архіву;
- перегляд ОСП;
- редагування ОСП;
- перегляд заяв працівника;
- перегляд таблицю робочого часу;
- перевірка залишку днів відпустки;
- перегляд доган;
- введення дисциплінарних санкцій;
- створення запрошення на внутрішні навчання;
- створення зустрічей із працівником;
- створення звіту по плинності кадрів в підрозділі/підрозділах;
- налаштування моделі ШІ;
- отримання результату прогнозування;
- інтеграція результатів передбачення;
- відправка службових записок;
- перегляд архіву класифікації працівників підприємства за ймовірністю звільнення;
- перегляд архіву працівників;
- звіт по рівню заробітної плати;
- вихід з облікового запису;

- перегляд запрошень на внутрішні навчання;
- перегляд графіку тренінгів;
- перегляд всіх створених заяв;
- підтвердження участі у тренінгу;
- написання заяв;
- скасування заяв;
- перегляд всіх опитувань;
- участь в опитуваннях;
- перегляд списку зустрічей з HR;
- створення повідомлення про проблему;
- перевірка статусу вирішення проблеми;
- подача заявки на редагування ОСП;
- подання запиту на зміну графіку;
- перегляд історії відпусток;
- отримання повідомлення про важливі події;
- доступ до адмін-панелі;
- тестування моделі ШІ;
- навчання моделі ШІ;
- перевірка логів застосунку;
- розділення наборів даних для навчання та тестування моделі ШІ;
- моніторинг некоректних відповідей моделі ШІ;
- трансформація та попередня обробка даних;
- створення резервних копій моделі ШІ;
- відтворення резервних копій моделі ШІ;
- завантаження вибірки;
- перегляд політики підприємства.

На основі цих вимог було визначено перелік сценаріїв роботи для HR-менеджера (рис. 3.1) [36].

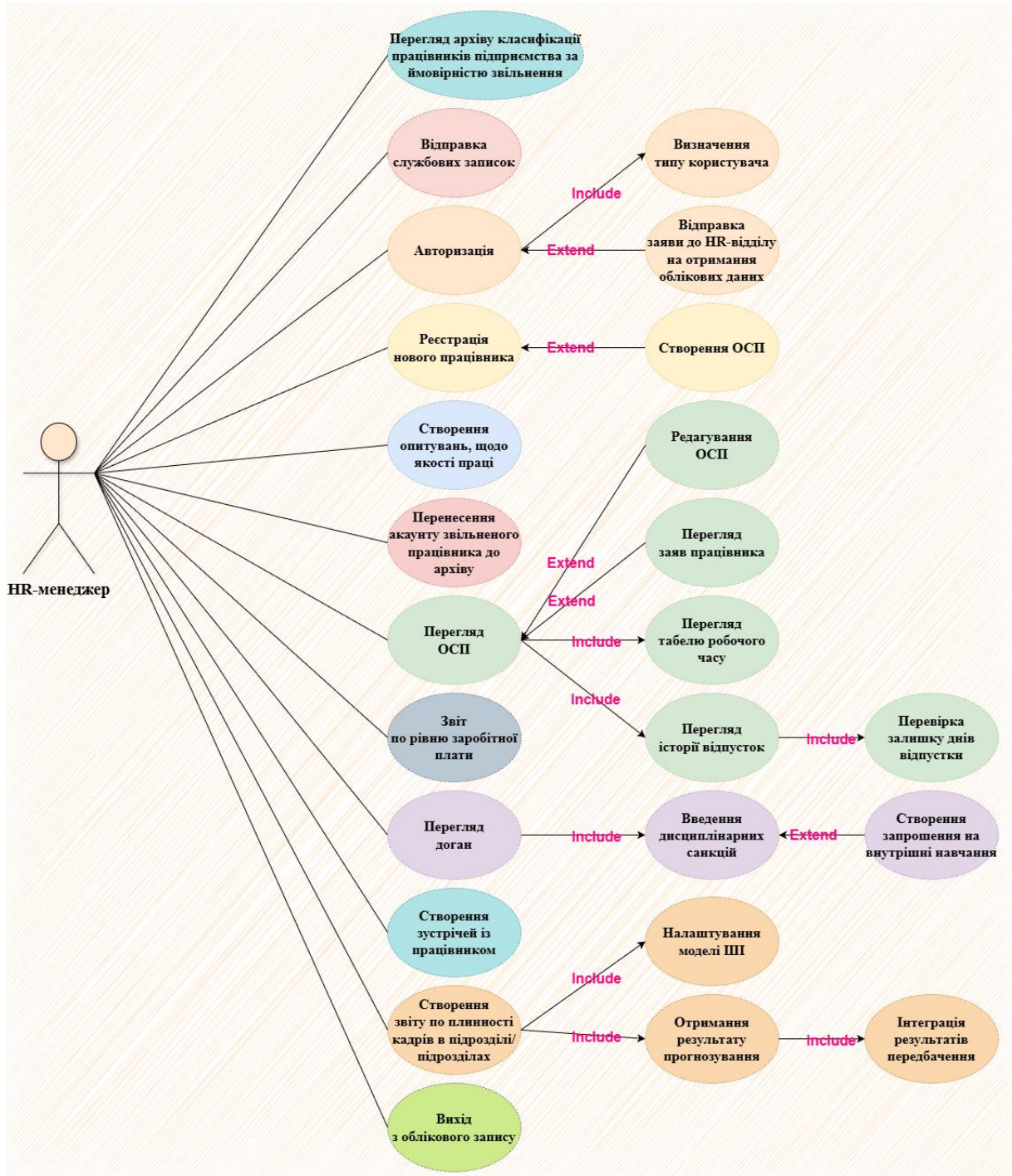


Рисунок 3.1 – Діаграма прецедентів HR-менеджера

Функціональні вимоги, викладені вище, стали основою для формування переліку робочих сценаріїв працівника (рис. 3.2).

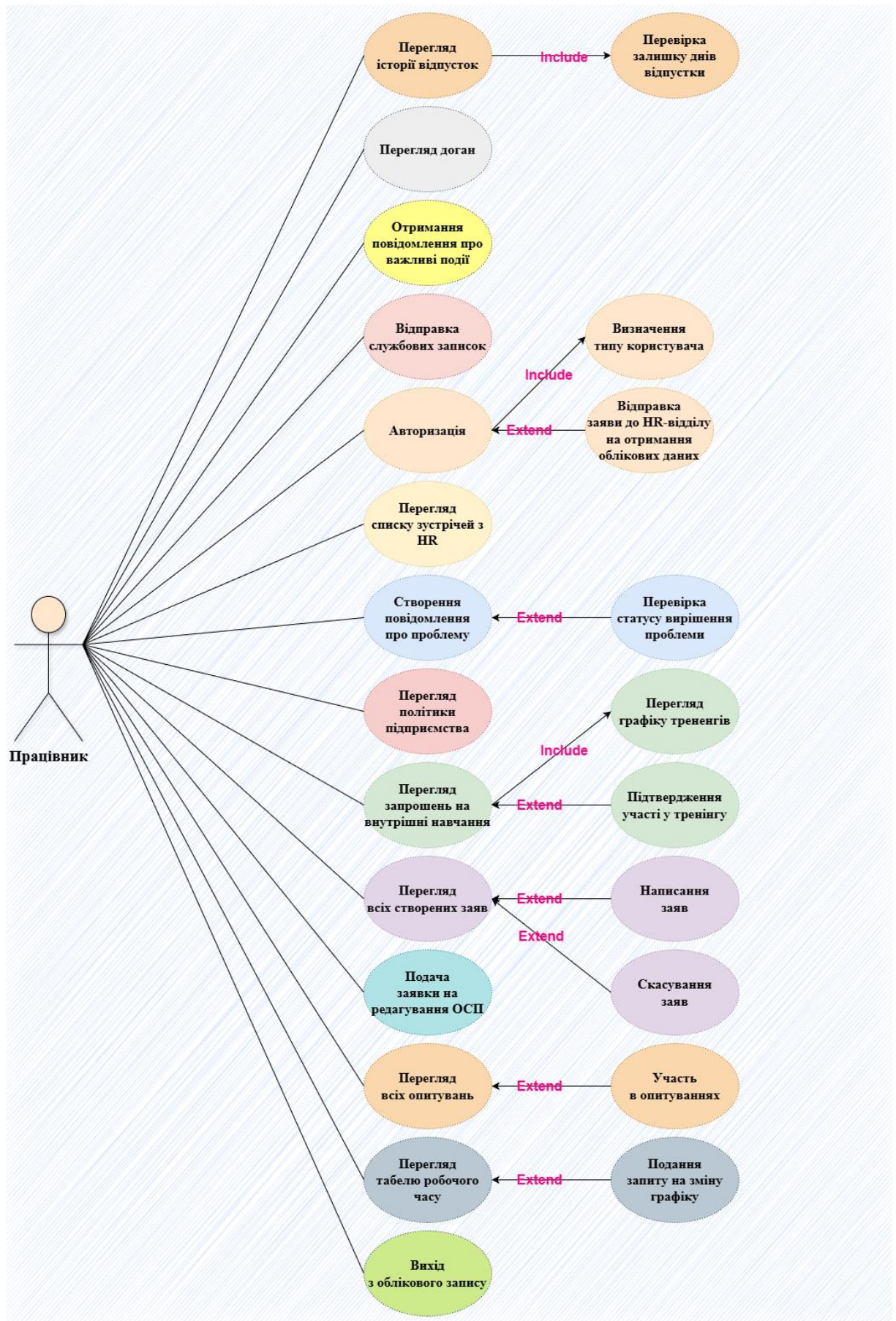


Рисунок 3.2 – Діаграма прецедентів працівника

Було сформовано перелік робочих сценаріїв адміністратора (рис. 3.3).

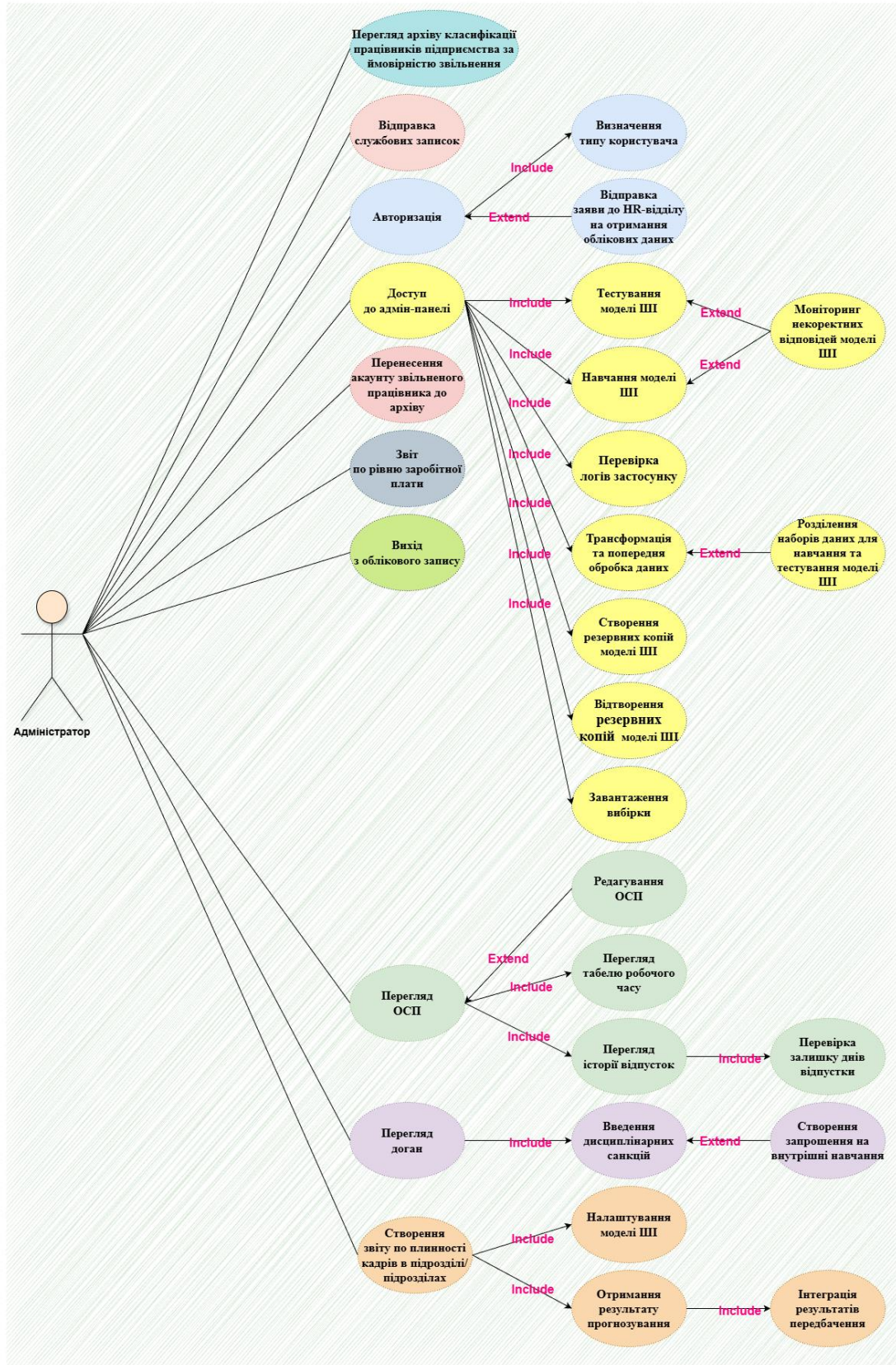


Рисунок 3.3 – Діаграма прецедентів адміністратора

3.3 Вибір оптимальної системи керування базою даних та обґрунтування її застосування в рамках розробки програмного продукту

MySQL – це система керування базами даних (СКБД) із відкритим вихідним кодом, яка широко використовується для створення реляційних баз даних [37]. Вона відзначається високою продуктивністю, надійністю та сумісністю з різними платформами та мовами програмування. MySQL є однією з найпоширеніших у веброзробці, зокрема в стеку Python, Django.

MySQL активно застосовується у багатьох сферах, таких як електронна комерція, банківська справа, аналітика, освітні та урядові системи.

Причини вибору MySQL для дослідження та програмної реалізації теми для даної дипломної кваліфікаційної роботи магістра:

- продуктивність: MySQL оптимізована для високошвидкісної обробки запитів та масштабованості при великій кількості з'єднань, що робить її придатною для динамічних вебдодатків;
- сумісність: підтримує стандарт SQL, що забезпечує простоту інтеграції з різними фреймворками та інструментами розробки;
- безпека: MySQL включає механізм автентифікації, контроль доступу, шифрування з'єднань, що забезпечує належний рівень захисту даних;
- регулярні оновлення, що забезпечують системі актуальність стабільність, відповідність сучасним вимогам до програмного забезпечення;
- легка міграція – можливість переходу на інші СУБД за допомогою спеціальних інструментів.

Окремо слід відзначити безкоштовність та кросплатформеність MySQL, що дозволяє використовувати її як у середовищі Windows, так і на Linux-системах, не залежно від комерційних ліцензій.

Нижче наведена таблиця (табл. 3.3), яка порівнює основні характеристики трьох популярних реляційних БД: MySQL, NoSQL [38] та Microsoft SQL Server [39].

Таблиця 3.3 – Порівняльна таблиця СКБД

Критерій	NoSQL	MySQL	Microsoft SQL Server
1	2	3	4
Тип БД	Документна	Реляційна	Реляційна
Безкоштовне використання	–	+	–
Підтримка SQL-запитів	–	+	+
Інструменти адміністрування	–	+	–
Кросплатформеність	+	+	–
Підтримка спільноти	+	+	–
Відкритий вихідний код	–	+	–
Підтримка транзакцій	–	+	+
Підтримка індексів	+	+	+
Простота встановлення	+	+	–
Безпека	–	+	+
Підтримка JOIN-запитів	–	+	+

Кінець таблиці 3.3

1	2	3	4
Низькі вимоги для ресурсів комп'ютера	+	+	–
Поширеність у веб-розробці	–	+	–

3.4 Проектування структури бази даних

Для реалізації цього проєкту використовуватиметься СКБД MySQL, а також відповідні таблиці, створені для зберігання та обробки необхідної інформації.

Таблиця `authentication_employeeprofile` зберігає персональні дані про працівників підприємства, включає поля:

- `id` (Integer) – унікальний ідентифікатор працівника;
- `first_name` (Varchar) – ім'я працівника;
- `last_name` (Varchar) – прізвище працівника;
- `start_date` (Date) – дата початку роботи в організації;
- `exit_date` (Date) – дата звільнення з організації;
- `title` (Varchar) – назва посади працівника;
- `supervisor` (Varchar) – ім'я керівника;
- `email` (Varchar) – електронна адреса працівника;
- `business_unit` (Varchar) – дільниця працівника;
- `employee_status` (Varchar) – поточний статус працівника;
- `employee_type` (Varchar) – тип зайнятості;
- `pay_zone` (Varchar) – діапазон заробітної плати;
- `employee_classification_type` (Varchar) – класифікація типу оплати;
- `termination_type` (Varchar) – тип звільнення;
- `termination_description` (Text) – додаткові деталі звільнення;

- department_type (Varchar) – цех/відділ працівника;
- division_description (Varchar) – опис підрозділу працівника;
- dob (Date) – дата народження працівника;
- state (Varchar) – регіон;
- job_function (Varchar) – опис функцій працівника;
- gender (Varchar) – стать працівника;
- location (Varchar) – характер роботи;
- race_ethnicity (Varchar) – етнічна належність;
- maritar_status (Varchar) – сімейний стан;
- performance_score (Varchar) – оцінка результативності працівника;
- current_employee_rating (Varchar) – поточна оцінка працівника;
- user_id (Integer) – посилання на таблицю auth_user.

Таблиця authentication_training зберігає інформацію про тренінги, що проводяться на підприємстві, включає поля:

- id (Integer) – унікальний ідентифікатор запису;
- title (Varchar) – назва тренінгу;
- description (Text) – опис тренінгу;
- start_datetime (DateTime) – дата та час початку тренінгу;
- end_datetime (DateTime) – дата та час завершення тренінгу;
- location (Varchar) – місце проведення;
- capacity (Integer) – максимальна кількість учасників;
- is_active (Bool) – позначає активність тренінгу;
- created_at (DateTime) – дата створення запису;
- updated_at (DateTime) – дата останнього оновлення.

Таблиця authentication_trainingregistration зберігає інформацію про реєстрацію працівників підприємства на тренінги, включає поля:

- id (Integer) – унікальний ідентифікатор запису;
- user_id (Integer) – посилання на таблицю auth_user;

- training_id (Bigint) – посилання на таблицю authentication_training;

- registered_at (DateTime) – дата реєстрації;
- confirmed (Bool) – позначає, чи підтверджено участь.

Таблиця document_manager_document зберігає інформацію про скан-копії документів, які завантажують працівники підприємства, включає поля:

- id (Integer) – унікальний ідентифікатор запису;
- user_id (Integer) – посилання на таблицю auth_user;
- title (Varchar) – назва документа;
- description (Text) – опис документа;
- scan_file (Varchar) – файл документа;
- uploaded_at (DateTime) – дата та час завантаження файлу

документа.

Таблиця document_manager_application зберігає інформацію про заяви, які подані працівниками підприємства, включає поля:

- id (Integer) – унікальний ідентифікатор запису;
- user_id (Integer) – посилання на таблицю auth_user;
- subject (Varchar) – тема заяви;
- body (Text) – текст заяви;
- category (Varchar) – категорія заяви;
- status (Varchar) – статус заяви;
- created_at (DateTime) – дата створення;
- updated_at (DateTime) – дата оновлення.

Таблиця document_manager_applicationattachment зберігає інформацію про скан-копії документів, що прикріплюють працівники підприємства до заяв, включає поля:

- id (Integer) – унікальний ідентифікатор запису;
- application_id (Integer) – посилання на таблицю document_manager_application;
- file (Varchar) – завантажений файл;

- `uploaded_at (DateTime)` – дата завантаження.

Таблиця `main_poll` зберігає інформацію про опитування працівників підприємства, включає поля:

- `id (Integer)` – унікальний ідентифікатор запису;
- `title (Varchar)` – назва опитування;
- `description (Text)` – опис опитування;
- `created_at (DateTime)` – дата створення;
- `is_active (Boolean)` – позначає, чи активне опитування.

Таблиця `main_question` зберігає інформацію про питання конкретного опитування, включає поля:

- `id (Integer)` – унікальний ідентифікатор запису;
- `poll_id (Integer)` – посилання на таблицю `main_poll`;
- `text (Varchar)` – текст питання;
- `question_type (Varchar)` – тип питання.

Таблиця `main_choice` зберігає інформацію про варіанти відповідей на питання опитувань, включає поля:

- `id (Integer)` – унікальний ідентифікатор запису;
- `question_id (Integer)` – посилання на таблицю `main_question`;
- `text (Varchar)` – текст варіанту відповіді.

