

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «бакалавр»
за спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Інженерія програмного забезпечення»)

на тему:

**«Інформаційна система онлайн-менеджменту
освітніх процесів Digital University UA: модулі
"Кадрове забезпечення", "Статистика та звіти"»**

Виконав студент групи ІПЗ-22-1
КУКСА Руслан Олександрович

Керівник роботи:
МОРОЗОВ Андрій Васильович

Рецензент:
ЧИЖМОТРЯ Олексій Володимирович

Житомир – 2026

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

«ЗАТВЕРДЖУЮ»

Зав. кафедри інженерії
програмного забезпечення
Тетяна Вакалюк

«27» лютого 2026 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **КУКСА Руслан Олександрович**

Керівник роботи: **МОРОЗОВ Андрій Васильович**

Тема роботи: **«Інформаційна система онлайн-менеджменту освітніх процесів Digital University UA: модулі "Кадрове забезпечення", "Статистика та звіти"»**,

затверджена Наказом закладу вищої освіти від **«27» лютого 2026 р., №92/с**

Вихідні дані для роботи: **розробка системи управління освітніми процесами в університеті, що покращить роботу його адміністрації та надасть функціонал для автоматизації та цифровізації рутинних завдань.**

Консультанти з бакалаврської кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Морозов А.В.	27.02.2026р.	27.02.2026р.
2	Морозов А.В.	24.04.2026р.	24.04.2026р.
3	Морозов А.В.	02.05.2026р.	02.05.2026р.

РЕФЕРАТ

Кваліфікаційна робота складається з програмного комплексу для інформаційної системи онлайн-менеджменту освітніх процесів «Digital University UA», зокрема модулів «Кадрове забезпечення» та «Статистика та звіти», а також пояснювальної записки. Пояснювальна записка до випускної роботи містить 99 сторінок, 34 ілюстрації та 1 таблиця.

Головною метою роботи є розробка модулів інформаційної системи, призначених для автоматизації обліку кадрового складу закладу вищої освіти та формування статистичних і аналітичних звітів. Розроблена система має забезпечувати зручність використання для різних категорій користувачів і підвищувати ефективність управління інформаційними процесами.

Під час виконання роботи було сформовано основні вимоги до програмного забезпечення, проведено аналіз наявних аналогів інформаційних систем та обрано технологічний стек, необхідний для реалізації поставлених завдань.

КЛЮЧОВІ СЛОВА: ІНФОРМАЦІЙНА СИСТЕМА, КАДРОВЕ ЗАБЕЗПЕЧЕННЯ, СТАТИСТИКА ТА ЗВІТИ, PHP, Yii2, ЄДЕБО.

					ІПЗ.КР.Б – 121 – 26 – ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	Інформаційна система онлайн-менеджменту освітніх процесів Digital University UA: модулі "Кадрове забезпечення", "Статистика та звіти" Пояснювальна записка	Літ.	Арк.	Аркушів
Розроб.		Кукса Р.О.					3	99
Керівник		Морозов А. В.				Житомирська політехніка, група ІПЗ-22-1		
Н. контр.								
Зав. каф.		Вакалюк Т.А.						

ABSTRACT

The qualification work consists of a software package for the online educational process management information system “Digital University UA”, particularly the modules “Human Resources Management” and “Statistics and Reports”, as well as an explanatory note. The explanatory note contains 99 pages, 34 illustrations and 1 table.

The main purpose of the work is to develop modules of the information system intended to automate the management of academic staff records in higher education institutions and generate statistical and analytical reports. The developed system is designed to provide ease of use for different categories of users and improve the efficiency of information management processes.

During the development process, the main software requirements were defined, existing analogs of information systems were analyzed, and an appropriate technology stack necessary for implementing the assigned tasks was selected.

KEYWORDS: INFORMATION SYSTEM, HUMAN RESOURCES MANAGEMENT, STATISTICS AND REPORTS, PHP, Yii2, EDEBO.

		Кукса Р.О.			ІІЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

ЗМІСТ

РЕФЕРАТ.....	3
ABSTRACT	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	7
ВСТУП.....	8
РОЗДІЛ 1. АНАЛІЗ ВИМОГ КОРИСТУВАЧА ТА КОНЦЕПТУАЛЬНЕ МОДЕЛЮВАННЯ	10
1.1. Постановка задачі	10
1.2. Аналіз аналогів програмного продукту	11
1.3. Вибір архітектури системи	16
1.4. Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення.....	19
Висновки до першого розділу.....	22
РОЗДІЛ 2. ПРОЕКТУВАННЯ СИСТЕМИ ОНЛАЙН-МЕНЕДЖМЕНТУ ОСВІТНІХ ПРОЦЕСІВ.....	24
2.1. Визначення варіантів використання та структури системи	24
2.2. Обґрунтування вибору системи управління базами даних.....	27
2.3. Розробка бази даних системи	32
2.4. Реалізація модуля «Кадрове забезпечення».....	35
2.5. Реалізація модуля «Статистика та звіти».....	41
Висновки до другого розділу.....	51
РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З СИСТЕМОЮ ОНЛАЙН-МЕНЕДЖМЕНТУ ОСВІТНІХ ПРОЦЕСІВ.....	53
3.1. Порядок встановлення та налаштування параметрів додатку.....	53
3.2. Структура інтерфейсу додатку	56
3.3. Тестування роботи розробленого додатку.....	73

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		5

Висновки до третього розділу	76
ВИСНОВКИ.....	78
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	80
ДОДАТКИ.....	83
Додаток А.....	84

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД – база даних

ЗВО – заклад вищої освіти

МОН – Міністерство освіти і науки України

НПП – Науково-педагогічний працівник

CRUD – Create-Read-Update-Delete

СУБД – Система управління базою даних

ЄДЕБО – Єдина державна електронна база з питань освіти

		Кукса Р.О.			ПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		7

ВСТУП

Цифрова трансформація закладів вищої освіти є одним із ключових напрямів розвитку сучасного освітнього середовища. Університети щоденно працюють із великими обсягами даних, що стосуються здобувачів освіти, науково-педагогічних працівників, кадрових документів, навчального навантаження, розкладу, вступної кампанії та внутрішньої звітності. Значна частина цих процесів у багатьох закладах досі виконується із застосуванням розрізнених електронних таблиць, локальних файлів або паперових документів, що ускладнює швидкий доступ до актуальної інформації, підвищує ризик дублювання даних і створює додаткове навантаження на адміністративний персонал.

Актуальність теми посилюється тим, що ефективне управління освітніми процесами сьогодні потребує не лише збереження інформації, а й оперативного її аналізу. Адміністрація закладу вищої освіти повинна мати можливість швидко отримувати достовірні відомості про кадровий склад, стан документів науково-педагогічних працівників, виконання ліцензійних вимог, динаміку вступної кампанії, рух контингенту здобувачів освіти та інші показники, які впливають на якість управлінських рішень. За відсутності єдиної інформаційної системи такі дані часто збираються вручну з різних джерел, що збільшує час підготовки звітів і може призводити до помилок.

Особливо важливим є впровадження цифрових інструментів у кадрових та аналітичних процесах університету. Кадрове забезпечення пов'язане з обліком великої кількості персональних, професійних і документальних даних працівників, зокрема інформації про посади, зайнятість, кваліфікацію, публікації, підвищення кваліфікації, контракти та виконання нормативних вимог. Водночас модуль статистики та звітів має забезпечувати узагальнення цих даних і формування аналітичної інформації для керівництва, структурних підрозділів та відповідальних служб. Саме тому створення інтегрованих модулів «Кадрове забезпечення» та «Статистика та звіти» у складі інформаційної системи є практично значущим завданням.

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		8

Метою кваліфікаційної роботи є проєктування та розробка програмних модулів інформаційної системи, призначених для автоматизації кадрового обліку, зберігання та опрацювання інформації про працівників закладу вищої освіти, а також для формування статистичних і аналітичних звітів.

Досягнення поставленої мети передбачає аналіз предметної області та наявних програмних рішень для управління освітніми процесами, визначення функціональних і нефункціональних вимог до розроблюваних модулів, обґрунтування архітектури системи та вибору інструментальних засобів, проєктування структури бази даних, реалізацію основного функціоналу модулів «Кадрове забезпечення» і «Статистика та звіти», а також перевірку працездатності створеного програмного рішення на основі ключових користувацьких сценаріїв.