Таблиця `main_answer` зберігає інформацію про відповіді на питання опитувань працівників підприємства, включає поля:

- `id (Integer)` – унікальний ідентифікатор запису;
- `user_id (Integer)` – посилання на таблицю `auth_user`;
- `poll_id (Integer)` – посилання на таблицю `main_poll`;
- `submitted_at (DateTime)` – дата подання відповіді;

Таблиця `main_choiceanswer` зберігає інформацію про вибіркові відповіді працівників підприємства на питання опитувань, включає поля:

- `id (Integer)` – унікальний ідентифікатор запису;
- `answer_id (Integer)` – посилання на таблицю `main_answer`;
- `question_id (Integer)` – посилання на таблицю `main_question`;

Таблиця `main_textanswer` зберігає інформацію про текстові відповіді працівників підприємства на питання опитувань, включає поля:

- `id (Integer)` – унікальний ідентифікатор запису;
- `answer_id (Integer)` – посилання на таблицю `main_answer`;
- `question_id (Integer)` – посилання на таблицю `main_question`;
- `text (Text)` – текст відповіді.

Таблиця `main_issue` зберігає інформацію про проблеми працівників підприємства, включає поля:

- `id (Integer)` – унікальний ідентифікатор запису;
- `user_id (Integer)` – посилання на таблицю `auth_user`;
- `title (Varchar)` – заголовок проблеми;
- `description (Text)` – опис проблеми;
- `category (Varchar)` – категорія запису;
- `status (Varchar)` – статус заяви;
- `created_at (DateTime)` – дата створення;
- `updated_at (DateTime)` – дата оновлення.

Таблиця `main_disciplinaryaction` зберігає інформацію про дисциплінарні санкції щодо працівника підприємства, включає поля:

- `id (Integer)` – унікальний ідентифікатор запису;
- `employee_id (Integer)` – посилання на таблицю `auth_user`;
- `reason (Text)` – причина догани;
- `issued_by_id (Integer)` – посилання на таблицю `auth_user`;
- `date_issued (DateField)` – дата видачі;
- `expiration_date (Date)` – дата закінчення дії;
- `is_active (Bool)` – чи активна догана.

Таблиця `main_corporatepolicy` зберігає інформацію про корпоративні політики підприємства, включає поля:

- `id (Integer)` – унікальний ідентифікатор запису;
- `title (Varchar)` – назва політики;
- `category (Varchar)` – категорія;

- content (Text) – зміст політики;
- document (Varchar) – завантажений файл політики;
- created_by_id (Integer) – автор політики;
- created_at (DateTime) – дата створення;
- updated_at (DateTime) – дата оновлення;
- is_active (Bool) – актуальність політики.

Таблиця main_hrmeeting зберігає інформацію про зустрічі HR-менеджерів з працівниками, включає поля:

- id (Integer) – унікальний ідентифікатор запису;
- disciplinary_action_id (Integer) – посилання на таблицю main_disciplinaryaction;
- date (DateTime) – дата та час зустрічі;
- location (Varchar) – місце проведення;
- notes (Text) – примітки керівника;
- attended (Boolean) – позначка, чи з'явився працівник.

Таблиця ml_predictor_employeeprediction зберігає інформацію про класифікацію ймовірності збільшення працівників підприємства, включає поля:

- id (Integer) – унікальний ідентифікатор запису;
- user_id (Integer) – посилання на таблицю auth_user;
- prediction_date (DateTime) – дата та час створення передбачення;
- will_leave (Bool) – прогноз звільнення працівника;
- probability (Real) – ймовірність звільнення працівника;
- model_version (Varchar) – версія моделі передбачення.

Таблиця auth_user зберігає інформацію про облікові дані, для входу до застосунку, працівників підприємства, включає поля:

- id (Integer) – унікальний ідентифікатор запису;
- password (Varchar) – хешований пароль користувача;

- last_login (DateTime) – дата та час останнього входу користувача в систему;
- is_superuser (Bool) – ознака, що користувач має права адміністратора;
- username (Varchar) – логін користувача;
- last_name (Varchar) – прізвище користувача;
- last_login (DateTime) – дата та час останнього входу користувача до системи;
- email (Varchar) – електронна адреса користувача;
- is_staff (Bool) – ознака доступу до адмін-панелі;
- is_active (Bool) – ознака активності облікового запису;
- date_joined (DateTime) – дата та час реєстрації користувача;
- first_name (Varchar) – ім'я користувача.

Таблиця auth_group зберігає інформацію про ролі працівників підприємства у застосунку, включає поля:

- id (Integer) – унікальний ідентифікатор запису;
- name (Varchar) – назва ролі.

Таблиця django_session зберігає інформацію про сесії працівників підприємства у застосунку, включає поля:

- session_key (Varchar) – унікальний ключ сесії;
- session_data (Text) – додаткові дані сесії;
- expire_date (DateTime) – дата та час, коли сесія стає недійсною.

Таблиця auth_permission зберігає інформацію про дозволи у застосунку для ролей працівників підприємства, включає поля:

- id (Integer) – унікальний ідентифікатор запису;
- name (Varchar) – назва дозволу;
- content_type_id (Integer) – посилання на таблицю django_content_type;
- codename (Varchar) – унікальний код дозволу.

Таблиця `django_content_type` зберігає інформацію про відстеження моделей у всіх пакетах-застосунках проєкту Django, включає поля:

- `id` (Integer) – унікальний ідентифікатор запису;
- `app_label` (Varchar) – назва застосунку;
- `model` (Varchar) – назва моделі.

Схема БД представлена нижче (рис. 3.4).

3.5 Висновки за розділом 3

В процесі проектування і реалізації програмного забезпечення було застосовано комплекс сучасних технологічних рішень та інструментів, що забезпечило досягнення високої ефективності розробленої системи. Ключову роль у створенні програмного продукту відіграла мова програмування Python, яка завдяки інтуїтивному синтаксису, численним бібліотекам та модулям надала змогу реалізувати структуровано-продуктивний код.

У якості фреймворку було обрано Django, оскільки він забезпечує високий рівень безпеки, гнучкість, масштабованість при розробці вебзастосунків.

Детально проаналізовано бізнес сценарії взаємодії HR-менеджера, працівника та адміністратора з системою, що дозволило сформулювати функціональні вимоги до програмного продукту та структурувати логіку взаємодії користувачів із ключовими модулями.

Для реалізації даного програмного продукту використовується СКБД MySQL, яка забезпечує надійне зберігання, обробку й ефективне управління даними. У межах проєкту створено низку взаємопов'язаних таблиць, що реалізують логічну схему БД відповідно до вимог предметної області. Представлено детальний опис основних таблиць БД та наведено їхню структуру, яка забезпечує цілісність даних, оптимізує взаємодію між об'єктами та спрощує реалізацію бізнес-логіки. Використання такої злагодженої системи між її компонентами забезпечує високий рівень нормалізації даних, спрощує виконання запитів і підтримує гнучкість та адаптивність системи в процесі подальшого розвитку програмного забезпечення [40].

4 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Порядок взаємодії користувача з програмою

Порядок роботи з функціями програми HR-менеджера, працівника та адміністратора показаний на (рис. 4.1).

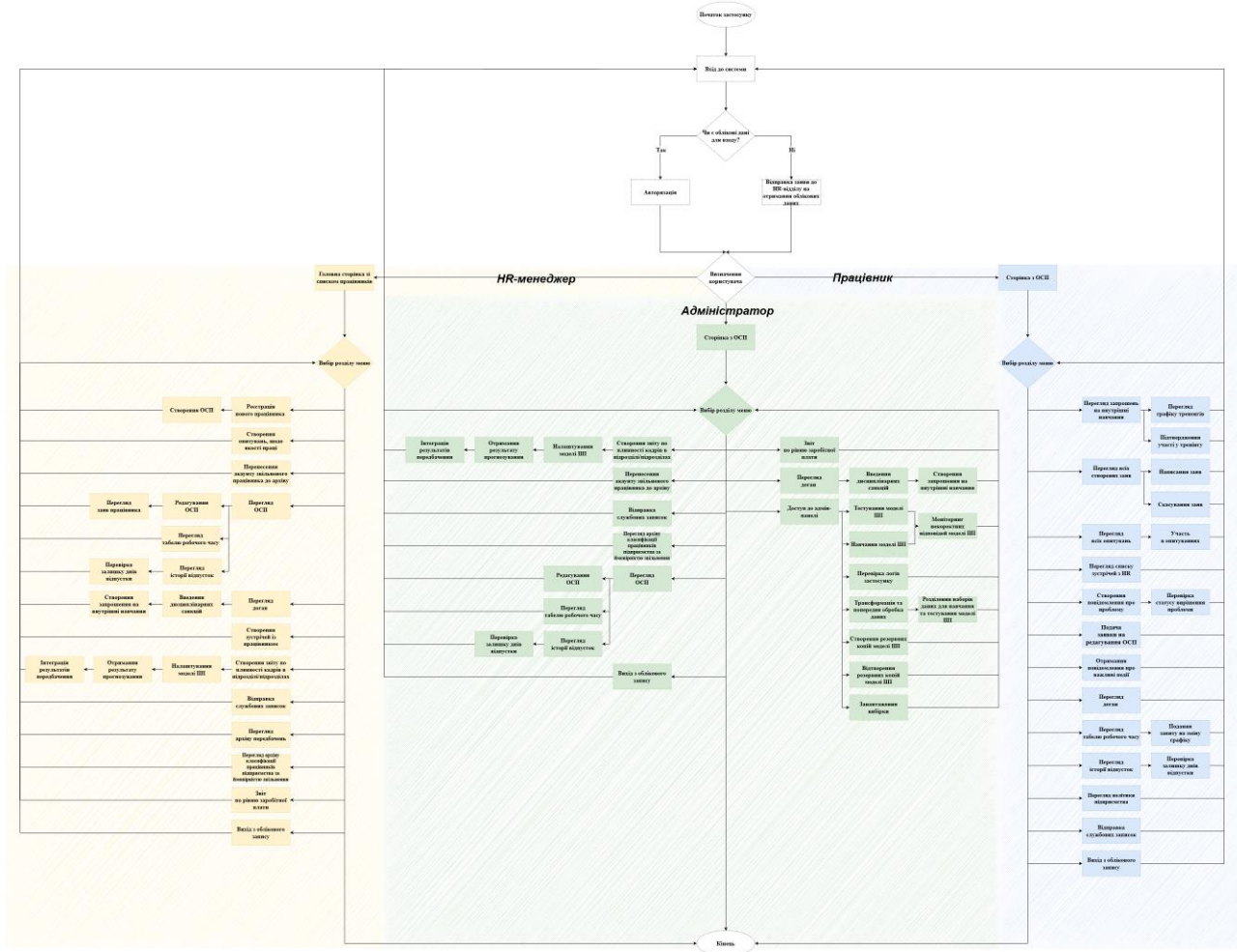


Рисунок 4.1 – Схема функціонування програми

4.2 Структура застосунку

Під час розробки проекту було застосовано фреймворк Django, який сформував структуру застосунку (рис. 4.2).

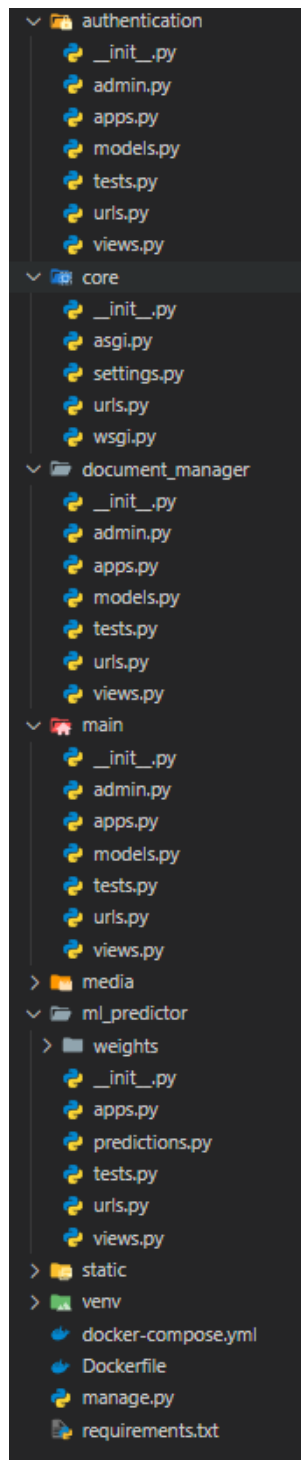


Рисунок 4.2 – Структура проєкту

Структура проєкту включає в себе такі компоненти:

- authentication/ – пакет-застосунок Django для аутентифікації;
- authentication/admin.py – налаштування адмін-панелі;
- authentication/apps.py – конфігурація застосунку;
- authentication/models.py – моделі БД;

- authentication/tests.py – тести;
- authentication/urls.py – Uniform Resource Locator (URL) адреси застосунку;
- authentication/views.py – представлення застосунку;
- core/ – проєкт Django з глобальними налаштуваннями;
- core/setting.py – налаштування проєкту;
- core/urls.py – URL-адреси проєкту;
- document_manager/ – пакет-застосунок Django для електронного документообігу;
- document_manager/admin.py – налаштування адмін-панелі;
- document_manager/apps.py – конфігурація застосунку;
- document_manager/models.py – моделі БД;
- document_manager/tests.py – тести;
- document_manager/urls.py – URL-адреси застосунку;
- document_manager/views.py – представлення застосунку;
- main/ – пакет-застосунок Django для основних сторінок сайту;
- main/admin.py – налаштування адмін-панелі;
- main/apps.py – конфігурація застосунку;
- main/models.py – моделі БД;
- main/tests.py – тести;
- main/urls.py – URL-адреси застосунку;
- main/views.py – представлення застосунку;
- media/ – тека для користувацьких файлів;
- static/ – тека статичних файлів (JS, CSS, зображення)
- templates/ – тека з глобальними HTML-шаблонами;
- ml_predictor/ – пакет-застосунок Django для машинного навчання та прогнозування плінності кадрів;
- ml_predictor/weights/ – результати навчання метода Random Forest;
- ml_predictor/apps.py – конфігурація застосунку;

- ml_predictor/predictions.py – реалізація логіки метода Random Forest;
- ml_predictor/tests.py – тести;
- ml_predictor/urls.py – URL-адреси застосунку;
- ml_predictor/views.py – представлення застосунку;
- venv/ – віртуальне оточення Python;
- docker-compose.yml – конфігурація Docker-контейнерів;
- Dockerfile – інструкції для створення Docker-образу;
- manage.py – скрипт для управління проектом Django;
- requirements.txt – перелік залежностей проекту, які потрібно встановити через pip.

4.3 Опис реалізованих класів моделей

Клас `EmployeeProfile` є похідним від базового класу `Model` фреймворку Django. Модель описує структуру таблиці `authentication_employeeprofile` в БД, що зберігає персональні дані про працівників підприємства, вона включає поля та методи:

- `first_name` (`CharField`) – ім'я працівника;
- `last_name` (`CharField`) – прізвище працівника;
- `start_date` (`DateField`) – дата початку роботи в організації;
- `exit_date` (`DateField`) – дата звільнення з організації;
- `title` (`CharField`) – назва посади працівника;
- `supervisor` (`CharField`) – ім'я керівника;
- `email` (`CharField`) – електронна адреса працівника;
- `business_unit` (`CharField`) – дільниця працівника;
- `employee_status` (`CharField`) – поточний статус працівника;
- `employee_type` (`CharField`) – тип зайнятості;
- `pay_zone` (`CharField`) – діапазон заробітної плати;

- `employee_classification_type` (`CharField`) – класифікація типу оплати;
- `termination_type` (`CharField`) – тип звільнення;
- `termination_description` (`TextField`) – додаткові деталі звільнення;
- `department_type` (`CharField`) – цех/відділ працівника;
- `division_description` (`CharField`) – опис підрозділу працівника;
- `dob` (`DateField`) – дата народження працівника;
- `state` (`CharField`) – регіон;
- `job_function` (`CharField`) – опис функцій працівника;
- `gender` (`CharField`) – стать працівника;
- `location` (`CharField`) – характер роботи;
- `race_ethnicity` (`CharField`) – етнічна належність;
- `maritar_status` (`CharField`) – сімейний стан;
- `performance_score` (`CharField`) – оцінка результативності працівника;
- `current_employee_rating` (`CharField`) – поточна оцінка працівника;
- `user` (`OneToOneField`) – посилання на стандартну модель `User`;
- `__str__()` – метод, що визначає текстове представлення об'єкта.

Клас `Training` є похідним від базового класу `Model` фреймворку Django. Модель описує структуру таблиці `authentication_training` в БД, що зберігає інформацію про тренінги, що проводяться на підприємстві, вона включає поля та методи:

- `title` (`CharField`) – назва тренінгу;
- `description` (`TextField`) – опис тренінгу;
- `start_datetime` (`DateTimeField`) – дата та час початку тренінгу;
- `end_datetime` (`DateTimeField`) – дата та час завершення тренінгу;

- location (CharField) – місце проведення;
- capacity (PositiveIntegerField) – максимальна кількість учасників;
- is_active (BooleanField) – позначає активність тренінгу;
- created_at (DateTimeField) – дата створення запису;
- updated_at (DateTimeField) – дата останнього оновлення;
- __str__() – метод, що визначає текстове представлення об'єкта;
- get_absolute_url() – метод, що використовується для формування URL для об'єкту;
- spots_left() – метод обчислює кількість вільних місць.

Клас TrainingRegistration є похідним від базового класу Model фреймворку Django. Модель описує структуру таблиці authentication_trainingregistration в БД, що зберігає інформацію про реєстрацію працівників підприємства на тренінги, вона включає поля та методи:

- user (ForeignKey(User)) – посилання на стандартну модель User;
- training (ForeignKey(Training)) – посилання на модель Training;
- registered_at (DateTimeField) – дата реєстрації;
- confirmed (BooleanField) – позначає, чи підтверджено участь;
- __str__() – метод, що визначає текстове представлення об'єкта.

Клас Document є похідним від базового класу Model фреймворку Django. Модель описує структуру таблиці document_manager_document в БД, що зберігає інформацію про скан-копії документів, які завантажують працівники підприємства, вона включає поля та методи:

- user (ForeignKey(User)) – посилання на стандартну модель User;
- title (CharField) – назва докуманта;
- description (TextField) – опис документа;

- `scan_file (FileField)` – файл документа;
- `uploaded_at (DateTimeField)` – файл документа;
- `__str__()` – метод, що визначає текстове представлення об'єкта;
- `get_absolute_url()` – метод, що використовується для формування URL для об'єкту.

Клас `Application` є похідним від базового класу `Model` фреймворку Django. Модель описує структуру таблиці `document_manager_application` в БД, що зберігає інформацію про заяви, які подані працівниками підприємства, вона включає поля та методи:

- `user (ForeignKey(User))` – посилання на стандартну модель `User`;
- `subject (CharField)` – тема заяви;
- `body (TextField)` – текст заяви;
- `category (CharField)` – категорія заяви;
- `status (CharField)` – статус заяви;
- `created_at (DateTimeField)` – дата створення;
- `updated_at (DateTimeField)` – дата оновлення;
- `__str__()` – метод, що визначає текстове представлення об'єкта;
- `get_absolute_url()` – метод, що використовується для формування URL для об'єкту.

Клас `ApplicationAttachment` є похідним від базового класу `Model` фреймворку Django. Модель описує структуру таблиці `document_manager_applicationattachment` в БД, що зберігає інформацію про скан-копії документів, що прикріплюють працівники підприємства до заяв, вона включає поля та методи:

- `application (ForeigKey(Application))` – посилання на модель `Application`;
- `file (FileField)` – завантажений файл;

- `uploaded_at` (DateTimeField) – дата завантаження;
- `__str__()` – метод, що визначає текстове представлення об'єкта.

Клас `Poll` є похідним від базового класу `Model` фреймворку Django. Модель описує структуру таблиці `main_poll` в БД, що зберігає інформацію про опитування працівників підприємства, вона включає поля та методи:

- `title` (CharField) – назва опитування;
- `description` (TextField) – опис опитування;
- `created_at` (DateTimeField) – дата створення;
- `is_active` (BooleanField) – позначає, чи активне опитування;
- `__str__()` – метод, що визначає текстове представлення об'єкта;
- `get_absolute_url()` – метод, що використовується для формування URL для об'єкту.

Клас `Question` є похідним від базового класу `Model` фреймворку Django. Модель описує структуру таблиці `main_question` в БД, що зберігає інформацію про питання конкретного опитування, вона включає поля та методи:

- `poll` (ForeignKey(Poll)) – посилання на модель `Poll`;
- `text` (CharField) – текст питання;
- `question_type` (CharField) – тип питання;
- `__str__()` – метод, що визначає текстове представлення об'єкта.