Об'єктом дослідження є процеси інформаційного забезпечення управління освітньою діяльністю закладу вищої освіти, зокрема процеси ведення кадрових даних, обліку документів працівників та формування статистичної звітності.

Предметом дослідження є методи, програмні засоби та архітектурні рішення, що використовуються для створення веборієнтованої інформаційної системи автоматизації кадрових і аналітичних процесів закладу вищої освіти.

		Кукса Р.О.			ПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. АНАЛІЗ ВИМОГ КОРИСТУВАЧА ТА КОНЦЕПТУАЛЬНЕ МОДЕЛЮВАННЯ

1.1. Постановка задачі

Сучасні інформаційні системи закладів вищої освіти потребують ефективних інструментів для управління людським капіталом та глибокого аналізу освітніх процесів. З постійним збільшенням обсягів даних щодо вступників, здобувачів освіти та науково-педагогічних працівників, критично важливою стає потреба в автоматизації кадрового діловодства та впровадженні потужних систем для формування статистичної звітності, які б відповідали за комплексний моніторинг діяльності установи.

Значна кількість освітніх закладів досі застосовує розрізнені або паперові підходи до ведення кадрової документації та агрегації звітів. Такий стан речей перевантажує адміністративний персонал, призводить до дублювання інформації та суттєво ускладнює процеси прийняття управлінських рішень. Окрім цього, відсутність єдиного центру збору статистики не дозволяє оперативно реагувати на зміни в освітньому процесі та якісно контролювати ключові показники університету.

Для подолання цих викликів доцільним є впровадження модулів «Кадрове забезпечення» та «Статистика та звіти» до складу інформаційної системи. Кадровий модуль має забезпечувати повноцінний цикл роботи з персоналом: від централізованого ведення профілів працівників, їхніх документів та публікацій до управління записами про зайнятість.

Окрему увагу в розробці приділено автоматизації рутинних адміністративних завдань. Модуль дозволяє генерувати контракти на основі готових шаблонів, розраховувати погодинну оплату, автоматизувати супровід конкурсів, а також контролювати виконання специфічних нормативів (зокрема, п. 38) та вести статистику табелів робочого часу. Це дає змогу суттєво знизити навантаження на відділ кадрів та мінімізувати вплив людського фактора.

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Морозов А.В.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

Не менш значущою є роль аналітичного інструментарію. Модуль статистики та звітів призначений для консолідації та візуалізації ключових метрик ЗВО. Він забезпечує прозорий моніторинг перебігу вступної кампанії, облік документів НПП та їхньої публікаційної активності. Завдяки системі аналізу адміністрація отримує доступ до актуальних даних щодо стану та руху контингенту здобувачів освіти, розкладу занять і захисту кваліфікаційних робіт.

Впровадження вказаних компонентів сприятиме підвищенню загальної ефективності управління університетом, забезпечить прозорість кадрових процесів та надасть керівництву надійну інформаційну базу для стратегічного планування.

Ключові завдання розробленої системи полягають у цифровізації обліку документів, публікацій і профілів працівників, веденні електронної історії зайнятості персоналу, автоматизації створення контрактів на основі шаблонів, організації обліку погодинної оплати та статистики табелів робочого часу. Окремим напрямом є цифровий супровід конкурсів і моніторинг виконання вимог пункту 38 ліцензійних умов. Для модуля «Статистика та звіти» важливими завданнями є збір і обробка аналітичних даних щодо вступної кампанії, моніторинг стану та руху контингенту здобувачів освіти, узагальнення інформації про публікаційну активність і документи НПП, формування статистичних вибірок щодо кваліфікаційних робіт та побудова звітності на основі даних університетського розкладу.

1.2. Аналіз аналогів програмного продукту

У відповідь на сучасні виклики в освітньому середовищі ринок програмного забезпечення пропонує низку цифрових рішень для оптимізації управління навчальним процесом. Під час дослідження було проаналізовано спеціалізовані системи, що забезпечують підтримку діяльності закладів вищої освіти.

Moodle [2] – це популярна система управління навчанням (LMS) із відкритим вихідним кодом, яка є одним із лідерів у сфері організації

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Морозов А.В.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

дистанційної освіти. Платформа пропонує розгалужений інструментарій для конструювання електронних курсів, проведення тестувань, моніторингу успішності здобувачів освіти та налагодження інтерактивної взаємодії між викладачами й студентами. Фундаментальною перевагою Moodle виступає її модульна архітектура, що дає змогу гнучко адаптувати середовище під специфічні вимоги будь-якого закладу освіти за допомогою встановлення додаткових плагінів.

Серед важливих характеристик системи – здатність інтегруватися з провідними сервісами відеоконференцій та спільної роботи (Zoom, Google Workspace, Microsoft Teams), що є критично важливим для забезпечення синхронного онлайн-навчання. Крім того, Moodle повністю підтримує загальноприйняті стандарти електронного навчання, зокрема SCORM та xAPI, гарантуючи сумісність із зовнішніми навчальними матеріалами.

Завдяки глобальній спільноті розробників, платформа має потужну інформаційну підтримку, що спрощує розгортання та обслуговування. Важливим фактором є те, що базова версія системи поширюється на безоплатній основі, що робить її привабливою для ЗВО з будь-яким бюджетом (рис. 1.1).

Офіційний сайт: <https://moodle.org/>

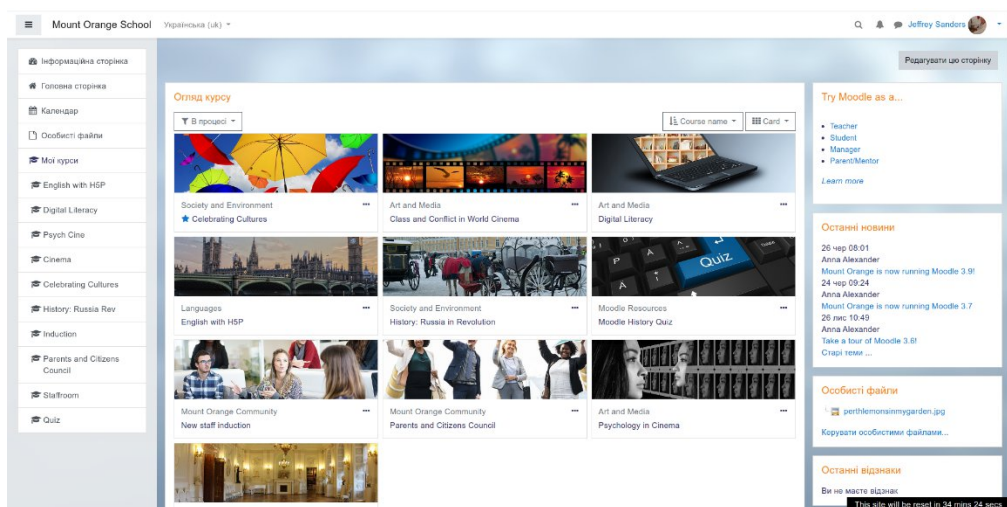


Рисунок 1.1 – Платформа Moodle

Головними перевагами Moodle залишаються відкритість коду, що дозволяє створювати кастомні рішення, кросплатформеність та висока гнучкість налаштувань. Однак система має і низку суттєвих недоліків. Для її розгортання

		Кукса Р.О.			ПІЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

й адміністрування потрібен кваліфікований ІТ-персонал. У разі великого навантаження система потребує серйозної оптимізації серверних потужностей для забезпечення стабільності. Також інтерфейс може здаватися перевантаженим для нових користувачів, а базовий функціонал орієнтований виключно на навчальний процес, залишаючи поза увагою адміністративні, кадрові чи фінансові питання ЗВО. Отже, Moodle є потужним інструментом для організації навчання, але потребує доповнення іншими системами для комплексного управління університетом.

АСУ ВНЗ [3] — це спеціалізована вітчизняна інформаційна система, розроблена для комплексної автоматизації управління навчальним процесом в українських закладах вищої освіти. Цей програмний комплекс орієнтований на вирішення адміністративних завдань, що виникають у повсякденній роботі ректорату, деканатів, кафедр та приймальної комісії. Головною особливістю системи є її глибока адаптація до вимог українського законодавства та стандартів вітчизняного документообігу.