Клас `Choice` є похідним від базового класу `Model` фреймворку Django. Модель описує структуру таблиці `main_choice` в БД, що зберігає інформацію про варіанти відповідей на питання опитувань, вона включає поля та методи:

- `question` (ForeignKey(Question)) – посилання на модель `Question`;
- `text` (CharField) – текст варіанту відповіді;

– `__str__()` – метод, що визначає текстове представлення об'єкта.

Клас `Answer` є похідним від базового класу `Model` фреймворку Django. Модель описує структуру таблиці `main_answer` в БД, що зберігає інформацію про відповіді на питання опитувань працівників підприємства, вона включає поля та методи:

- `user (ForeignKey(User))` – посилання на модель `User`;
- `poll (ForeignKey(Poll))` – посилання на модель `Poll`;
- `submitted_at (DateTimeField)` – дата подання відповіді;
- `__str__()` – метод, що визначає текстове представлення об'єкта.

Клас `ChoiceAnswer` є похідним від базового класу `Model` фреймворку Django. Модель описує структуру таблиці `main_choiceanswer` в БД, що зберігає інформацію про вибіркові відповіді працівників підприємства на питання опитувань, вона включає поля та методи:

- `answer (ForeignKey(Answer))` – посилання на модель `Answer`;
- `question (ForeignKey(Question))` – посилання на модель `Question`;
- `choices (ManyToManyField(Choice))` – вибрані варіанти відповідей;
- `__str__()` – метод, що визначає текстове представлення об'єкта.

Клас `TextAnswer` є похідним від базового класу `Model` фреймворку Django. Модель описує структуру таблиці `main_textanswer` в БД, що зберігає інформацію про текстові відповіді працівників підприємства на питання опитувань, вона включає поля та методи:

- `answer (ForeignKey(Answer))` – посилання на модель `Answer`;
- `question (ForeignKey(Question))` – посилання на модель `Question`;
- `text (TextField)` – текст відповіді;

– `__str__()` – метод, що визначає текстове представлення об'єкта.

Клас `Issue` є похідним від базового класу `Model` фреймворку Django. Модель описує структуру таблиці `main_issue` в БД, що зберігає інформацію про проблеми працівників підприємства, вона включає поля та методи:

- `user (ForeignKey(User))` – посилання на модель `User`;
- `title (CharField)` – заголовок проблеми;
- `description (TextField)` – опис проблеми;
- `category (CharField)` – категорія запису;
- `status (CharField)` – статус заяви;
- `created_at (DateTimeField)` – дата створення;
- `updated_at (DateTimeField)` – дата оновлення;
- `__str__()` – метод, що визначає текстове представлення об'єкта;
- `get_absolute_url()` – метод, що використовується для формування URL для об'єкту.

Клас `DisciplinaryAction` є похідним від базового класу `Model` фреймворку Django. Модель описує структуру таблиці `main_disciplinaryaction` в БД, що зберігає інформацію про дисциплінарні санкції щодо працівника підприємства, вона включає поля та методи:

- `employee (ForeignKey(User))` – посилання на модель `User`;
- `reason (TextField)` – причина догани;
- `issued_by (ForeignKey(User))` – посилання на модель `User`;
- `date_issued (DateField)` – дата видачі;
- `expiration_date (DateField)` – дата закінчення дії;
- `is_active (BooleanField)` – чи активна догана;
- `__str__()` – метод, що визначає текстове представлення об'єкта.

Клас `CompanyPolicy` є похідним від базового класу `Model` фреймворку Django. Модель описує структуру таблиці `main_companypolicy` в

БД, що зберігає інформацію про корпоративні політики підприємства, вона включає поля та методи:

- title (CharField) – назва політики;
- category (CharField) – категорія;
- content (TextField) – зміст політики;
- document (FileField) – завантажений файл політики;
- created_by (ForeignKey(User)) – посилання на модель User;
- created_at (DateTimeField) – дата створення політики;
- updated_at (DateTimeField) – дата оновлення політики;
- completed (DateTimeField) – дата закінчення політики;
- is_active (BooleanField) – актуальність політики;
- __str__() – метод, що визначає текстове представлення об'єкта;
- get_absolute_url() – метод, що використовується для формування URL для об'єкту.

Клас HRMeeting є похідним від базового класу Model фреймворку Django. Модель описує структуру таблиці main_hrmeeting в БД, що зберігає інформацію про зустрічі HR-менеджерів з працівниками, вона включає поля та методи:

- disciplinary_action (ForeignKey(DisciplinaryAction)) – посилання на модель DisciplinaryAction;
- date (DateTimeField) – дата та час зустрічі;
- location (CharField) – місце проведення;
- notes (TextField) – примітки керівника;
- attended (BooleanField) – чи з'явився працівник;
- __str__() – метод, що визначає текстове представлення об'єкта;
- get_absolute_url() – метод, що використовується для формування URL для об'єкту.

Клас `EmployeePrediction` є похідним від базового класу `Model` фреймворку `Django`. Модель описує структуру таблиці `ml_predictor_employeeprediction` в БД, що зберігає інформацію про класифікацію ймовірності збільшення працівників підприємства, вона включає поля та методи:

- `user (ForeignKey(User))` – посилання на модель `User`;
- `prediction_date (DateTimeField)` – дата та час створення передбачення;
- `will_leave (BooleanField)` – прогноз звільнення працівника;
- `probability (FloatField)` – ймовірність звільнення працівника;
- `model_version (CharField)` – версія моделі передбачення;
- `__str__()` – метод, що визначає текстове представлення об'єкта;
- `get_absolute_url()` – метод, що використовується для формування URL для об'єкту.

4.4 Опис реалізованих класів представлень

Клас `EmployeeProfileDetailView` реалізує відображення ОСП працівника на основі `Class-Based View (CBV)`. Представлення успадковує поведінку від стандартних механізмів `Django`, таких як клас `DetailView`, що використовується для виведення детальної інформації про один об'єкт моделі `User`, та `LoginRequiredMixin`, який обмежує доступ неавторизованим користувачам до сторінки.

Клас `EmployeeProfileUpdateView` реалізує оновлення ОСП працівника на основі `Class-Based View (CBV)`. Представлення успадковує поведінку від стандартних механізмів `Django`, таких як клас `UpdateView`, що забезпечує автоматичну генерацію форми редагування на основі моделі `User` та зазначених полів, `UserPassesTestMixin` дозволяє перевірити чи належить працівник до HR-відділу через метод `test_func ()` та `LoginRequiredMixin`, який

обмежує доступ неавторизованим користувачам до сторінки. У межах класу перевизначено `get_success_url ()` визначає URL, на який працівник буде переправлений після успішного оновлення профілю.

Клас `TrainingListView` реалізує відображення списку тренінгів на основі CBV. Представлення успадковує поведінку від стандартних механізмів Django, таких як клас `ListView`, що забезпечує автоматичне формування сторінки зі списком об'єктів моделі `Training`, та `LoginRequiredMixin`, який обмежує доступ неавторизованим користувачам до сторінки. У межах класу встановлено пагінацію по десять записів на сторінку та перевизначено метод `get_queryset ()`, щоб обрати лише активні навчальні заходи й впорядкувати за датою їх створення.

Клас `TrainingCreateView` реалізує створення запису про новий тренінг на основі CBV. Представлення успадковує поведінку від стандартних механізмів Django, таких як клас `CreateView`, призначений для створення нового об'єкту моделі `Training` через веб-форму, `UserPassesTestMixin`, який дозволяє перевірити чи належить працівник до HR-відділу через метод `test_func()`, та `LoginRequiredMixin`, який обмежує доступ неавторизованим користувачам до сторінки. У межах класу перевизначено метод `get_success_url ()`, що визначає URL, на який працівник буде переправлений після заповнення форми.

Клас `TrainingRegistrationCreateView` реалізує створення запису про реєстрацію працівника підприємства на тренінг на основі CBV. Представлення успадковує поведінку від стандартних механізмів Django, таких як клас `CreateView`, призначений для створення нового об'єкту моделі `TrainingRegistration` через вебформу, та `LoginRequiredMixin`, який обмежує доступ неавторизованим користувачам до сторінки. У межах класу перевизначено метод `form_valid ()`, що призначає авторизованого працівника в поле `user` нового запису реєстрації, та `get_success_url ()`, що визначає URL, на який працівник буде переправлений після заповнення форми.

Клас `TrainingRegistrationListView` реалізує відображення списку реєстрацій на тренінги на основі CBV. Представлення успадковує поведінку від стандартних механізмів Django, таких як клас `ListView`, що забезпечує автоматичне формування сторінки зі списком об'єктів моделі `TrainingRegistration`, та `LoginRequiredMixin`, який обмежує доступ неавторизованим користувачам до сторінки. У межах класу встановлено пагінацію по десять записів на сторінку та перевизначено метод `get_queryset()`, щоб обмежити вибірку для звичайних працівників, а для HR-відділу – розширити.

Клас `DocumentCreateView` реалізує завантаження скан-копії документів на основі CBV. Представлення успадковує поведінку від стандартних механізмів Django, таких як клас `CreateView`, призначений для створення нового об'єкту моделі `Document` через вебформу, та `LoginRequiredMixin`, який обмежує доступ неавторизованим користувачам до сторінки. У межах класу перевизначено метод `form_valid()`, що призначає авторизованого працівника в поле `user` нового запису скан-копії, та `get_success_url()`, що визначає URL, на який працівник буде переправлений після заповнення форми.

Клас `DocumentDeleteView` реалізує видалення скан-копії документа на основі CBV. Представлення успадковує поведінку від стандартних механізмів Django, таких як клас `DeleteView`, що автоматично обробляє видалення об'єкту моделі `Document`, та `LoginRequiredMixin`, який обмежує доступ неавторизованим користувачам до сторінки. У межах класу перевизначено метод `get_queryset()`, що запобігає видаленню чужих документів, та `get_success_url()`, що визначає URL, на який працівник буде переправлений після успішного видалення.

Клас `DocumentListView` реалізує відображення списку скан-копій документів працівників на основі CBV. Представлення успадковує поведінку від стандартних механізмів Django, таких як клас `ListView`, що забезпечує автоматичне формування сторінки зі списком об'єктів моделі `Document`, та

LoginRequiredMixin, який обмежує доступ неавторизованим користувачам до сторінки. У межах класу встановлено пагінацію по десять записів на сторінку та перевизначено метод `get_queryset ()`, щоб обмежити вибірку для звичайних працівників, а для HR-відділу – розширити.

Клас `EmployeePredictionCreateView` реалізує створення запису прогнозів плинності кадрів на основі CBV. Представлення успадковує поведінку від стандартних механізмів Django, таких як клас `CreateView`, призначений для створення нового об'єкту моделі `EmployeePrediction` через веб-форму, `UserPassesTestMixin`, що дозволяє перевірити, чи належить працівник до HR-відділу через метод `test_func ()`, та `LoginRequiredMixin`, який обмежує доступ неавторизованим користувачам до сторінки. У межах класу перевизначено метод `form_valid ()`, він призначає авторизованого працівника в поле `user` нового запису.

Клас `EmployeePredictionListView` реалізує відображення прогнозів плинності кадрів на основі CBV. Представлення успадковує поведінку від стандартних механізмів Django, таких як клас `ListView`, що забезпечує автоматичне формування сторінки зі списком об'єктів моделі `EmployeePrediction`, `UserPassesTestMixin`, що дозволяє перевірити, чи належить працівник до HR-відділу через метод `test_func ()`, та `LoginRequiredMixin`, який обмежує доступ неавторизованим користувачам до сторінки. У межах класу встановлено пагінацію по десять записів на сторінку.

Клас `EmployeePredictionUpdateView` реалізує оновлення прогнозу плинності кадрів на основі CBV. Представлення успадковує поведінку від стандартних механізмів Django, таких як клас `UpdateView`, що забезпечує автоматичну генерацію форми редагування на основі моделі `EmployeePrediction` та зазначених полів, `UserPassesTestMixin`, що дозволяє перевірити, чи належить працівник до HR-відділу через метод `test_func ()`, та `LoginRequiredMixin`, який обмежує доступ неавторизованим користувачам до

сторінки. У межах класу перевизначено `get_success_url ()`, що визначає URL, на який працівник буде переправлений після успішного оновлення профілю.

Клас `EmployeePredictionDeleteView` реалізує видалення прогнозу плинності кадрів на основі CBV. Представлення успадковує поведінку від стандартних механізмів Django, таких як клас `DeleteView`, що автоматично обробляє видалення об'єкту моделі `EmployeePrediction`, `UserPassesTestMixin`, що дозволяє перевірити, чи належить працівник до HR-відділу через метод `test_func ()`, та `LoginRequiredMixin`, який обмежує доступ неавторизованим користувачам до сторінки. У межах класу перевизначено метод `get_success_url ()`, що визначає URL, на який працівник буде переправлений після успішного видалення.

Клас `ApplicationCreateView` реалізує створення запису заяв на основі CBV. Представлення успадковує поведінку від стандартних механізмів Django, таких як клас `CreateView`, призначений для створення нового об'єкту моделі `Application` через вебформу, та `LoginRequiredMixin`, який обмежує доступ неавторизованим користувачам до сторінки. У межах класу перевизначено метод `form_valid ()`, що призначає авторизованого працівника в поле `user` нового запису скан-копії, та `get_success_url ()`, що визначає URL, на який працівник буде переправлений після заповнення форми.

Клас `ApplicationDetailView` реалізує відображення заяв працівника на основі CBV. Представлення успадковує поведінку від стандартних механізмів Django, таких як клас `DetailView`, що використовується для виведення детальної інформації про один об'єкт моделі, та `LoginRequiredMixin`, який обмежує доступ неавторизованим користувачам до сторінки. У межах класу перевизначено метод `get_queryset ()`, щоб обмежити вибірку лише тими заявами, які належать поточному працівнику, а для HR-відділу – доступні усі заяви.

Клас `PollCreateView` реалізує створення запису опитування на основі CBV. Представлення успадковує поведінку від стандартних механізмів Django, таких як клас `CreateView`, призначений для створення нового об'єкту

моделі Poll через вебформу, `UserPassesTestMixin`, що дозволяє перевірити, чи належить працівник до HR-відділу через метод `test_func ()`, та `LoginRequiredMixin`, який обмежує доступ неавторизованим користувачам до сторінки. У межах класу перевизначено метод `form_valid()`, що призначає авторизованого працівника в поле `user` нового запису опитування, та `get_success_url ()`, що визначає URL, на який працівник буде переправлений після заповнення форми.

Клас `QuestionCreateView` реалізує створення запису питання для опитування на основі CBV. Представлення успадковує поведінку від стандартних механізмів Django, таких як клас `CreateView`, призначений для створення нового об'єкту моделі `Question` через вебформу, `UserPassesTestMixin`, що дозволяє перевірити чи належить працівник до HR-відділу через метод `test_func ()`, та `LoginRequiredMixin`, який обмежує доступ неавторизованим користувачам до сторінки. У межах класу перевизначено метод `form_valid ()`, що призначає авторизованого працівника в поле `user` нового запису опитування, `dispatch ()`, що отримує конкретне опитування з моделі `Poll`, щоб прив'язати до нього питання, та `get_success_url ()`, що визначає URL, на який працівник буде переправлений після заповнення форми.

Клас `AnswerCreateView` реалізує створення запису відповіді користувача на питання на основі CBV. Представлення успадковує поведінку від стандартних механізмів Django, таких як клас `CreateView`, призначений для створення нового об'єкту моделі `Answer` через вебформу, та `LoginRequiredMixin`, який обмежує доступ неавторизованим користувачам до сторінки. У межах класу перевизначено метод `form_valid ()`, що призначає авторизованого працівника в поле `user` нового запису опитування, `dispatch ()`, що отримує конкретне опитування та питання з моделей `Poll` і `Question`, щоб прив'язати їх до відповіді працівника, та `get_success_url ()`, що визначає URL, на який працівник буде переправлений після заповнення форми.

4.5 Опис реалізованих класів III

Клас `EmployeeTurnoverModel` є об'єктно-орієнтованою реалізацією ансамблевого методу `Random Forest` у проєкті `Django`, який використовується для класифікації працівників підприємства за ймовірністю звільнення. У своїй структурі розроблений клас поєднує функціональність різних бібліотек мови програмування `Python` – `scikit-learn`, `NumPy` та `pandas`, що забезпечує ефективну обробку даних, нормалізацію ознак та побудову оптимальної моделі машинного навчання. В основі використовується `RandomForestClassifier` із бібліотеки `sklearn.ensemble`, який формує ансамбль незалежних дерев рішень. Клас реалізує методи:

- метод `preprocess` виконує попередню обробку даних, стандартизуючи вхідні значення з метою зниження впливу масштабних відмінностей між ознаками на процес навчання;
- метод `train` є основною процедурою навчання, він поділяє вихідний набір даних на навчальну та тестову вибірки за допомогою функції `train_test_split`. Застосовується відбір ознак, що дозволяє підвищити інтерпретованість результатів та зменшити ризик перенавчання;
- метод `predict` забезпечує прогнозування плинності кадрів на основі вхідного масиву характеристик працівника, виконуючи попередню перевірку факту навчання моделі;
- метод `evaluate` обчислює метрики точності, формує узагальнений звіт та матрицю помилок, що надає аналіз результатів передбачень.

4.6 Висновки за розділом 4

У четвертому розділі дипломної кваліфікаційної роботи було здійснено практичну реалізацію ПЗ для автоматизації процесів управління

персоналом, що передбачає створення та опис функціональних можливостей програми для HR-менеджера і працівника.

Структура проєкту побудована на основі фреймворку Django за модульним принципом і включає низку взаємопов'язаних компонентів, кожен із яких виконує окрему функціональну роль у роботі системи. Така структура забезпечує логічну організацію коду, зручність супроводу, ефективну інтеграцію між усіма її компонентами.

Особливу увагу приділено опису класів моделей, які відповідають за зберігання й оброблення інформації про працівників, особисті дані, заяви, опитування, скан-копії документів, збереження передбачень плинності персоналу. Реалізовані моделі забезпечують узгодженість даних, використовуючи механізми валідації та зв'язків між сутностями БД.

Приділено увагу класам представлень, які формують логіку оброблення запитів користувачів, розділення повноважень між працівниками, керують відображенням інформації та забезпечують інтерактивність інтерфейсу.

Важливим аспектом розробки стала інтеграція класів ШІ з проєктом Django, що забезпечила автоматизацію складних аналітичних процесів, зокрема прогнозування показників плинності кадрів. Клас `EmployeeTurnoverModel` реалізує повний цикл машинного навчання – від попередньої обробки даних до аналізу ефективності побудованої моделі, забезпечуючи практичну придатність у наукових та прикладних дослідженнях з прогнозування поведінкових характеристик персоналу.

5 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

5.1 Опис програми

ПЗ розроблено з метою комплексної автоматизації процесів управління персоналом на підприємстві. Його функціональні можливості спрямовані на підвищення ефективності роботи кадрових служб, оптимізацію інформаційних потоків та забезпечення прозорості управлінських рішень у сфері ЛР.

Система підтримує повноцінний електронний документообіг, що дозволяє мінімізувати використання паперових носіїв, прискорити процеси узгодження та підписання документів, а також забезпечити їх централізоване зберігання і доступність у межах встановлених прав доступу.

Передбачено зручні інструменти для пошуку, перегляду та обробки інформації щодо кожного працівника. У БД зберігаються відомості про особисті дані співробітників, їх трудову історію, результати атестацій, показники продуктивності праці, кваліфікаційний рівень і поведінкові характеристики. Це дає змогу формувати аналітичні звіти, виявляти тенденції в розвитку персоналу та підтримувати процес прийняття організаційних рішень на основі достовірних даних.

Окремий модуль системи реалізує моніторинг плинності кадрів, що забезпечує аналіз динаміки змін у кадровому складі підприємства, визначення факторів, які впливають на стабільність колективу та розроблення своєчасних заходів щодо підвищення рівня утримання персоналу. Як у цілому, так і щодо кожного окремого працівника. Це все спрямоване на стратегічну орієнтацію для розвитку кадрового потенціалу підприємства.

5.2 Експлуатація програми

Для початку роботи з вебзастосунком "HRFactory" користувачеві необхідно перейти за відповідною URL-адресою, яка надає доступ до системи. Після завантаження сторінки відкривається форма автентифікації (рис. 5.1), що забезпечує перевірку облікових даних користувача та гарантує безпечний доступ до інформаційних ресурсів застосунку.

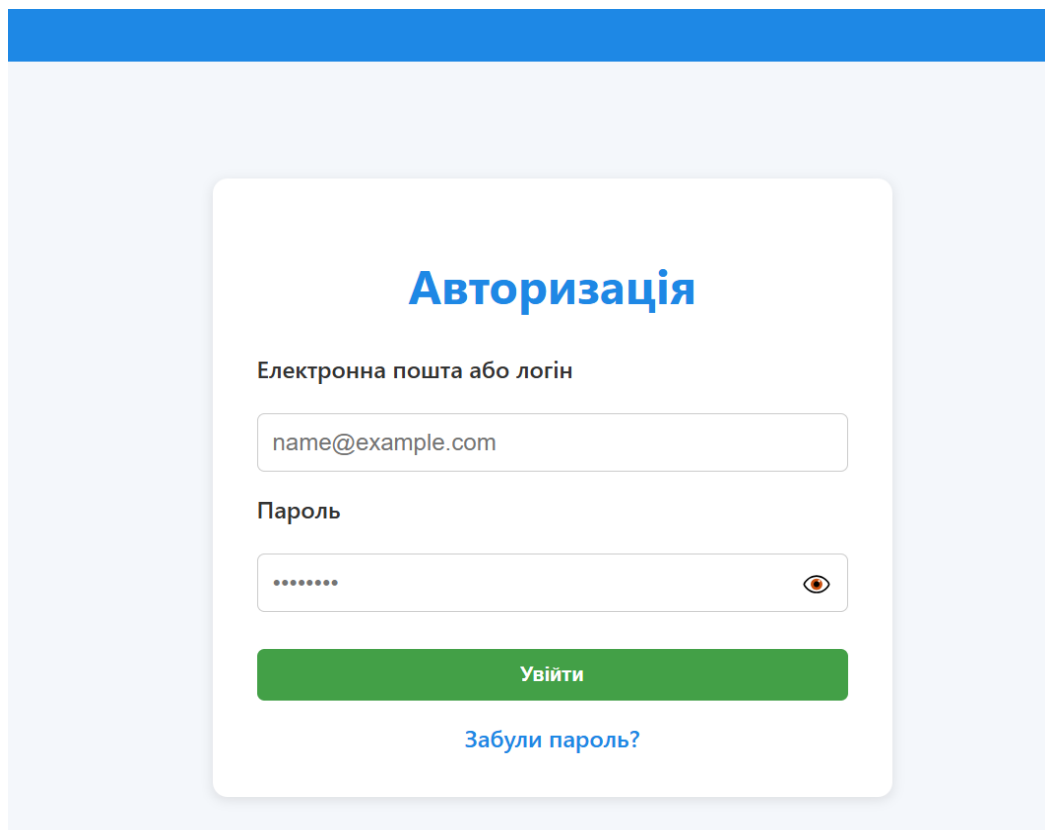


Рисунок 5.1 – Форма авторизації

Після успішного проходження етапу автентифікації користувача, система автоматично перенаправляє його на головну сторінку інтерфейсу, структура та функціональні можливості якої залежать від ролі користувача в системі. Зокрема, якщо авторизується HR-менеджер (рис. 5.2), інтерфейс містить інструменти для управління персоналом, перегляд ОСП, формування звітів, реєстрація нових працівників тощо.

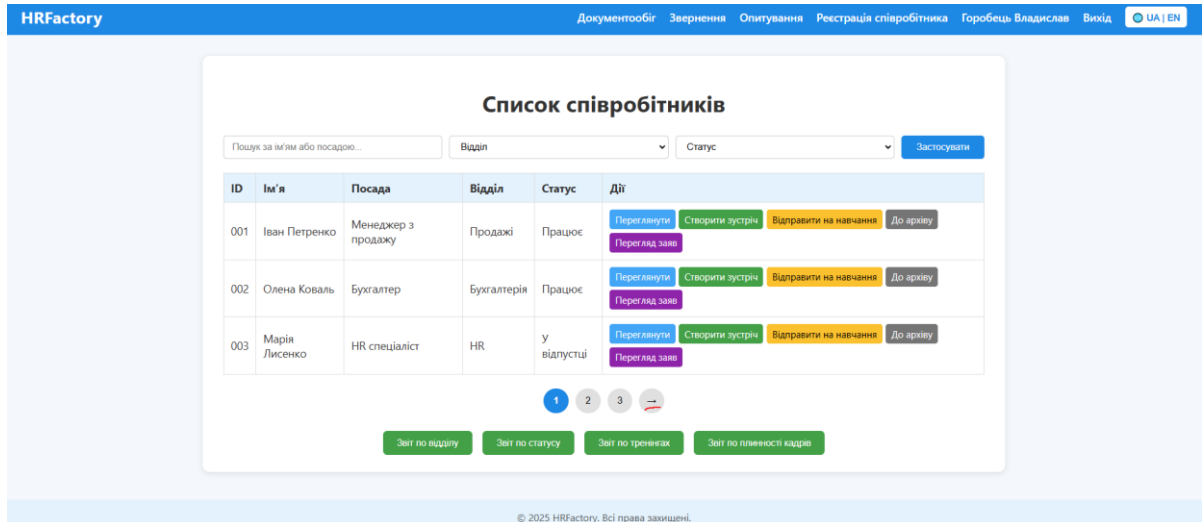


Рисунок 5.2 – Головна сторінка HR-менеджера

У випадку, коли до системи входить працівник, головна сторінка (рис. 5.3) відобразить інший набір функцій, орієнтований на перегляд особистих даних, подачу заяв, участь в опитуваннях тощо.

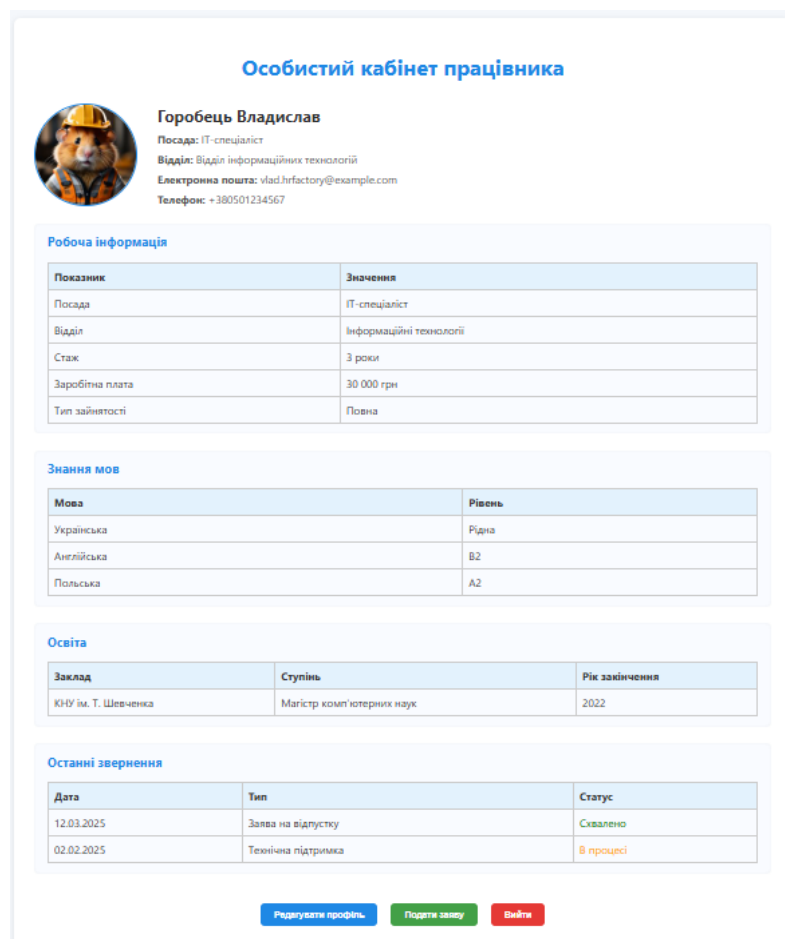
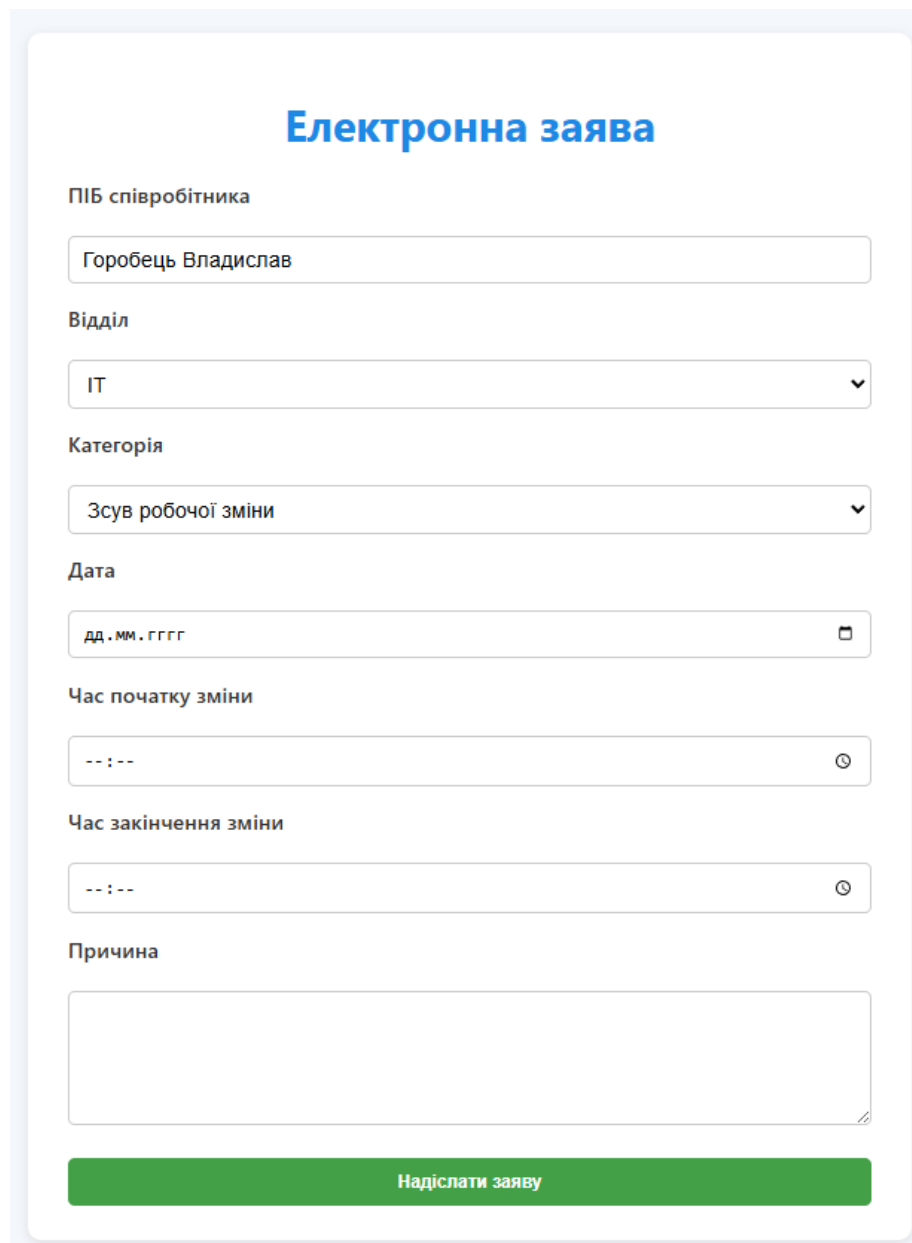


Рисунок 5.3 – Особистий кабінет працівника

Працівник має можливість створювати та подавати електронні заяви через інтерфейс застосунку. Система передбачає подання запитів на надання відпустки, зміну або зсув робочої зміни, а також інші типи внутрішніх звернень, пов'язаних із кадровими процесами (рис. 5.4). Кожна заява автоматично фіксується в БД і надсилається на розгляд відповідальній посадовій особі.




The image shows a web form titled "Електронна заява" (Electronic Request). The form contains the following fields:

- ПІБ співробітника** (Employee Name): A text input field containing "Горобець Владислав".
- Відділ** (Department): A dropdown menu with "ІТ" selected.
- Категорія** (Category): A dropdown menu with "Зсув робочої зміни" (Shift Change) selected.
- Дата** (Date): A date picker field showing "дд.мм.гггг".
- Час початку зміни** (Start Time): A time picker field showing "-- : --".
- Час закінчення зміни** (End Time): A time picker field showing "-- : --".
- Причина** (Reason): A large text area for providing details.
- Надіслати заяву** (Submit Request): A green button at the bottom.

Рисунок 5.4 – Форма електронної заяви

У разі необхідності реєстрації нових працівників HR-менеджер переходить до спеціалізованої сторінки, яка реалізована у вигляді електронної анкети (рис. 5.5). У цій формі передбачено поля для внесення основних персональних та професійних даних працівника, що забезпечує формування структурованого запису в БД системи та подальше використання інформації в процесах управління персоналом.

Реєстрація нового працівника



Выберите файл | Файл не выбран

Особиста інформація

ПІБ:
 Місце народження:

Дата народження:
 Номер телефону:

Кількість дітей:
 Сімейний статус:

Робоча інформація

Відділ:
 Заробітна плата:

Стаж (роки):

Знання мов [Додати мову](#)

Мова	Рівень	Дія
<input type="text" value="Українська"/>	<input type="text" value="Високий"/>	<input type="button" value="Видалити"/>
<input type="text" value="Українська"/>	<input type="text" value="Високий"/>	<input type="button" value="Видалити"/>

Освіта [Додати освіту](#)

Університет / заклад	Ступінь	Дія
<input type="text" value="КНУ"/>	<input type="text" value="Магістр"/>	<input type="button" value="Видалити"/>
<input type="text" value="КНУ"/>	<input type="text" value="Магістр"/>	<input type="button" value="Видалити"/>

Колішні місця роботи [Додати місце роботи](#)

Компанія	Період	Дія
<input type="text" value="ТОВ 'Продажі Плюс'"/>	<input type="text" value="2015–2018"/>	<input type="button" value="Видалити"/>
<input type="text" value="ТОВ 'Продажі Плюс'"/>	<input type="text" value="2015–2018"/>	<input type="button" value="Видалити"/>

Рисунок 5.5 – Реєстрація нового співробітника

HR-менеджер володіє інструментами для формування та аналізу різноманітних звітів, що відображають ключові показники стану й розвитку персоналу в організації. Він може створювати звіт про плинність кадрів (рис. 5.6).

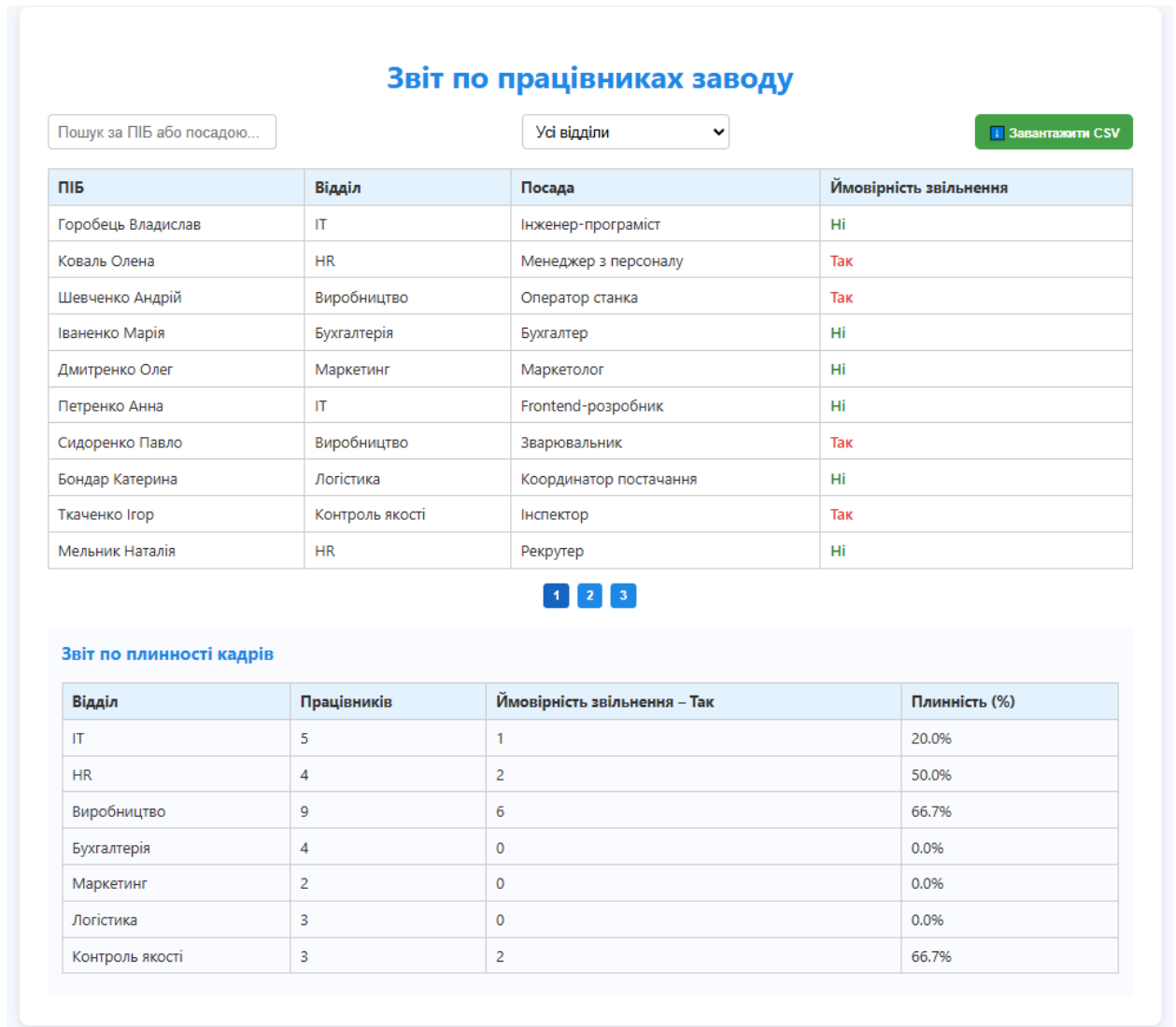


Рисунок 5.6 – Звіт по плинності кадрів

Крім того, HR-фахівець формує звітність за даними з різних джерел, інтегруючи інформацію з кадрових, бухгалтерських і виробничих систем. Одним з прикладів є звіт про заробітну плату за підрозділами, який відображає рівень оплати праці та її розмір між різними категоріями працівників (рис. 5.7).