Система дозволяє автоматизувати облік студентського контингенту, формування наказів, розрахунків стипендій, створення навчальних планів та генерацію додатків до дипломів європейського зразка. Критично важливою функцією «АСУ ВНЗ» є вбудована можливість синхронізації даних із Єдиною державною електронною базою з питань освіти (ЄДЕБО), що значно спрощує роботу приймальних комісій та відділів кадрів (рис. 1.2).

Офіційний сайт: <https://vuz.osvita.net>

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

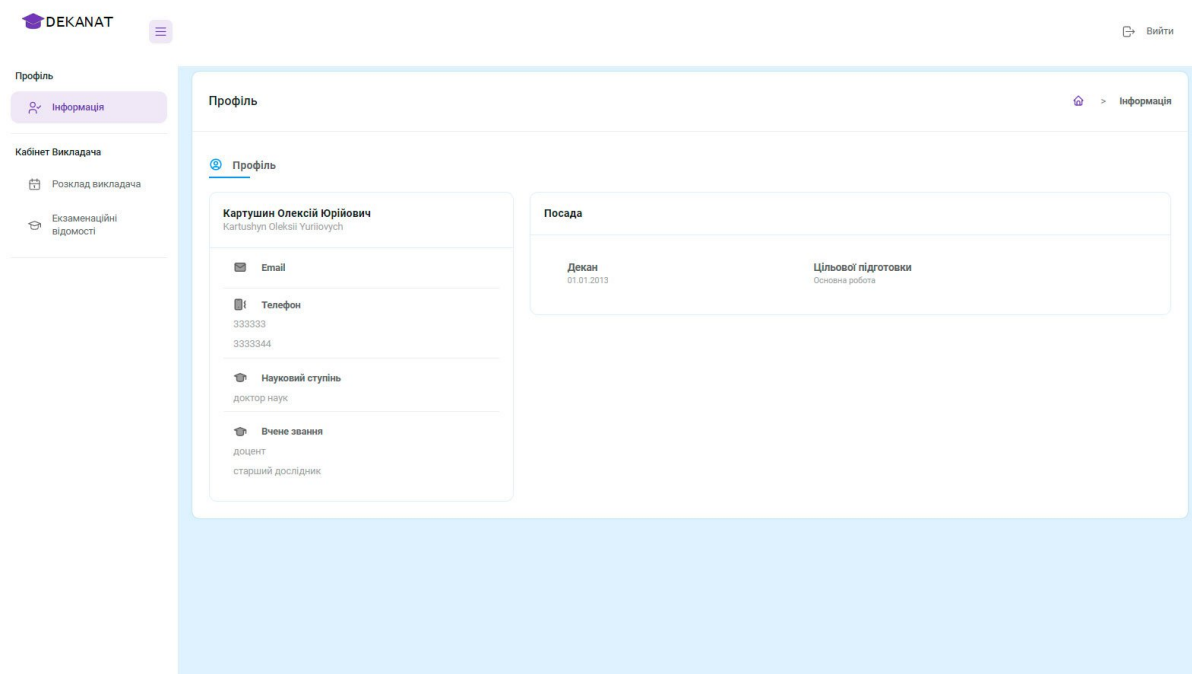


Рисунок 1.2 – Система «АСУ ВНЗ»

До ключових переваг цієї системи належить її здатність ефективно оптимізувати бюрократичні процедури та зменшувати рутинне навантаження на адміністративний персонал. Вона забезпечує централізоване зберігання даних про студентів та викладачів.

Однак система має і помітні недоліки. Насамперед, це закритий вихідний код та комерційна модель поширення, що вимагає регулярних фінансових витрат на ліцензування та технічну підтримку. Крім того, інтерфейс користувача здебільшого є застарілим і менш інтуїтивним порівняно з сучасними вебдодатками. Функціонал для безпосередньої організації дистанційного навчання у таких системах зазвичай або відсутній, або значно поступається спеціалізованим платформам.

ПС-Університет [4] — це ще одне потужне комплексне рішення на ринку України, призначене для інформатизації процесів управління закладами вищої освіти. Система побудована за модульним принципом і охоплює практично всі аспекти життєдіяльності університету: від роботи приймальної комісії до управління кадрами, фінансами та планування навчального процесу.