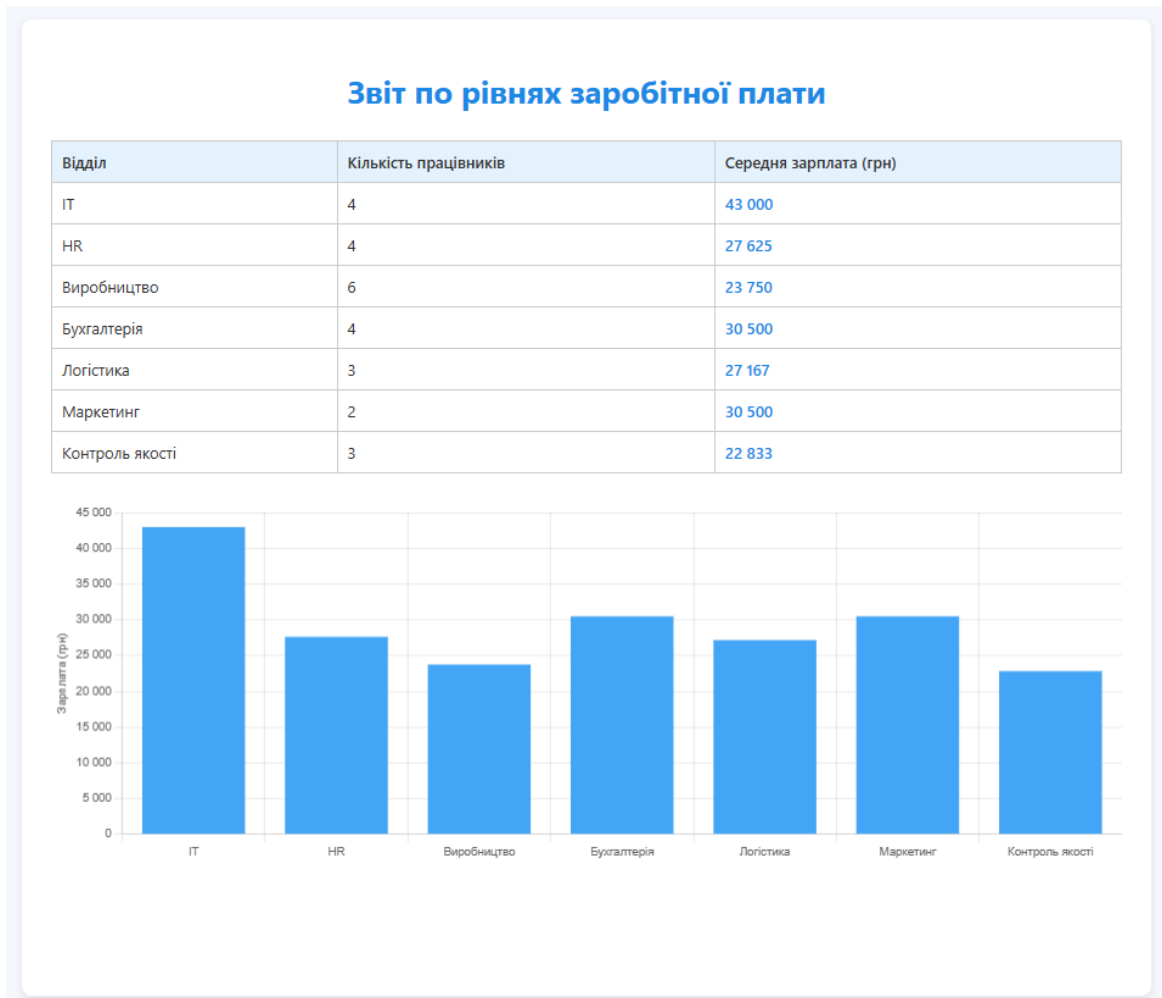


Рисунок 5.7 – Звіт по рівнях заробітної плати

HR-менеджер здійснює перенесення відомостей про працівників, які припинили трудові відносини з підприємством, до архіву, з метою збереження, систематизації та подальшого використання інформації (рис. 5.8)

Архів звільнених працівників

ПІВ	Відділ	Посада	Дата звільнення	Причина
Іваненко Олександр	Продажі	Менеджер з продажу	2025-03-12	Скорочення штату
Пітрченко Наталія	HR	HR-менеджер	2025-05-01	Вихід на пенсію
Сидоренко Андрій	IT	Програміст	2025-04-20	Звільнення за власним бажанням
Ковальчук Ольга	Бухгалтерія	Бухгалтер	2025-02-15	Сваймі обставини
Мельник Сергій	Маркетинг	Маркетолог	2025-06-10	Порушення правил
Горбачев Владислав	IT	Системний адміністратор	2025-01-30	Вихід на пенсію
Ткаченко Марія	Продажі	Керівник відділу	2025-03-22	Скорочення штату
Бондаренко Ігор	Виробництво	Оператор станка	2025-04-05	Звільнення за власним бажанням
Козак Лілія	HR	Рекрутер	2025-06-01	Сваймі обставини
Даниленко Олег	Логістика	Логіст	2025-05-18	Порушення правил
Шевченко Наталя	Маркетинг	Дизайнер	2025-02-28	Звільнення за власним бажанням
Романенко Віктор	Виробництво	Майстер зварки	2025-03-05	Вихід на пенсію
Кравченко Ірина	Бухгалтерія	Бухгалтер	2025-04-12	Сваймі обставини
Мороз Юрій	IT	Тестувальник	2025-06-15	Скорочення штату
Левченко Оксана	HR	HR-спеціаліст	2025-01-28	Порушення правил
Федоренко Дмитро	Виробництво	Інженер	2025-05-25	Звільнення за власним бажанням
Гончаренко Світлана	Маркетинг	Маркетолог	2025-02-10	Сваймі обставини
Клименко Роман	Продажі	Менеджер з продажу	2025-03-18	Скорочення штату
Павленко Ірина	IT	Аналітик	2025-04-30	Вихід на пенсію
Савченко Олег	Виробництво	Оператор	2025-06-05	Звільнення за власним бажанням

Рисунок 5.8 – Архів звільнених працівників

Після авторизації адміністратор потрапляє до адмін-панелі (рис. 5.9), де йому надається широкий спектр функціональних можливостей для управління та підтримки моделі ШІ. Адміністратор може тестувати та навчати модель ШІ, перевіряти логи, розділяти набори даних, моніторити некоректні відповіді, обробляти дані, створювати резервні копії.

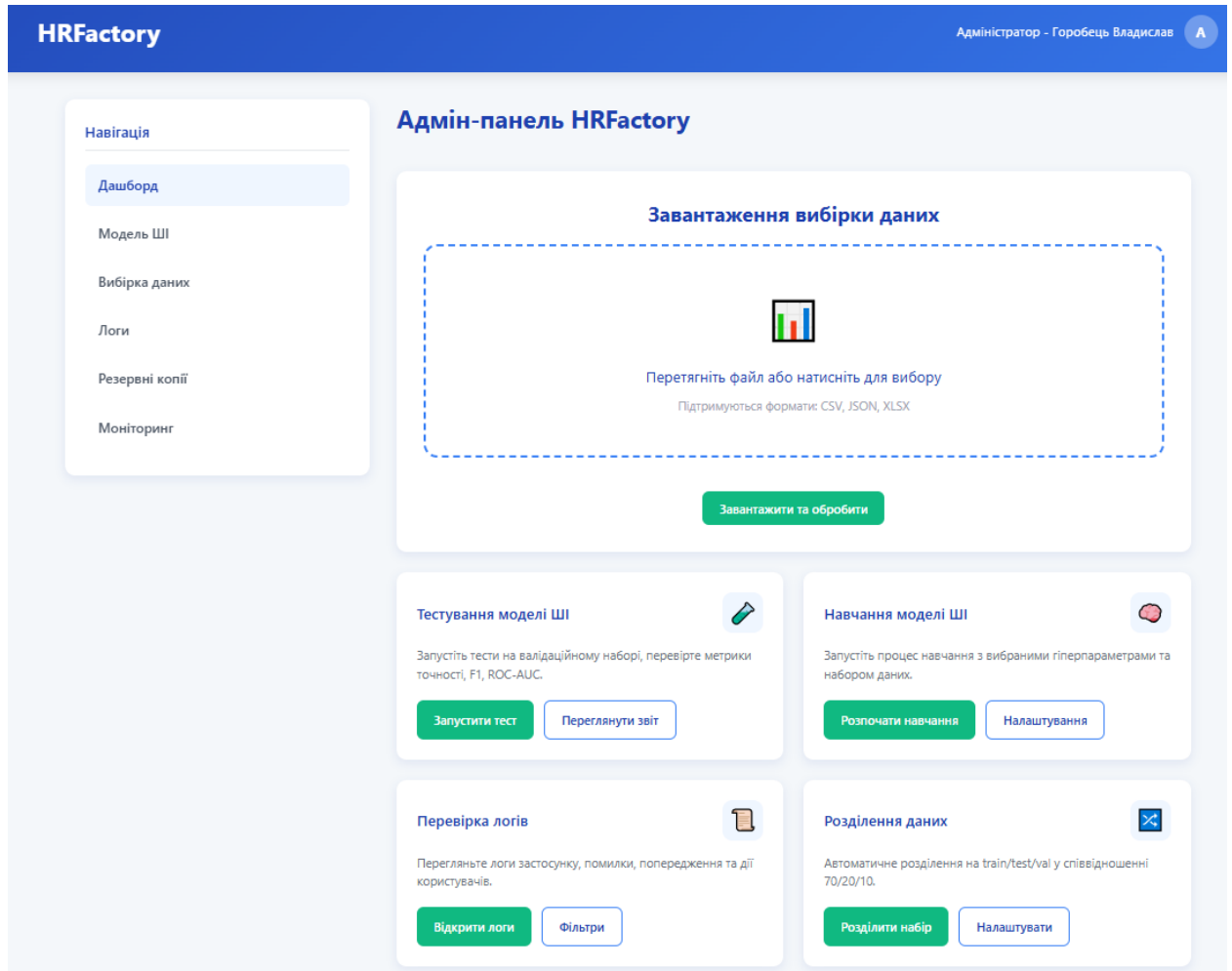


Рисунок 5.9 – Адмін-панель

5.3 Експериментальне дослідження

На початковому етапі експериментального дослідження здійснено завантаження відкритого набору даних Employee/HR Dataset (ALL in One), який містить інформацію про персонал підприємства у форматі CSV [41]. До складу даних входили змінні, що характеризують особисті, професійні та

адміністративні атрибути співробітників. Для імпортування даних у середовище Python використовувалися функціональні можливості бібліотеки pandas, зокрема метод `read_csv()`, який забезпечує зчитування інформації з CSV-файлів безпосередньо у структуру DataFrame.

На етапі очищення та корекції даних було проведено перевірку на наявність пропущених значень у атрибутах. Виявлені пропуски були замінені медіанними значеннями. Було виконано видалення дубльованих записів та скореговані категоріальні змінні, що мали різні формати запису.

Вибір ознак проводиться, щоб визначити найбільш релевантні змінні, які мають статистично значущий вплив на залежну змінну Turnover. Було використано клас `SelectKBest` з бібліотеки `scikit-learn` [42], що призначений для відбору ознак з параметрами `score_func = "chi2"` та `k = "all"`. Даний підхід дозволяє кількісно оцінити силу зв'язку між кожною незалежною змінною та цільовою ознакою, що є оптимальним для завдань класифікації. Після розрахунків було відібрано підмножину найбільш інформативних параметрів (табл. 5.1).

Таблиця 5.1 – Оцінка важливості ознак

Ознака	Статистичний критерій (χ^2)
1	2
Pay_Zone	5,1594
Performance_Score	3,5923
Age_Status	1,3811
Title	1,2955
Work_Experience_Status	0,9397
Marital_Status	0,8429
Employee_Status	0,6509
Department_Type	0,2352

Кінець таблиці 5.1

1	2
Employee_Classification_Type	0,2137
Current_Employee_Rating	0,1406
Business_Unit	0,1228
Employee_Type	0,0908
Gender	0.0260
State	0.0132
Race	0.0004

Після проведення відбору релевантних ознак дані були розподілені на навчальну та тестову вибірки у пропорції 80% до 20% відповідно. Такий розподіл забезпечує достатній обсяг даних для навчання моделі, водночас залишаючи частину даних для оцінки її якості та узагальнювальної здатності. Для реалізації цієї операції було використано функціональність бібліотеки `scikit-learn`, а саме функцію `train_test_split`, яка дозволяє випадковим чином розподілити спостереження між вибірками.

Переходячи до шостого етапу, створюється модель машинного навчання на основі алгоритму `Random Forest` для прогнозування плинності кадрів. Для реалізації моделі використовується бібліотека `scikit-learn`, зокрема клас `RandomForestClassifier`. Для налаштування моделі задаються ключові параметри, кількість дерев у ансамблі встановлюється на рівні 100, що дозволяє досягти балансу між точністю та обчислювальною ефективністю. Максимальна глибина дерев обмежується значенням 10, щоб запобігти надмірному пристосуванню моделі до навчальних даних.

На наступному етапі було проведено побудову та навчання моделей прогнозування з використанням бібліотеки `scikit-learn`. Для порівняння ефективності застосовувалися три алгоритми машинного навчання: `RandomForestClassifier` у складі запропонованого методу класифікації

працівників підприємства за ймовірністю звільнення, DecisionTreeClassifier та LogisticRegression. Кожна модель навчалася на однаковому наборі даних з однаковими параметрами, що забезпечувало об'єктивність дослідження (табл. 5.2).

Таблиця 5.2 – Ефективність застосування алгоритмів ШІ

Метод / модель	Accuracy	Precision	Recall	F1-score
Запропонований метод класифікації працівників підприємства за ймовірністю звільнення (Random Forest)	0,8966	0,8732	0,8957	0,8794
Decision Tree	0,7866	0,7515	0,8114	0,7803
Logistic Regression	0,8100	0,7845	0,8185	0,8011

За результатами експериментального дослідження встановлено, що модель Random Forest у складі запропонованого методу класифікації працівників підприємства за ймовірністю звільнення разом зі всіма його відповідними етапами продемонструвала найвищу точність прогнозування плинності кадрів, забезпечуючи оптимальний баланс між ключовими показниками оцінки. Висока стійкість моделі до шуму та здатність ефективно працювати з великою кількістю нелінійних змінних підтверджують доцільність застосування ансамблевих методів для прогнозування відтоку працівників. Підвищення точності класифікації відповідно призводить до покращення інформативності рішень, які приймаються стосовно управління персоналом, базуючись на більш точній інформації про плинність кадрів. Виразенням такого збільшення обґрунтованості стратегій є посилення

ефективності управління персоналом. Це має призводити до зниження витрат, пов'язаних із заміною працівників, а також до своєчасного виявлення ризиків втрати ключових кадрів.

5.4 Висновки за розділом 5

У п'ятому розділі дипломної кваліфікаційної роботи представлено детальний опис ПЗ, що було розроблено, яке призначене для комплексної автоматизації процесів управління персоналом на підприємстві. Проведена розробка показує ефективну інтеграцію сучасних технологій у сферу кадрового менеджменту, що сприяє підвищенню ефективності роботи HR-служб та оптимізації інформаційних потоків. Особливу увагу приділено модулю моніторингу плинності кадрів, який дає змогу здійснювати глибокий аналіз змін у структурі персоналу, визначати чинники, що впливають на стабільність колективу.

Детально розглянуто процес експлуатації вебдодатку "HRFactory" користувачами різних рівнів доступу. Інтерфейс відзначається структурованістю та адаптивністю – він автоматично підлаштовується під функціональні обов'язки користувача, що сприяє ефективній роботі як адміністраторів, кадрових менеджерів, так і рядових працівників.

У розділі представлено результати експериментального дослідження, спрямованого на побудову, вибір ознак, навчання та оцінювання моделі прогнозування плинності кадрів на основі сучасних методів машинного навчання. У результаті застосування запропонованого методу класифікації працівників підприємства за ймовірністю звільнення було отримано модель, здатну здійснювати класифікацію і відповідно прогноз загальної кадрової динаміки з високим рівнем точності.

ВИСНОВКИ

У процесі розробки програмного забезпечення в рамках дипломної кваліфікаційної роботи було виконано низку завдань, спрямованих на досягнення поставленої мети дослідження. Проведено детальний аналіз предметної області, що охоплює проблему кадрової політики на підприємстві. Проаналізовано особливості функціонування сучасних інформаційних HR-систем, які здатні ефективно підтримувати управління персоналом та прогнозування його відтоку. Здійснено порівняльний аналіз популярних платформ – BambooHR, Zoho People та КАДРОВИК.UA / КАДРИ.UA – щодо розроблюваного застосунку HRFactory. Для кожної системи проведено оцінку функціональних можливостей, орієнтованих на підтримку HR-процесів, таких як самообслуговування працівників, управління кадровою інформацією, автоматизація електронного документообігу та аналітична обробка даних. Аналіз виявив певні обмеження та недоліки аналогів, серед яких: немає безкоштовних версій, відсутність інтеграцій з ШІ, складність експлуатації ПЗ, низький рівень адаптивності.

У другому розділі для прогнозування процесу плинності кадрів було використано набір даних Employee/HR Dataset (ALL in One), що містить повну інформацію про співробітників та відповідає критеріям, необхідним для побудови моделі машинного навчання. Забезпечено формулювання задачі прогнозування плинності кадрів на підприємстві. Здійснено аналіз і порівняння основних моделей ШІ, серед яких розглянуто Linear Regression, Decision Tree та Random Forest. На основі порівняльного аналізу визначено, що алгоритм Random Forest є оптимальним для реалізації поставленої задачі. Було розкрито основні етапи побудови методу класифікації працівників підприємства за ймовірністю звільнення, даний підхід передбачає послідовну обробку даних, їх очищення та трансформацію для подальшого розподілу на відокремлені набори для навчання та тестування.

В процесі розробки ПЗ застосовано сучасні технології та інструменти. Мова програмування Python, завдяки широкому набору бібліотек, дозволила реалізувати структурований та оптимізований код, а фреймворк Django сприяє конфігурованості та масштабованості застосунку. Проведено аналіз бізнес-сценаріїв взаємодії HR-менеджера, адміністратора та працівника, що визначило функціональні вимоги та структуру логіки користувацького інтерфейсу. Для зберігання та обробки даних використано СКБД MySQL із розробленою системою взаємопов'язаних таблиць, що забезпечує цілісність, оптимізацію запитів та гнучкість системи для її ефективного використання.

Представлено практичну реалізацію ПЗ, а саме розроблення та опис функціональних можливостей системи, орієнтованої на потреби HR-менеджера та працівників організації. Особливу увагу приділено розробленню класів моделей, які відповідають за зберігання та оброблення даних. Реалізовані моделі забезпечують цілісність і узгодженість даних. Розглянуто класи представлень, що формують логіку оброблення користувацьких запитів, забезпечують інтерактивну взаємодію користувача з системою. Важливою складовою розробки стала інтеграція елементів ШІ з проєктом Django, що дозволило автоматизувати складні аналітичні процеси.

Детально проаналізовано особливості експлуатації ПЗ, що розроблюється, – "HRFactory" користувачами різних груп повноважень. Інтерфейс системи відзначається структурованістю та адаптивністю.

За підсумками експериментального дослідження встановлено, що запропонований у роботі метод продемонстрував найвищу точність класифікації працівників підприємства за ймовірністю звільнення, забезпечуючи оптимальний баланс між основними метриками оцінки та призводячи до прогнозування плинності кадрів на підприємстві сукупно. Метод характеризується високою стійкістю до шумових даних та здатністю ефективно опрацьовувати значну кількість нелінійних ознак, що підтверджують доцільність застосування ансамблевих методів машинного навчання для прогнозування кадрової політики.

Основні результати, отримані в роботі, представлені в роботах [43], [44], [45].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Процик, І. Плинність кадрів на підприємстві та шляхи її зменшення / І. Процик, Н. Кара // Молодий вчений. – 2020. – № 3 (79). – С. 246-252. – DOI : 10.32839/2304-5809/2020-3-79-52.
2. Цимбалюк, І. Динаміка ринку праці в Україні під час війни: аналіз міграційних тенденцій та показників безробіття / І. Цимбалюк, Н. Павліха, С. Цимбалюк // Інноваційна економіка. – 2023. – № 2. – С. 101-109. – DOI : 10.37332/2309-1533.2023.2.13.
3. Свидрук, І. Вплив штучного інтелекту на мотивацію праці / І. Свидрук, М. Кардаш // Вісник ЛТЕУ. Економічні науки. – 2024. – № 78. – С. 49-54. – DOI : 10.32782/2522-1205-2024-78-07.
4. Bibikova, Y. HR administration in Ukraine and its automation – a must-have for business in 2025 [Electronic resource] / Y. Bibikova, A. Evstifeeva // Accace. – 2025. – Access mode : <https://accace.com/hr-administration-in-ukraine/>.
5. Пристемський, О. Вплив сучасних інформаційних технологій на ефективність управління підприємством / О. Пристемський, А. Пашинний // Актуальні проблеми менеджменту : теоретичні і практичні аспекти: матеріали шостої міжнар. наук.-практ. коф., 28-29 вересня 2023 р. – Одеса : Одеський національний економічний університет, 2023. – С. 306-308.
6. Дриньов, Д. Штучний інтелект в процесі прийняття та реалізації управлінських рішень / Д. Дриньов, К. Войтех, Р. Тимошенко // Таврійський науковий вісник. Серія : Економіка. – 2023. – Вип. 18. – С. 74-79. – DOI : 10.32782/2708-0366/2023.18.7.
7. Danyiuk, I. HRM Systems of Personnel Management / I. Danyiuk, L. Babala, N. Khoma // Economic Analysis. – 2022. – Vol. 32, № 3. – P. 240-246. – DOI : 10.35774/econa2022.03.240.
8. Tsymbaliuk, S. Artificial Intelligence in HR : Practices and Prospects of the Spread in Ukraine / S. Tsymbaliuk, A. Vasylyk, O. Bilyk // The

Eurasia Proceedings of Science Technology Engineering and Mathematics (EPSTEM). – 2022. – Vol. 17. – P. 152-160. – DOI :10.55549/epstem.1176076.

9. BambooHR [Electronic resource]. – Access mode : <https://www.bamboohr.com/>.

10. Manager Self-Service [Electronic resource] // PrismHR. – Access mode : <https://www.prismhr.com/solutions/employee-portal/manager-self-service/>.

11. Employee Self Service (ESS): Everything You Need to Know [Electronic resource] // Planpoint. – Access mode : <https://planpoint.eu/employee-self-service-ess-everything-you-need-to-know/>.

12. Zoho People [Electronic resource]. – Access mode : <https://www.zoho.com/people/>.

13. КАДРОВИК.UA/КАДРИ.UA [Електрон. ресурс]. – Режим доступу : https://www.kadrovik.ua/?srsltid=AfmBOop9gFi2ftgoWesDh3PU44RFLCBL_IHyHRG_DvIcRv_tEb4qHelj.

14. Кравчук, О. Трансформація HR-менеджменту через призму штучного інтелекту : комплексний аналіз імплементації, викликів та етичних аспектів / О. Кравчук, І. Варіс, Н. Каленська // Проблеми сучасних трансформацій. Серія : Економіка та управління. – 2024. – № 15. – DOI : 10.54929/2786-5738-2024-15-04-09.

15. Johnson, R. Human Resource Information Systems : Basics, Applications, and Future Directions / R. Johnson, K. Carlson, M. Kavanagh. – 6th ed. – Thousand Oaks, CA : SAGE Publications, Inc., 2025. – 432 p.

16. Napper, C. People Analytics : Using Data-Driven HR and Gen AI as a Business Asset / C. Napper. – London : Kogan Page Ltd, 2025. – 328 p.

17. Edwards, M. Predictive HR Analytics : Mastering the HR Metric / M. Edwards, K. Edwards, D. Jang. – 3rd ed. – London : Kogan Page Ltd., 2024. – 528 p.

18. Employee/HR Dataset (All in One) [Electronic resource] // KaggleDatasets. – Access mode : <https://www.kaggle.com/datasets/ravindrasinghrana/employeeedataset>.
19. Cook, D. Practical Machine Learning with H2O : Powerful, Scalable Techniques for Deep Learning and AI / D. Cook. – Sebastopol, CA : O'Reilly Media, 2016. – 298 p.
20. Montgomery, D. Introduction to linear regression analysis / D. Montgomery, G. Vinind, E. Peck. – 6th ed. – Hoboken, NJ : Wiley, 2021. – 704 p.
21. Theobad, O. Machine learning and AI for absolute beginners : decision trees / O. Theobad. – Sebastopol, CA : O'Reilly Media, 2024.
22. Salame, N. The decision tree algorithm use in supervised machine learning [Electronic resource] / N. Salame // IJSREM. – Access mode : https://www.researchgate.net/publication/372409344_The_Decision_Tree_Algorithm_Use_in_Supervised_Machine_Learning.
23. Rose, M. Survey of random forest pruning techniques [Electronic resource] / M. Rose, H. Hassen // Computer Science & Information Techology. – 2019. – Vol. 9, № 18. – Access mode : <https://airconline.com/csit/papers/vol9/csit91808.pdf>.
24. Raza, A. Predicting employee attrition using machine learning approaches / A. Raza, K. Munir, F. Younas // Applied Sciences. – 2022. – Vol. 12, № 13. – Article 6424. – DOI : 10.3390/app12136424.
25. Ersozlu, Z. A review of machine iearning methods used for educational data / Z. Ersozlu, S. Taheri, I. Koch // Education and information technologies. – 2024. – Vol. 29 (16). – P. 22125-22145. – DOI : 10.1007/s10639-024-12704-0.
26. Salman, H. Random forest algorithm overview [Electronic resource] / H. Salman, A. Kalakech. A. Steiti // Babylonian Journal of Machine Learning. 2024. – Access mode : <https://journals.mesopotamian.press/index.php/BJML/article/view/417>.

27. Andersen, G. Backend and frontend development technical architecture guid [Electronic resource] / G. Andersen // MoldStud. – 2024. – Access mode : <https://moldstud.com/articles/p-exploring-backend-and-frontend-development-in-technical-architecture>.
28. Munawar, K. The impact of language syntax on the complexity of programs: a case study of Java and Python [Electronic resource] / K. Munawar, S. Naveed // International Journal of Innovations in Science & Techology. – 2022. – Vol. 4, № 3. – Access mode : <https://journal.50sea.com/index.php/IJIST/article/view/339>.
29. Олійник, П. Інформатика та програмування. Частина 4. Об'єктно-орієнтовне програмування. Мова програмування C# [Електрон. ресурс] / П. Олійник. – Київ : КПІ ім. Ігоря Сікорського. – 2024. – Режим доступу : <https://ela.kpi.ua/handle/123456789/72269>.
30. Python Tutorial [Electronic resource] // W3Schools. – Access mode : <https://www.w3schools.com/python/>.
31. Django Tutorials [Electronic resource] // Real Python. – Access mode : <https://realpython.com/tutorials/django/>.
32. Flask [Electronic resource]. – Access mode : <https://flask.palletsprojects.com/en/stable/>.
33. FastAPI [Electronic resource]. – Access mode : <https://fastapi.tiangolo.com/>.
34. Фреймворк Django для створення сайтів [Електрон. ресурс] // Wezom. – Режим доступу : <https://wezom.com.ua/ua/blog/razrabotka-sajtov-na-python-django>.
35. Що таке функціональні вимоги : приклади та шаблони [Електрон. ресурс] // Visure Solutions. – Режим доступу : <https://visuresolutions.com/uk/alm-guide/функціональні-вимоги/>.
36. Каграманова, Ю. Як будувати UML-діаграми. Розбираємо три найпопулярніші варіанти [Електрон. ресурс] / Ю. Каграманова // DOU. – 2022. – Режим доступу : <https://dou.ua/forums/topic/40575/>.

37. MySQL [Electronic resource]. – Access mode : <https://www.mysql.com>.
38. Що таке бази даних NoSQL? [Електрон. ресурс]. – Режим доступу : <https://freehost.com.ua/ukr/faq/wiki/chto-takoe-bazi-dannih-nosql/>.
39. Microsoft SQL Server [Electronic resource]. – Access mode : <https://www.microsoft.com/ru-ru/sql-server>.
40. Grippa, V. Learning MySQL : Get a Handle on Your Data / V. Grippa, S. Kuzmichev. – 2th ed. – Sebastopol (CA) : O'Reilly Media, 2021. – 629 p.
41. DataFrame.to_csv [Electronic resource] // Pandas. – Access mode : https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to_csv.html
42. SelectKBest [Electronic resource] // Scikit-learn. – Access mode : https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html.
43. Горобець В. Дослідження та програмна реалізація методів прогнозування плинності кадрів підприємства / В. Горобець, В. Льовкін // II Міжнародна науково-практична конференція студентів, аспірантів та молодих вчених "Молодіжна наука: інновації та глобальні виклики" : зб. тез. – Полтава : Національний університет «Полтавська політехніка імені Юрія Кондратюка», 2025. – Препринт.
44. Льовкін, В. М. Метод прогнозування групової динаміки людських ресурсів підприємства для системного розв'язання проблеми сертифікації / В. М. Льовкін, В. І. Горобець // Автоматизація технологічних і бізнес-процесів. – 2025. – № 4. – Препринт.
45. Льовкін, В. М. Системне розв'язання проблеми прогнозування групової динаміки людських ресурсів підприємства в умовах сертифікації / В. М. Льовкін, В. І. Горобець // Проблеми використання інформаційних технологій в освіті, науці та промисловості : XIX Міжнародна науково-практична конференція : зб. наук. пр. – Дніпро : Журфонд, 2025. – Препринт.

ДОДАТОК А
Технічне завдання

Вступ

Проблема плинності кадрів є однією з найактуальніших на сучасних підприємствах. Високий рівень звільнень працівників негативно впливає на стабільність роботи організації та її конкурентоспроможність. Часті кадрові зміни призводять до втрати досвіду, зниження продуктивності праці та додаткових витрат на пошук і навчання нових співробітників.

Для зниження плинності кадрів необхідно впроваджувати програму розвитку персоналу, системи наставництва та справедливого стимулювання. Перспективним напрямом є інтеграція програмних систем на основі ШІ, які за визначеними критеріями можуть прогнозувати, чи має намір працівник звільнитися. Такі рішення дають змогу вчасно виявляти ризики звільнення та вживати превентивних заходів для утримання кадрів. Важливо також формувати позитивний імідж роботодавця, який дбає про своїх співробітників. Мінімізація плинності персоналу із використанням сучасних технологій є важливим чинником забезпечення стабільного розвитку підприємства та його довгострокового успіху.

A.1 Підстави для розробки

Дипломна кваліфікаційна робота виконується за темою «Дослідження та програмна реалізація методів прогнозування плинності кадрів підприємства» у відповідності з наказом № 447 від 30 вересня 2025 р. за Національним університетом «Запорізька політехніка».

A.2 Призначення розробки

Програма має використовуватися для дослідження процесів плинності кадрів на підприємстві, а також комплексного аналізу та оптимізації управлінських рішень у сфері кадрової політики, а саме: збір, зберігання та

обробка персональних даних працівників; моніторинг задоволеності персоналу; аналіз динаміки кадрових змін; забезпечення підтримки управлінських рішень; автоматизація кадрового документообігу та аналітика.

A.3 Основні вимоги до програми

A.3.1 Вимоги до функціональних характеристик

Повний перелік функціональних вимог до програми включає:

- авторизація;
- відправка заяви до HR-відділу на отримання облікових даних;
- визначення типу користувача;
- реєстрація нового працівника;
- створення ОСП;
- створення опитувань, щодо якості праці;
- перенесення акаунту звільненого працівника до архіву;
- перегляд ОСП;
- редагування ОСП;
- перегляд заяв працівника;
- перегляд таблицю робочого часу;
- перевірка залишку днів відпустки;
- перегляд доган;
- введення дисциплінарних санкцій;
- створення запрошення на внутрішні навчання;
- створення зустрічей із працівником;
- створення звіту по плинності кадрів в підрозділі/підрозділах;
- налаштування моделі ШІ;
- отримання результату прогнозування;
- інтеграція результатів передбачення;
- відправка службових записок;

- перегляд архіву класифікації працівників підприємства за ймовірністю звільнення;
- перегляд архіву працівників;
- звіт по рівню заробітної плати;
- вихід з облікового запису;
- перегляд запрошень на внутрішні навчання;
- перегляд графіку тренінгів;
- перегляд всіх створених заяв;
- підтвердження участі у тренінгу;
- написання заяв;
- скасування заяв;
- перегляд всіх опитувань;
- участь в опитуваннях;
- перегляд списку зустрічей з HR;
- створення повідомлення про проблему;
- перевірка статусу вирішення проблеми;
- подача заявки на редагування ОСП;
- подання запиту на зміну графіку;
- перегляд історії відпусток;
- отримання повідомлення про важливі події;
- доступ до адмін-панелі;
- тестування моделі ШІ;
- навчання моделі ШІ;
- перевірка логів застосування;
- розділення наборів даних для навчання та тестування моделі ШІ;
- моніторинг некоректних відповідей моделі ШІ;
- трансформація та попередня обробка даних;
- створення резервних копій моделі ШІ;
- відтворення резервних копій моделі ШІ;

- завантаження вибірки;
- перегляд політики підприємства.

Одним із головних принципів роботи проєкту є розділення ролей між HR-менеджером та працівником. Це програмний інструмент, призначений для автоматизації та підтримки управлінських рішень у сфері роботи з персоналом. Система допомагає HR-менеджеру ефективно керувати кадровими процесами, а інтеграція програми зі ШІ дає змогу прогнозувати плінність кадрів, своєчасне виявлення ризиків звільнень та формування рекомендацій щодо мотивації співробітників. Програмний модуль, призначений для працівника, допомагає взаємодіяти з відділом кадрів, управляти власними даними, переглядати особисту інформацію.

А.3.2 Умови експлуатації

Експлуатація ПЗ, призначеного для автоматизації та управління усіма процесами роботи з персоналом, має здійснюватися за умови дотримання низки технічних та програмних вимог до серверної інфраструктури:

- забезпечення вебсерверу: операційна система Linux з останніми оновленнями, актуальна версія мови програмування Python, контейнеризатор Docker, права адміністратора, доступ до мережі інтернет;
- завантаження всіх необхідних бібліотек з файлу requirements.txt;
- налаштування Django-застосунку;
- виконання міграції для створення та підтримання структури БД із використанням MySQL.

А.3.3 Вимоги до складу та параметрів технічних засобів

Для забезпечення коректного функціонування програми необхідне дотримання певних вимог до апаратного забезпечення сервера:

- процесор AMD Ryzen 5 5600x і вище;
- оперативна пам'ять 8 Гб і вище;
- постійна пам'ять 512 Гб і вище.

А.4 Порядок контролю та приймання

Виконання роботи здійснюється відповідно до затвердженого індивідуального завдання та перебуває під систематичним контролем керівника. Результати проведеного дослідження відображаються у пояснювальній записці, що узагальнює основні етапи та здобуті висновки. Завершальним етапом підготовки, контролю та оцінювання магістерської кваліфікаційної роботи є її публічний захист перед екзаменаційною комісією, який підтверджує рівень професійної підготовки здобувача вищої освіти.

ДОДАТОК Б
Текст програми

Б.1 Текст файлу urls.py пакету-застосунку core

```
urlpatterns = [
    path("", include('main.urls')),
    path('machinelearning/', include('ml_predictor.urls')),
    path('documentmanager/', include('document_manager.urls')),
    path('adminpanel/', admin.site.urls),
    path('authenticationprofile/', include('authentication.urls')),
]
```

Б.2 Текст файлу models.py пакету-застосунку authentication

```
class EmployeeProfile(models.Model):
    user = models.OneToOneField(User, on_delete=models.CASCADE)
    first_name = models.CharField(max_length=100)
    last_name = models.CharField(max_length=100)
    start_date = models.DateField()
    exit_date = models.DateField(null=True, blank=True)
    title = models.CharField(max_length=100)
    supervisor = models.CharField(max_length=100)
    email = models.EmailField()
    business_unit = models.CharField(max_length=100)
    employee_status = models.CharField(max_length=50)
    employee_type = models.CharField(max_length=50)
    pay_zone = models.CharField(max_length=50)
    employee_classification_type = models.CharField(max_length=50,
null=True, blank=True)
    termination_type = models.CharField(max_length=50, null=True,
blank=True)
    termination_description = models.TextField(null=True, blank=True)
```

```

department_type = models.CharField(max_length=100)
division_description = models.CharField(max_length=255, null=True,
blank=True)
dob = models.DateField()
state = models.CharField(max_length=100)
job_function = models.CharField(max_length=255, null=True,
blank=True)
EMPLOYEE_GENDER_CHOICES = (
    ('M', 'MALE'),
    ('F', 'FEMALE'),
)
gender = models.CharField(max_length=1,
choices=EMPLOYEE_GENDER_CHOICES)
location = models.CharField(max_length=100)
race_ethnicity = models.CharField(max_length=100, null=True,
blank=True)
marital_status = models.CharField(max_length=50)
performance_score = models.CharField(max_length=50)
current_employee_rating = models.CharField(max_length=50)
def __str__(self):
    return f'{self.first_name} {self.last_name} - {self.title} {self.email}'
class Training(models.Model):
    title = models.CharField(max_length=255, verbose_name='Назва
тренінгу')
    description = models.TextField(blank=True, verbose_name='Опис')
    start_datetime = models.DateTimeField(verbose_name='Дата та час
початку')
    end_datetime = models.DateTimeField(verbose_name='Дата та час
кінця')
```

```

        location      =      models.CharField(max_length=255,      blank=True,
verbose_name='Місце проведення')
        capacity      =      models.PositiveIntegerField(default=0,
verbose_name='Кількість місць')
        is_active     =      models.BooleanField(default=True,
verbose_name='Актуальний')
        created_at = models.DateTimeField(auto_now_add=True)
        updated_at = models.DateTimeField(auto_now=True)
        def __str__(self):
            return      f'{self.title}      {self.is_active}      {self.start_datetime}
{self.end_datetime}'
        def spots_left(self, training_pk):
            training_registration      =
TrainingRegistration.objects.filter(pk=training_pk)
            return max(0, self.capacity - training_registration)
class TrainingRegistration(models.Model):
    user = models.OneToOneField(
        User,
        on_delete=models.CASCADE,
        related_name='training_registration'
    )
    training = models.ForeignKey(
        Training,
        on_delete=models.CASCADE,
        related_name='registration',
    )
    registered_at = models.DateTimeField(auto_now_add=True)
    confirmed      =      models.BooleanField(default=True,
verbose_name='Підтвердження')
class Meta:

```

```

        verbose_name = 'Запис на трейнінг (навчальний курс)'
    def __str__(self):
        return f'{self.user.first_name} {self.user.last_name}
{self.training.title}'

```

Б.3 Текст файлу views.py пакету-застосунку authentication

```

class EmployeeProfileDetailView(LoginRequiredMixin, DetailView):
    model = User
    template_name = 'employee_profile_detail.html'
    context_object_name = 'employee_profile'
class EmployeeProfileUpdateView(
    LoginRequiredMixin,
    UserPassesTestMixin,
    UpdateView,
):
    model = User
    template_name = 'employee_profile_update.html'
    context_object_name = 'employee_profile'
    def test_func(self):
        return self.request.user.groups.filter(name='HR').exists()
    def get_success_url(self):
        return reverse_lazy(
            'employee_profile:employee_profile_detail',
            kwargs={'pk': self.object.pk},
        )
class TrainingListView(LoginRequiredMixin, ListView):
    model = Training
    template_name = 'training_list.html'
    context_object_name = 'training'

```

```
paginate_by = 10
def get_queryset(self):
    return Training.objects.filter(is_active=True).order_by('-created_at')
class TrainingCreateView(
    LoginRequiredMixin,
    UserPassesTestMixin,
    CreateView,
):
    model = Training
    template_name = 'training_create.html'
    fields = [
        'title',
        'description',
        'start_datetime',
        'end_datetime',
        'location',
        'capacity',
        'is_active',
    ]
    def test_func(self):
        return self.request.user.groups.filter(name='HR').exists()
    def get_success_url(self):
        return reverse_lazy(
            'employee_profile:employee_profile_list',
            kwargs={'pk': self.object.pk},
        )
class TrainingRegistrationCreateView(
    LoginRequiredMixin,
    CreateView,
):
```

```

model = TrainingRegistration
template_name = 'training_registration_create.html'
fields = [
    'training',
    'confirmed',
]
def form_valid(self, form):
    form.instance.user = self.request.user
    return super().form_valid(form)
def get_success_url(self):
    return reverse_lazy(
        'employee_profile:training_list',
        kwargs={'pk': self.object.pk},
    )
class TrainingRegistrationListView(LoginRequiredMixin, ListView):
    model = TrainingRegistration
    template_name = 'training_registration_list.html'
    context_object_name = 'training_registration'
    paginate_by = 10
    def get_queryset(self):
        queryset = super().get_queryset()
        if self.request.user.groups.filter(name='HR'):
            return queryset
        else:
            return queryset.filter(user=self.request.user)

```

Б.4 Текст файлу urls.py пакету-застосунку authentication

```

app_name = 'employee_profile'
urlpatterns = [

```

```
path(
    'profile/<int:pk>',
    EmployeeProfileDetailView.as_view(),
    name='employee_profile_detail',
),
path(
    'profile/update/<int:pk>',
    EmployeeProfileUpdateView.as_view(),
    name='employee_profile_update',
),
path(
    'training/',
    TrainingListView.as_view(),
    name='employee_training_list',
),
path(
    'training/create/',
    TrainingCreateView.as_view(),
    name='employee_training_create',
),
path(
    'trainingregistration/create/',
    TrainingRegistrationCreateView.as_view(),
    name='employee_trainingregistration_create',
),
path(
    'trainingregistration',
    TrainingRegistrationListView.as_view(),
    name='employee_trainingregistration_list',
),
```

]

Б.5 Текст файлу models.py пакету-застосунку document_manager

```

class Document(models.Model):
    user = models.ForeignKey(
        settings.AUTH_USER_MODEL,
        on_delete=models.CASCADE,
        related_name='documents',
    )
    title = models.CharField(max_length=255)
    description = models.TextField(blank=True)
    scan_file = models.FileField()
    uploaded_at = models.DateTimeField(auto_now_add=True)
    def __str__(self):
        return f'Документ: {self.title} -> {self.uploaded_at}'

class Application(models.Model):
    user = models.ForeignKey(
        settings.AUTH_USER_MODEL,
        on_delete=models.CASCADE,
        related_name='applications',
    )
    subject = models.CharField(max_length=255, verbose_name='Тема')
    body = models.TextField(verbose_name='Докладно')
    category = models.CharField(max_length=50,
        verbose_name='Категорія')
    status = models.CharField(max_length=50, verbose_name='Статус')
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
    def __str__(self):

```

```
return f'Заява: {self.subject} -> {self.category} -> {self.status}'
```

Б.6 Текст файлу views.py пакету-застосунку document_manager

```
from django.contrib.auth.mixins import LoginRequiredMixin,
UserPassesTestMixin

from django.views.generic import DetailView, UpdateView, ListView,
CreateView, DeleteView

from django.contrib.auth.models import User

from django.urls import reverse_lazy

class DocumentCreateView(
    LoginRequiredMixin,
    CreateView,
):
    model = Document
    template_name = 'document_manager_create.html'
    fields = [
        'title',
        'description',
        'scan_file',
    ]
    def form_valid(self, form):
        form.instance.user = self.request.user
        return super().form_valid(form)
    def get_success_url(self):
        return reverse_lazy('document_manager:document_create')

class DocumentDeleteView(
    LoginRequiredMixin,
    DeleteView,
):
```

```

model = Document
template_name = 'document_manager_delete.html'
context_object_name = 'document_manager'
def get_queryset(self):
    return super().get_queryset().filter(user=self.request.user)
def get_success_url(self):
    return reverse_lazy('document_manager:document_delete')
class DocumentListView(
    LoginRequiredMixin,
    ListView,
):
    model = Document
    template_name = 'document_manager_list.html'
    context_object_name = 'document_manager'
    paginate_by = 10
    def get_queryset(self):
        if self.request.user.groups.filter(name='HR').exists():
            return super().get_queryset()
        else:
            return super().get_queryset().filter(user=self.request.user)
class ApplicationCreateView(
    LoginRequiredMixin,
    CreateView,
):
    model = Application
    fields = [
        'subject',
        'body',
        'category',
        'status'