Програмний комплекс включає підсистеми для деканатів (ведення карток студентів, облік успішності), навчального відділу (розподіл навантаження,

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				14
Змн.	Арк.	№ докум.	Підпис	Дата		

складання розкладу занять), а також модулі для бухгалтерії. Завдяки єдиній корпоративній базі даних, «ПС-Університет» забезпечує цілісність інформації та унеможливорює її дублювання різними структурними підрозділами. Система також підтримує формування різноманітної статистичної та аналітичної звітності за формами, затвердженими Міністерством освіти і науки України (рис. 1.3).

Офіційний сайт: <https://www.politek-soft.kiev.ua>

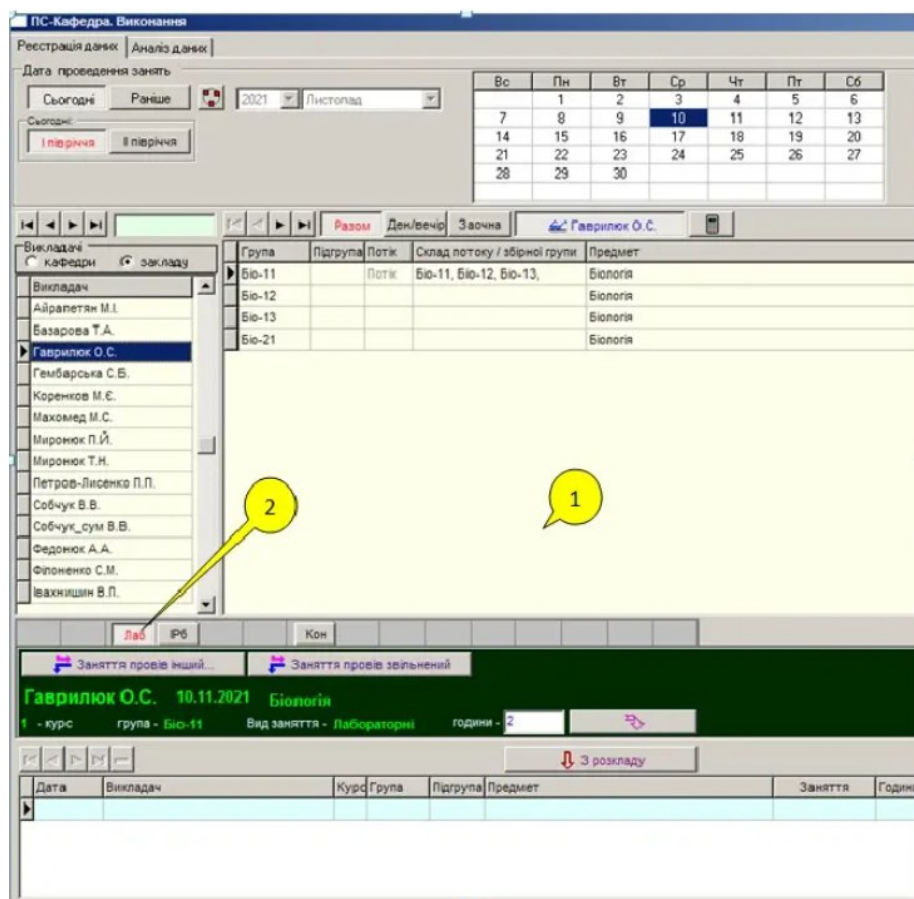


Рисунок 1.3 – Система «ПС-Університет»

Перевагами «ПС-Університет» є високий рівень безпеки даних, можливість розгортання в закритій локальній мережі університету та всеосяжне покриття адміністративних процесів ЗВО.

Разом із тим, впровадження такого масштабу є тривалим і технічно складним процесом. Користувачі часто відзначають високий поріг входження та необхідність тривалого навчання персоналу. Архітектура системи, яка часто базується на класичних десктопних додатках, ускладнює кросплатформений доступ і віддалене використання з мобільних пристроїв. Вартість розгортання,

кастомізації та супроводу такої АСУ є доволі високою, що робить її доцільною переважно для великих університетів, готових до значних інвестицій в ІТ-інфраструктуру.

1.3. Вибір архітектури системи

У сучасних інформаційних системах закладів вищої освіти фундаментальне значення відіграє грамотне проєктування архітектури програмного забезпечення. Це є запорукою стабільності, безпеки та здатності системи до подальшого масштабування. Постійне зростання обсягів даних та кількості активних користувачів (здобувачів освіти, науково-педагогічних працівників, адміністративного персоналу) вимагає ефективного управління кодовою базою та логікою взаємодії між компонентами. Правильний архітектурний підхід значно спрощує подальший розвиток та технічну підтримку цифрового продукту.

Традиційним та найбільш розповсюдженим підходом є використання монолітної архітектури, за якої всі функціональні компоненти – серверна логіка, взаємодія з базою даних, модулі кадрового діловодства та генерації аналітичної звітності – об'єднані в єдину цілісну структуру. Така модель забезпечує відносну простоту розробки та високу швидкість початкового розгортання. Усі складові системи функціонують у спільному середовищі, що робить процеси налагодження, тестування та моніторингу максимально прозорими. Крім того, завдяки відсутності мережових затримок між внутрішніми компонентами, моноліт демонструє високу продуктивність під час виконання комплексних запитів, а єдина кодова база полегшує управління версіями.

Однак із розширенням функціоналу інформаційної системи монолітна структура може стати надто масивною. Збільшення кількості бізнес-правил підвищує ризик виникнення конфліктів між окремими модулями. Зміни в одній частині коду можуть непередбачувано вплинути на інші компоненти, що ускладнює розробку нового функціоналу та вимагає повного регресійного тестування всієї системи навіть після незначних оновлень.

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				16
Змн.	Арк.	№ докум.	Підпис	Дата		

Як альтернатива моноліту, у сучасній ІТ-індустрії активно застосовується мікросервісна архітектура. Вона базується на декомпозиції моноліту на ізольовані невеликі сервіси, кожен з яких виконує власну вузьку бізнес-задачу. Наприклад, у системі можна було б виділити абсолютно незалежні сервіси для управління профілями працівників, обробки статистичних даних вступної кампанії та ведення розкладу. Вони взаємодіють між собою через чітко визначені мережеві API. Це дозволяє командам розробників працювати паралельно та незалежно масштабувати лише ті компоненти, які зазнають найбільшого навантаження (наприклад, сервіс статистики в період формування семестрових звітів).

Попри очевидні переваги, мікросервісний підхід супроводжується значними технічними та інфраструктурними викликами. Він вимагає ретельного планування мережевих з'єднань, впровадження складних систем оркестрації та моніторингу. Розподіленість компонентів неминуче призводить до мережевих затримок, що впливає на загальну швидкість. Крім того, виникають серйозні архітектурні труднощі із забезпеченням цілісності даних між, наприклад, модулем кадрів та глобальною статистикою.

Остаточний вибір архітектурного патерну залежить від специфіки бізнес-процесів, масштабу проєкту та способу взаємодії компонентів. Для реалізації модулів «Кадрове забезпечення» та «Статистика та звіти» в межах інформаційної системи оптимальним та обґрунтованим рішенням було обрано монолітну архітектуру.

Головним аргументом на користь цього вибору є надзвичайно тісна зв'язність даних: агрегація глобальної інституційної статистики безпосередньо і нерозривно залежить від кадрової інформації, руху контингенту та інших суміжних процесів. Більшість компонентів ефективно взаємодіють усередині єдиного середовища на базі фреймворку Yii2 з використанням спільної реляційної бази даних. Це дозволяє використовувати складні, оптимізовані SQL-запити із множинними об'єднаннями таблиць для миттєвої генерації звітів, усуваючи необхідність пересилання гігабайт даних через мережу між

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				17
Змн.	Арк.	№ докум.	Підпис	Дата		

мікросервісами. Монолітна архітектура в цьому випадку забезпечує централізований контроль над складною бізнес-логікою ЗВО, гарантує абсолютну консистентність даних та робить процес розробки максимально ефективним.

У контексті розроблюваної системи монолітна архітектура не означає відсутність внутрішнього поділу на логічні компоненти. Навпаки, у межах фреймворку Yii2 передбачено чітке розмежування відповідальності між моделями, контролерами, представленнями, сервісними класами та компонентами доступу до даних. Такий підхід дозволяє зберегти простоту фізичного розгортання системи, але водночас уникнути хаотичного змішування бізнес-логіки, логіки інтерфейсу та операцій із базою даних. Для модулів «Кадрове забезпечення» та «Статистика та звіти» це має особливе значення, оскільки обидві частини системи використовують спільні довідники, єдину систему авторизації, загальні механізми фільтрації та одну реляційну базу даних.