```

```

]
def form_valid(self, form):
    form.instance.user = self.request.user
    return super().form_valid(form)
def get_success_url(self):
    return reverse_lazy('document_manager:application_create')
class ApplicationDetailView(
    LoginRequiredMixin,
    DetailView,
):
    model = Application
    template_name = 'application_detail.html'
    context_object_name = 'application'
    def get_queryset(self):
        if self.request.user.groups.filter(name='HR').exists():
            return super().get_queryset()
        else:
            return super().get_queryset().filter(user=self.request.user)

```

Б.7 Текст файлу models.py пакету-застосунку ml_predictor

```

class EmployeePrediction(models.Model)
    user = models.ForeignKey(
        User,
        on_delete=models.CASCADE,
        related_name='predictions',
    )
    will_leave = models.BooleanField(help_text='Можливість звільнення')
    model_version = models.CharField(max_length=50)
    def __str__(self):

```

```

        return f'Передбачення звільнення працівника {self.user.first_name}
        {self.user.last_name} -> {self.will_leave}'

```

Б.8 Текст файлу predictions.py пакету-застосунку ml_predictor

```

import pandas as pd
import numpy as np
import matplotlib.pyplot as random_plt
from sklearn.model_selection import (
    train_test_split,
    GridSearchCV,
)
from sklearn.ensemble import (
    RandomForestClassifier,
    RandomForestRegressor,
)
from sklearn.metrics import (
    accuracy_score,
    classification_report,
    confusion_matrix,
)
from sklearn.preprocessing import StandardScaler
class EmployeeTurnoverModel:
    def __init__(
        self,
        n_estimators=100,
        random_state=42,
        scaler=StandardScaler()
    ):
        self._model = RandomForestClassifier(

```

```

        n_estimators=n_estimators,
        random_state=random_state,
    )
    self._scaler = scaler
    self._is_trained = False
    self._feature_importances = None
def preprocess(self, input_arr_data):
    tmp_arr_data_scaled = input_arr_data.copy()
    tmp_arr_data_scaled[:] = self._scaler.fit_transform(
        tmp_arr_data_scaled[:]
    )
    return tmp_arr_data_scaled
def train(
    self,
    input_x: np.ndarray,
    input_y: np.ndarray ,
    test_arr_data=0.2,
    perform_feature_selection=False,
    top_n_features=None
):
    self._x_train, self._x_test, self._y_train, self._y_test = train_test_split(
        input_x, input_y, test_size=test_arr_data, stratify=y
    )
    self._x_train = self.preprocess(self._x_train)
    self._x_test = self.preprocess(self._x_test)
    self._model.fit(self._x_train, self._y_train)
    self._perform_feature_selection = perform_feature_selection
    self._top_n_features = top_n_features
    self._is_trained = True
def predict(self, x_input_data):

```

```

    if not self._is_trained:
        raise ValueError('Модель Random Forest ненавчена')
    tmp_num_col =
x_input_data.select_dtypes(include=[np.number]).columns
    tmp_num_col = self.preprocess(tmp_num_col)
    if self._feature_importances is not None:
        tmp_main_feature_importances =
self._feature_importances['feature'].tolist()
        tmp_num_col = tmp_num_col[tmp_main_feature_importances]
    return self._model(tmp_num_col)

def evaluate(self):
    if not self._is_trained:
        raise ValueError('Модель Random Forest ненавчена')
    tmp_y_pred = self.predict(self._x_test)
    tmp_acc = accuracy_score(self._y_test, tmp_y_pred)
    tmp_report = classification_report(self._y_test, tmp_y_pred)
    return (
        tmp_acc,
        tmp_report,
        confusion_matrix(self._y_test, tmp_y_pred),
    )

```

Б.9 Текст файла views.py пакета-застосунку ml_predictor

```

class EmployeePredictionListView(
    LoginRequiredMixin,
    UserPassesTestMixin,
    ListView,
):
    model = EmployeePrediction

```

```

template_name = 'employeePrediction_list.html'
context_object_name = 'employee_prediction'
paginate_by = 10
def test_func(self):
    my_result_prediction =
self.request.user.groups.filter(name='HR').exists()
    return my_result_prediction
class EmployeePredictionDeleteView(
    LoginRequiredMixin,
    UserPassesTestMixin,
    DeleteView,
):
    model = EmployeePrediction
    template_name = 'employeePrediction_delete.html'
    context_object_name = 'employee_prediction'
    def test_func(self):
        my_result_prediction =
self.request.user.groups.filter(name='HR').exists()
        return my_result_prediction
    def get_success_url(self):
        return reverse_lazy('employeePrediction_list')
class EmployeePredictionCreateView(
    LoginRequiredMixin,
    UserPassesTestMixin,
    CreateView,
):
    model = EmployeePrediction
    template_name = 'employeePrediction_create.html'
    context_object_name = 'employee_prediction'
    fields = [

```

```

        'user',
        'will_leave',
        'model_version',
    ]
    def test_func(self):
        my_result_prediction =
self.request.user.groups.filter(name='HR').exists()
        return my_result_prediction
    def get_success_url(self):
        return reverse_lazy('employeeprediction_list')
class EmployeePredictionUpdateView(
    LoginRequiredMixin,
    UserPassesTestMixin,
    UpdateView,
):
    model = EmployeePrediction
    template_name = 'employeePrediction_update.html'
    context_object_name = 'employee_prediction'
    fields = [
        'user',
        'will_leave',
        'model_version',
    ]
    def test_func(self):
        my_result_prediction =
self.request.user.groups.filter(name='HR').exists()
        return my_result_prediction

```

Б.10 Текст файлу urls.py пакету-застосунку ml_predictor

```
app_name = 'employeeprediction'
urlpatterns = [
    path(
        "",
        EmployeePredictionDeleteView.as_view(),
        name='employeePrediction_delete',
    ),
    path(
        "",
        EmployeePredictionListView.as_view(),
        name='employeePrediction_list',
    ),
    path(
        "",
        EmployeePredictionCreateView.as_view(),
        name='employeePrediction_create',
    ),
    path(
        "",
        EmployeePredictionUpdateView.as_view(),
        name='employeePrediction_create',
    ),
]
```

Б.11 Текст файлу models.py пакету-застосунку main

```
class Poll(models.Model):
    title = models.CharField(
```

```

        max_length=255,
        verbose_name='Назва опитування для працівників',
    )
description = models.TextField(
    blank=True,
    verbose_name='Докладний опис опитування для працівників',
)
created_at = models.DateTimeField(auto_now_add=True)
is_active = models.BooleanField(default=True)
class Meta:
    verbose_name = 'Опитування для працівника підприємства'
def __str__(self):
    return f'Опитування {self.title} -> {self.created_at} ->
{self.is_active}'
class Question(model.Models):
    QUESTION_ENUM = [
        ('text', 'Текст відповіді працівника'),
        ('single', 'Один варіант відповіді працівника'),
        ('multiple', 'Декілька відповідей працівника')
    ]
    poll = models.ForeignKey(
        Poll,
        on_delete=models.CASCADE,
        related_name='questions',
    )
    text = models.CharField(
        max_length=500,
        verbose_name='Текст питання для працівника',
    )
    question_type = models.CharField(

```

```

        max_length=60,
        choices=QUESTION_ENUM,
        default=QUESTION_ENUM[0],
    )
    def __str__(self):
        return f'Питання: {self.type} -> {self.question_type}'
class Issue(models):
    user = models.ForeignKey(
        User,
        on_delete=models.SET_NULL,
        null=True,
        blank=True
    )
    title = models.CharField(
        max_length=255,
        verbose_name='Короткі відомості про проблему',
    )
    description = models.TextField(
        verbose_name='Докладний опис проблеми працівником'
    )
    category = models.CharField(
        max_length=20,
        choices=IssueCategory.choices,
        default=IssueCategory.OTHER_ISSUE,
        verbose_name='Статус проблеми',
    )
    status = models.CharField(
        max_length=20,
        choices=IssueStatus.choices,
        default=IssueStatus.NEW_PROBLEM,

```

```

        verbose_name='Статус проблеми',
    )
    created_at = models.DateTimeField(
        auto_now_add=True,
        verbose_name='Створено запит на вирішення проблеми',
    )
    updated_at = models.DateTimeField(
        auto_now_add=True,
        verbose_name='Оновлення запиту на вирішення проблеми',
    )
    class Meta:
        verbose_name = 'Проблема працівника підприємства'
        verbose_name_plural = 'Проблеми працівника підприємства'
    def __str__(self):
        return f'Проблема: {self.title} -> {self.category} -> {self.status} ->
{self.created_at}'
    class DisciplinaryAction(models.Model):
        employee = models.ForeignKey(
            User,
            on_delete=models.CASCADE,
            related_name='disciplinary_action',
        )
        reason = models.TextField(
            verbose_name= 'Докладний опис причини санкції для працівника'
        )
        issued_by = models.ForeignKey(
            User,
            on_delete=models.SET_NULL,
            null=True,
            blank=True,

```

```

        verbose_name='Автор санкції для працівника',
    )
    date_issued = models.DateField(
        null=True,
        blank=True,
        verbose_name='Дата ознайомлення працівника зі санкцією',
    )
    expiration_date = models.DateField(
        null=True,
        blank=True,
        verbose_name='Дата закінчення дії санкції працівника',
    )
    is_active = models.BooleanField(
        default=True,
        verbose_name='Активна догана чи ні?'
    )
    class Meta:
        verbose_name = 'Санкція для працівника підприємства'
        verbose_name_plural = 'Санкції для працівника підприємства'
    def __str__(self):
        return f'Санкція {self.reason} -> {self.date_issued}'
class HRMeeting(models.Model):
    disciplinary_action = models.ForeignKey(
        DisciplinaryAction,
        on_delete=models.CASCADE,
        related_name='hr_meetings',
    )
    date = models.DateTimeField(
        verbose_name='Дата та час зустрічі працівника з HR-менеджером',
    )

```

```

location = models.CharField(
    max_length=255,
    verbose_name='Місце проведення зустрічі',
)
notes = models.TextField(
    verbose_name='Нотатки HR-менеджера перед початком зустрічі',
    blank=True,
)
attended = models.BooleanField(
    default=False,
    verbose_name='Чи з\'явився працівник на зустріч з HR-
менеджером',
)
def __str__(self):
    return f'Зустріч з HR-менеджером {self.date} -> {self.attended}'
class CompanyPolicy(models.Model):
    title = models.CharField(
        max_length=255,
        verbose_name='Назва СТП підприємства',
    )
    category = models.CharField(
        max_length=50,
        choices=CompanyPolicyCategory.choices,
        default=CompanyPolicyCategory.FINANCE,
        verbose_name='Категорія СТП підприємства',
    )
    content = models.TextField(verbose_name='Докладний зміст СТП
підприємства')
    document = models.FileField(
        upload_to='company_policies/',

```

```

        blank=True,
        null=True,
        verbose_name='Файл з СТП підприємства',
    )
    created_by = models.ForeignKey(
        User,
        on_delete=models.SET_NULL,
        null=True,
        blank=True,
        verbose_name='Відповідальний за створення СТП підприємства',
    )
    created_at = models.DateTimeField(
        auto_now_add=True,
        verbose_name='Створено СТП підприємства',
    )
    updated_at = models.DateTimeField(
        auto_now_add=True,
        verbose_name='Оновлення СТП підприємства',
    )
    is_active = models.BooleanField(
        default=True,
        verbose_name='Активне СТП чи ні?'
    )
    class Meta:
        verbose_name = 'СТП підприємства'
    def __str__(self):
        return f'СТП {self.title} -> {self.created_at} -> {self.is_active}'

```

Б.12 Текст файлу views.py пакету-застосунку main

```
class PollCreateView(
    LoginRequiredMixin,
    UserPassesTestMixin, CreateView,
):
    model = Poll
    fields = [
        'title',
        'description',
        'is_active',
    ]
    template_name = 'poll_create.html'
    def test_func(self):
        return self.request.user.groups.filter(name='HR').exists()
    def form_valid(self, form):
        form.instance.user = self.request.user
        return super().form_valid(form)
    def get_success_url(self):
        return reverse('poll_detail', kwargs={'pk': self.object.pk})

class QuestionCreateView(
    LoginRequiredMixin,
    UserPassesTestMixin,
    CreateView,
):
    model = Question
    fields = [
        'poll',
        'text',
        'question_type',
```

```

]
template_name = 'question_create.html'
def test_func(self):
    return self.request.user.groups.filter(name='HR').exists()
def form_valid(self, form):
    form.instance.user = self.request.user
    form.instance.poll = self.poll
    return super().form_valid(form)
def dispatch(self, request, *args, **kwargs):
    self.poll = get_object_or_404(Poll, pk=self.kwargs['poll_id'])
    return super().dispatch(request, *args, **kwargs)
def get_success_url(self):
    return reverse('poll_detail', kwargs={'pk': self.object.pk})
class AnswerCreateView(
    LoginRequiredMixin,
    UserPassesTestMixin,
    CreateView,
):
    model = Answer
    template_name = 'answer_create.html'
    def test_func(self):
        return self.request.user.groups.filter(name='HR').exists()
    def form_valid(self, form):
        form.instance.user = self.request.user
        form.instance.poll = self.poll
        form.instance.poll = self.poll
        return super().form_valid(form)
    def dispatch(self, request, *args, **kwargs):
        self.poll = get_object_or_404(Poll, pk=self.kwargs['poll_id'])

```

```

        self.question = get_object_or_404(Question,
pk=self.kwargs['poll_id'])
        return super().dispatch(request, *args, **kwargs)
    def get_success_url(self):
        return reverse('poll_detail', kwargs={'pk': self.object.pk})

```

Б.13 Текст файлу admin.py пакету-застосунку ml_predictor

```

@admin.register(EmployeePrediction)
class EmployeePredictionAdmin(admin.ModelAdmin):
    list_display = (
        'user',
        'prediction_date',
        'will_leave',
        'probability',
        'model_version',
    )
    list_filter = (
        'will_leave',
        'model_version',
        'prediction_date',
    )
    search_fields = (
        'user__username',
        'user__first_name',
        'user__last_name',
        'user__title',
        'user__pay_zone',
        'user__employee',
    )

```

```

readonly_fields = (
    'prediction_date',
)
fieldsets = (
    ('Інформація класифікації працівників за ймовірністю звільнення',
    {
        'fields': (
            'user',
            'prediction_date',
            'model_version',
        )
    }),
    ('Результати класифікації працівників за ймовірністю звільнення',
    {
        'fields': (
            'will_leave',
            'probability',
        )
    })
)

```

Б.14 Текст файлу `admin.py` пакету-застосунку `document_manager`

```

@admin.register(Document)
class DocumentAdmin(admin.ModelAdmin):
    list_display = (
        'user',
        'title',
        'uploaded_at',
        'description',

```

```
        'scan_file',
    )
    list_filter = (
        'uploaded_at',
    )
    search_fields = (
        'user__username',
        'user__first_name',
        'user__last_name',
        'title',
        'description',
    )
    readonly_fields = (
        'uploaded_at',
    )
    fieldsets = (
        ('Інформація про завантажену скан-копію документа', {
            'fields': (
                'user',
                'title',
                'description',
            )
        }),
        ('Завантажена скан-копія документа', {
            'fields': (
                'scan_file',
                'uploaded_at',
            )
        })
    )
)
```

```

@admin.register(ApplicationAttachment)
class ApplicationAttachmentAdmin(admin.ModelAdmin):
    list_display = (
        'id',
        'application',
        'file',
        'uploaded_at',
    )
    list_filter = (
        'uploaded_at',
    )
    search_fields = (
        'application__subject',
        'application__category',
        'application__status',
    )
    readonly_fields = (
        'uploaded_at',
    )
    fieldsets = (
        ('Інформація про завантажену скан-копію документа до заяви
працівника', {
            'fields': (
                'application',
            )
        }),
        ('Завантажена скан-копія документа до заяви працівника', {
            'fields': (
                'file',
                'uploaded_at',))
    ))

```

ДОДАТОК В
Слайди презентації

Національний університет «Запорізька політехніка»
Кафедра програмних засобів

Дипломна кваліфікаційна робота магістра

Дослідження та програмна реалізація методів прогнозування плинності кадрів підприємства

Research and Software Implementation of Methods for Forecasting Employee Turnover in an Enterprise

Виконав
студент групи КНТ-214м
Владислав ГОРОБЕЦЬ

Керівник роботи
к.т.н., доцент
Валерій ЛЬОВКІН

2025

Рисунок В.1 – Слайд № 1

2

Метою кваліфікаційної роботи є зменшення витрат на управління персоналом підприємства шляхом розробки програмних засобів прогнозування плинності кадрів підприємства.

Об'єкт дослідження – процес обчислення плинності кадрів підприємства.

Для досягнення поставленої мети в даній роботі визначено низку наступних завдань:

- виконання аналізу предметної області;
- розробка методу класифікації працівників щодо ймовірності їх звільнення для планування кадрової політики підприємства;
- проектування архітектури ПЗ;
- реалізація ПЗ;
- виконання експериментального дослідження класифікації працівників щодо ймовірності їх звільнення.

Рисунок В.2 – Слайд № 2

Порівняння програмних аналогів 3

Критерій	BambooHR	Zoho People	КАДРОВИК.UA/ КАДРИ.UA	HRFactory
Категорія застосунку	HR-система	HR-система	HR-система	HR-система
Зручний інтерфейс	+	-	+	+
Множина базових функцій	+	-	-	+
Кастомізація	-	-	-	+
Масштабованість	-	+	-	+
Використання ШІ для прогнозування плинності кадрів	-	-	-	+
Безкоштовний сервіс	-	-	-	+
Наявність MSS	+	+	-	+
Наявність ESS	+	+	-	+
Орієнтованість на український ринок	-	+	+	+

Рисунок В.3 – Слайд № 3

Порівняння методів машинного навчання 4

Критерій	Linear Regression	Decision Tree	Random Forest
Тип задачі	регресія	класифікація, регресія	класифікація, регресія
Тип моделі	лінійна регресія	одиначне дерево рішень	ансамблевий метод (комбінація дерев рішень)
Точність передбачень	-	-	+
Стійкість до перенавчання	-	-	+
Підтримка нелінійних зв'язків	-	+	+
Обробка шумів/аномалій	-	-	+
Робота з великою кількістю змінних	-	-	+
Складність налаштування	+	+	-

Рисунок В.4 – Слайд № 4

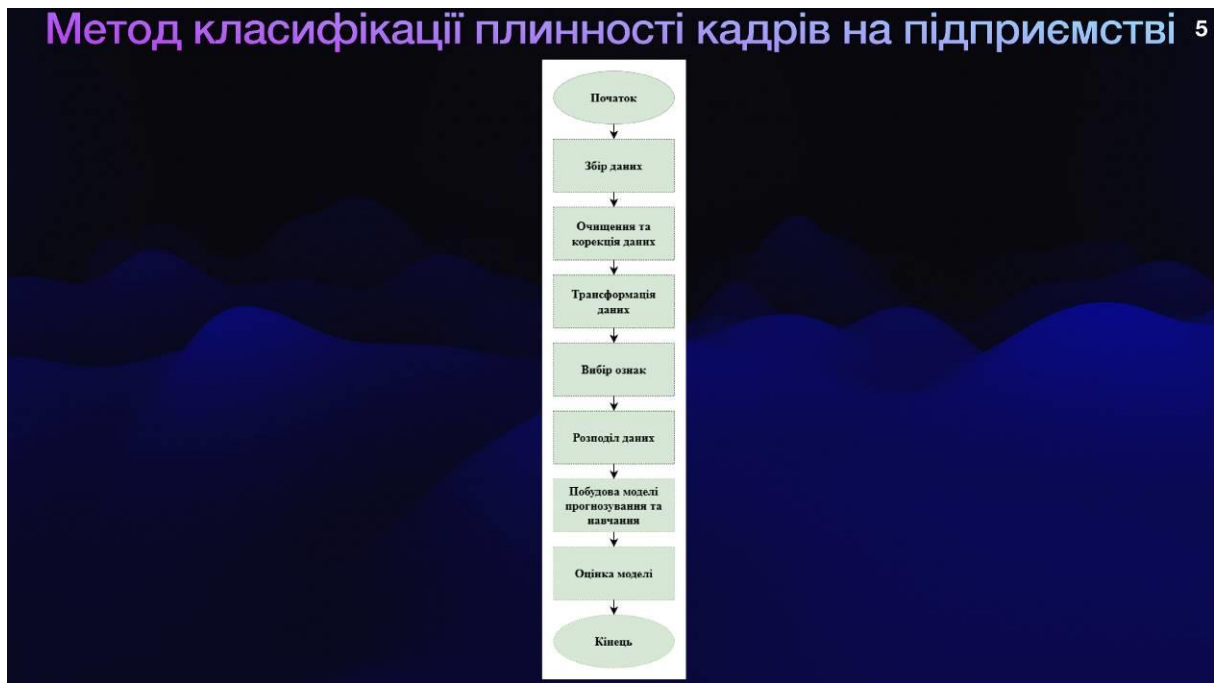


Рисунок В.5 – Слайд № 5

Порівняння мов програмування для створення ПЗ 6

Критерій	Python	Java	C#
Простота синтаксису та зрозумілість коду	+	-	-
Швидкість розробки backend-застосунків	+	-	-
Екосистема бібліотек для AI/ML	+	-	-
Продуктивність у високонавантажених системах	-	+	+
Розмір спільноти та підтримка	+	+	-
Масштабованість для великих проєктів	+	+	+
Крос-платформність	+	+	+
Збирач сміття	+	+	+
Підтримка ООП	+	+	+
Підтримка функціонального програмування	+	-	-
Інструменти налагодження	-	+	+
Різноманітність фреймворків для створення веб-застосунків	+	-	-

Рисунок В.6 – Слайд № 6

Порівняння СКБД

Критерій	NoSQL	MySQL	Microsoft SQL Server
Тип БД	Документа	Реляційна	Реляційна
Безкоштовне використання	-	+	-
Підтримка SQL-запитів	-	+	+
Інструменти адміністрування	-	+	-
Кросплатформеність	+	+	-
Підтримка спільноти	+	+	-
Відкритий вихідний код	-	+	-
Підтримка транзакцій	-	+	+
Підтримка індексів	+	+	+
Простота встановлення	+	+	-
Безпека	-	+	+
Підтримка JOIN-запитів	-	+	+
Низькі вимоги для ресурсів комп'ютера	+	+	-
Поширеність у веб-розробці	-	+	-

Рисунок В.9 – Слайд № 9

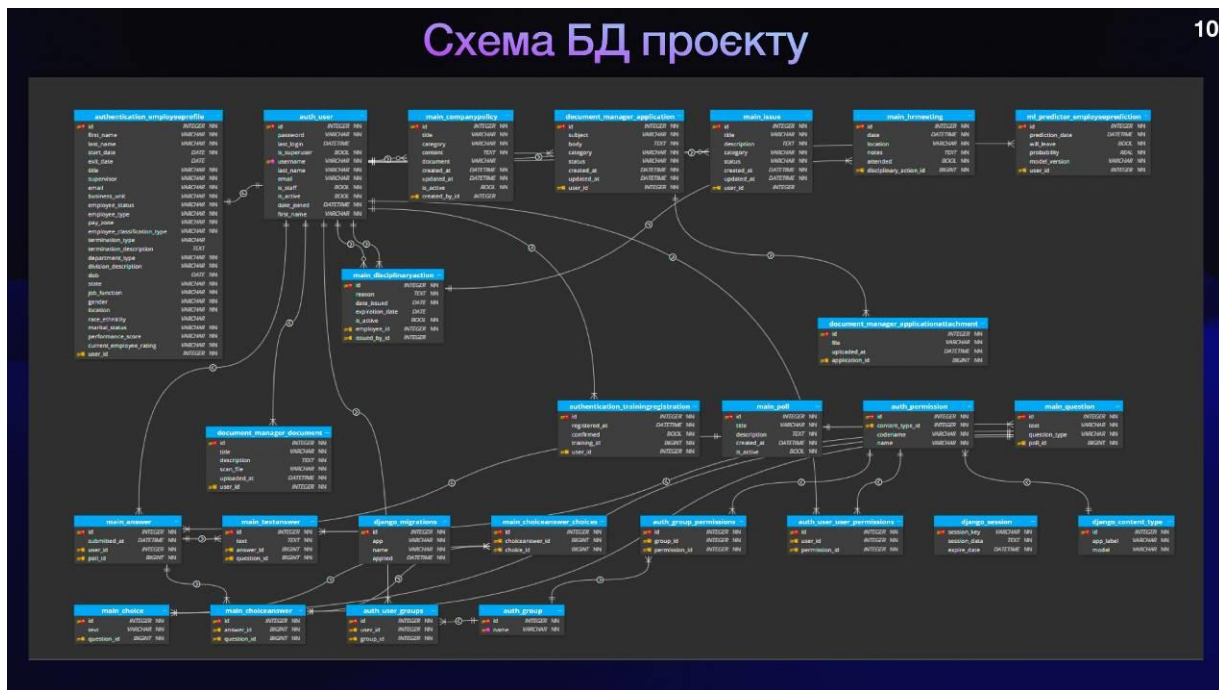


Рисунок В.10 – Слайд № 10



Рисунок В.11 – Слайд № 11

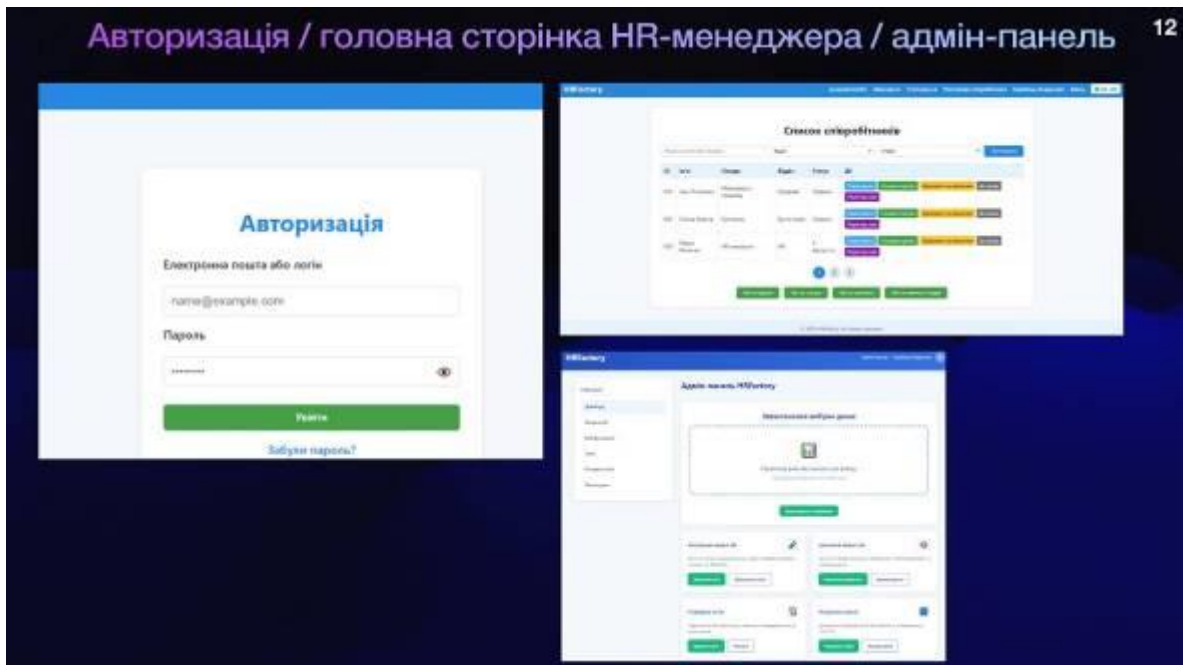



Рисунок В.12 – Слайд № 12

Особистий кабінет працівника / створення електронної заяви

Особистий кабінет працівника



Горобець Владислав
 Посада: IT спеціаліст
 Відділ: Відділ інформаційних технологій
 Електронна пошта: vlad.mfactory@maaprlife.com
 Телефон: +38051234567

Робоча інформація

Показник	Значення
Посада	IT спеціаліст
Відділ	Інформаційні технології
Стаж	3 роки
Заробітна плата	30 000 грн
Тип зайнятості	Повна

Знання мов

Мова	Рівень
Українська	Рідна
Англійська	B2
Польська	A2

Освіта

Задача	Ступінь	Рік закінчення
КНУ ім. Т. Шевченка	Магістр комп'ютерних наук	2022

Останні звернення

Дата	Тип	Статус
12.03.2025	Заява на відпустку	Создано
02.02.2025	Технічна підтримка	В процесі

[Розглянути профіль](#)
[Повна історія](#)
[Вийти](#)

Електронна заява

ПІБ співробітника

Горобець Владислав

Відділ

IT

Категорія

Зсув робочої зміни

Дата

ДД.ММ.РРРР

Час початку зміни

--:--

Час закінчення зміни

--:--

Причина

Надіслати заяву

Рисунок В.13 – Слайд № 13

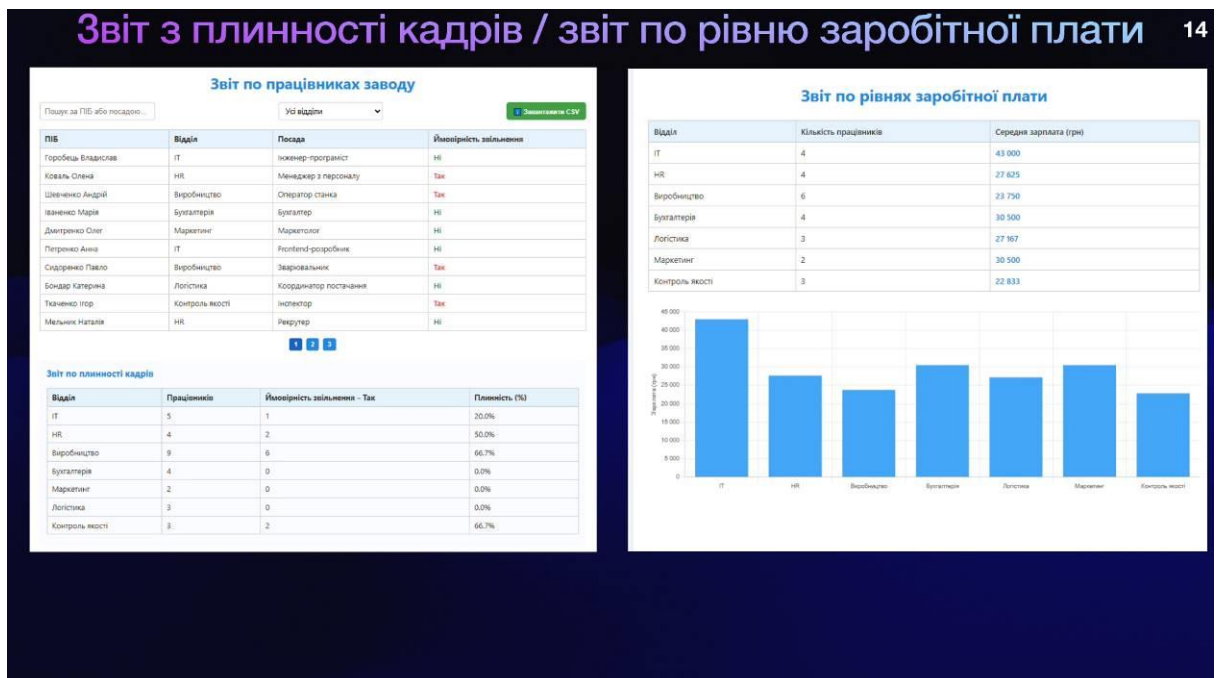




Рисунок В.14 – Слайд № 14

Архів звільнених працівників / реєстрація нового працівника



Архів звільнених працівників

№	Ім'я	Піде	Дата звільнення	Поводження
1	Іванов Іван	123456789	2023-10-25	Звільнення за власним бажанням
2	Петров Петро	987654321	2023-10-25	Звільнення за власним бажанням
3	Сидоров Сидор	456789123	2023-10-25	Звільнення за власним бажанням
4	Коваленко Коваленко	321654987	2023-10-25	Звільнення за власним бажанням
5	Левченко Левченко	210987654	2023-10-25	Звільнення за власним бажанням
6	Мельник Мельник	109876543	2023-10-25	Звільнення за власним бажанням
7	Попов Попов	098765432	2023-10-25	Звільнення за власним бажанням
8	Романенко Романенко	987654321	2023-10-25	Звільнення за власним бажанням
9	Савченко Савченко	876543210	2023-10-25	Звільнення за власним бажанням
10	Тарасенко Тарасенко	765432109	2023-10-25	Звільнення за власним бажанням
11	Васильченко Васильченко	654321098	2023-10-25	Звільнення за власним бажанням
12	Григоренко Григоренко	543210987	2023-10-25	Звільнення за власним бажанням
13	Даниленко Даниленко	432109876	2023-10-25	Звільнення за власним бажанням
14	Жукович Жукович	321098765	2023-10-25	Звільнення за власним бажанням
15	Земляк Земляк	210987654	2023-10-25	Звільнення за власним бажанням
16	Іванов Іван	109876543	2023-10-25	Звільнення за власним бажанням
17	Коваленко Коваленко	098765432	2023-10-25	Звільнення за власним бажанням
18	Левченко Левченко	987654321	2023-10-25	Звільнення за власним бажанням
19	Мельник Мельник	876543210	2023-10-25	Звільнення за власним бажанням
20	Попов Попов	765432109	2023-10-25	Звільнення за власним бажанням
21	Романенко Романенко	654321098	2023-10-25	Звільнення за власним бажанням
22	Савченко Савченко	543210987	2023-10-25	Звільнення за власним бажанням
23	Тарасенко Тарасенко	432109876	2023-10-25	Звільнення за власним бажанням
24	Васильченко Васильченко	321098765	2023-10-25	Звільнення за власним бажанням
25	Григоренко Григоренко	210987654	2023-10-25	Звільнення за власним бажанням
26	Даниленко Даниленко	109876543	2023-10-25	Звільнення за власним бажанням
27	Жукович Жукович	098765432	2023-10-25	Звільнення за власним бажанням
28	Земляк Земляк	987654321	2023-10-25	Звільнення за власним бажанням
29	Іванов Іван	876543210	2023-10-25	Звільнення за власним бажанням
30	Коваленко Коваленко	765432109	2023-10-25	Звільнення за власним бажанням



Реєстрація нового працівника

Формуляція для реєстрації нового працівника з полями для введення персональних даних, контактною інформацією та налаштуваннями.

Рисунок В.15 – Слайд № 15

Експериментальне дослідження

Метод / модель	Accuracy	Precision	Recall	F1-score
Запропонований метод класифікації працівників підприємства за ймовірністю звільнення (Random Forest)	0,8966	0,8732	0,8957	0,8794
Decision Tree	0,7868	0,7515	0,8114	0,7803
Logistic Regression	0,8100	0,7845	0,8185	0,8011

Рисунок В.16 – Слайд № 16

Висновки

17

У процесі розробки програмного забезпечення HRFactory в рамках дипломної кваліфікаційної роботи було виконано низку завдань, спрямованих на досягнення поставленої мети дослідження.

Виконано аналіз предметної області та порівняння сучасних HR-систем, таких як BambooHR, Zoho People і КАДРОВИК.UA, що виявило функціональні обмеження застосунків.

Для прогнозування плинності кадрів використано набір даних Employee/HR Dataset, проведено його очищення, трансформацію та розподіл на навчальні й тестові вибірки. Здійснено порівняння моделей машинного навчання, серед яких Linear Regression, Decision Tree та Random Forest, з визначенням останньої як оптимальної для задачі класифікації працівників підприємства за ймовірністю звільнення.

Розроблено функціональні моделі, класи представлень та структуру бази даних MySQL, що забезпечують цілісність, узгодженість і гнучкість системи. Інтеграція з технологіями ШІ, що дозволило автоматизувати аналітичні процеси та підвищити точність прогнозів. Інтерфейс HRFactory відзначається структурованістю, адаптивністю та зручністю для користувачів різних груп повноважень.

Експериментальне дослідження підтвердило високу ефективність моделі Random Forest у прогнозуванні відтоку кадрів. Модель демонструє стійкість до шумових даних та здатність обробляти значну кількість нелінійних ознак. Результати дослідження підтверджують доцільність застосування ансамблевих методів машинного навчання та сучасних технологій для підтримки управління персоналом.

Наукова новизна роботи полягає в методі класифікації працівників підприємства за ймовірністю звільнення, що характеризується використанням моделі, побудованої методом Random Forest, виконанням автоматичного відбору ознак для побудови моделі, що дозволяє, сукупно за підприємством, виконувати прогнозування плинності кадрів.

Рисунок В.17 – Слайд № 17

Перелік публікацій

18

1. Горобець В. Дослідження та програмна реалізація методів прогнозування плинності кадрів підприємства / В. Горобець, В. Льовкін // II Міжнародна науково-практична конференція студентів, аспірантів та молодих вчених "Молодіжна наука: інновації та глобальні виклики" : зб. тез. – Полтава : Національний університет «Полтавська політехніка імені Юрія Кондратюка», 2025. – Препринт.
2. Льовкін, В. М. Метод прогнозування групової динаміки людських ресурсів підприємства для системного розв'язання проблеми сертифікації / В. М. Льовкін, В. І. Горобець // Автоматизація технологічних і бізнес-процесів. – 2025. – № 4. – Препринт.
3. Льовкін, В. М. Системне розв'язання проблеми прогнозування групової динаміки людських ресурсів підприємства в умовах сертифікації / В. М. Льовкін, В. І. Горобець // Проблеми використання інформаційних технологій в освіті, науці та промисловості : XIX Міжнародна науково-практична конференція : зб. наук. пр. – Дніпро : Журфонд, 2025. – Препринт.
4. Горобець, В.І. Програмна система пошуку та публікації вакансій для працевлаштування / В.І. Горобець, В. М. Льовкін // 27-й Міжнародний молодіжний форум «Радіоелектроніка та молодь у XXI столітті». Зб. матеріалів форуму. Т. 6, Ч. I. – Харків : ХНУРЕ, 2023. – С. 407-408.
5. Льовкін, В.М. Програмний компонент з тестування в системі пошуку вакансій для працевлаштування / В.М. Льовкін, В.І. Горобець // Тиждень науки-2023. Факультет комп'ютерних наук і технологій. Тези доповідей науково-технічної конференції, Запоріжжя, 24-28 квітня 2023 р. [Електронний ресурс] / Редкол. : Вадим ШАЛОМЄЄВ (відпов. ред.) Електрон. дані. – Запоріжжя : НУ «Запорізька політехніка», 2023. – С. 151-153.
6. Horobets, V.I. Detection of unauthorised actions and attacks in networks of the method of wavelet analysis / V.I. Horobets, V.I. Dubrovin, J.V. Tverdohlib // Прикладні питання математичного моделювання. – 2022. – Т. 5, № 1. – С. 9-20.

Рисунок В.18 – Слайд № 18

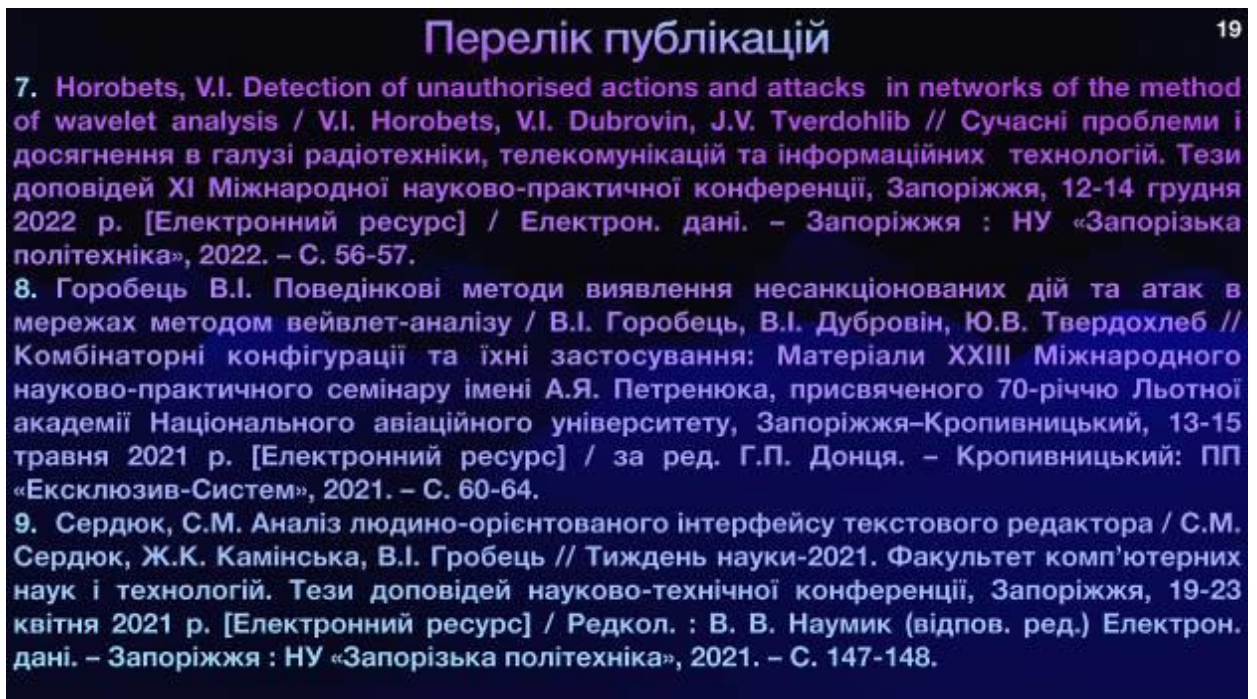


Рисунок В.19 – Слайд № 19