Важливою перевагою обраної архітектури є можливість централізованого керування правами доступу. У системі, яка працює з кадровими документами, персональними даними працівників і статистичними показниками університету, неконтрольований доступ до інформації є неприпустимим. Монолітна архітектура спрощує реалізацію єдиного механізму перевірки ролей користувачів, оскільки всі запити проходять через спільний програмний контур. Це дає змогу гарантувати, що працівник відділу кадрів, адміністратор, представник деканату або керівник структурного підрозділу отримує доступ лише до тих функцій, які відповідають його повноваженням.

Окремо слід зазначити, що монолітний підхід є зручним для реалізації складних транзакційних операцій. Наприклад, під час оновлення інформації про працівника можуть одночасно змінюватися його кадрові записи, документи, статуси перевірки, дані про зайнятість і показники, які використовуються у звітності. Якщо такі частини системи були б винесені в окремі мікросервіси, виникла б потреба в додаткових механізмах синхронізації та контролю узгодженості даних. У межах єдиного застосунку ці процеси простіше

		Кукса Р.О.			ПІЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				18
Змн.	Арк.	№ докум.	Підпис	Дата		

контролювати за допомогою транзакцій бази даних і внутрішніх сервісних методів.

Водночас обрана архітектура залишає можливість майбутнього масштабування. Якщо в подальшому окремі частини системи почнуть створювати надмірне навантаження або потребуватимуть незалежного розвитку, їх можна буде поступово винести в окремі сервіси без повної перебудови програмного продукту. Насамперед це може стосуватися інтеграції з ЄДЕБО, обробки великих статистичних вибірок або генерації ресурсомістких звітів. Тобто монолітна архітектура на поточному етапі не обмежує розвиток системи, а створює стабільну основу для її першочергової реалізації.

Таким чином, вибір монолітної архітектури є обґрунтованим з огляду на масштаб кваліфікаційної роботи, тісну пов'язаність кадрових і статистичних даних, необхідність забезпечення цілісності інформації та потребу в швидкій реалізації основного функціоналу. Такий підхід дозволяє зменшити технічну складність проєкту, спростити тестування, забезпечити стабільну взаємодію між модулями та створити програмну основу, придатну для подальшого розвитку інформаційної системи.

1.4. Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення

Наступним важливим етапом проєктування стало визначення та обґрунтування технологічного стеку для розробки програмного забезпечення. Враховуючи специфіку архітектури веб-додатків для закладів вищої освіти, базовою мовою програмування було обрано PHP. Це рішення зумовлене її доведеною надійністю в сегменті веб-розробки, розвиненою екосистемою, наявністю вичерпної документації та активною підтримкою міжнародної спільноти.

Для безпосередньої реалізації бізнес-логіки модулів «Кадрове забезпечення» та «Статистика та звіти» застосовано фреймворк Yii2. Цей інструмент гарантує гнучкість архітектурних рішень, забезпечує високий рівень

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Морозов А.В.				19
Змн.	Арк.	№ докум.	Підпис	Дата		

безпеки компонентів та дозволяє максимально швидко створювати необхідний функціонал завдяки вбудованим механізмам генерації коду. Як систему управління реляційними базами даних (СУБД) використано MySQL, що є індустріальним стандартом та гарантує стабільність і високу продуктивність під час обробки масивних таблиць із кадровими даними та аналітичними вибірками.

Окремою перевагою використання РНР є його практична придатність для створення веборієнтованих інформаційних систем, які мають працювати у браузері без встановлення додаткового програмного забезпечення на комп'ютери користувачів. Для закладу вищої освіти це важливо, оскільки доступ до системи можуть отримувати працівники різних структурних підрозділів, які використовують різні робочі місця, операційні системи та браузери. Серверна обробка запитів дозволяє централізовано оновлювати програмний код, контролювати версії системи та уникати необхідності ручного встановлення оновлень на кожному окремому пристрої.

Фреймворк Yii2 було обрано через його відповідність архітектурним потребам розроблюваної системи. Він підтримує MVC-підхід, має вбудовані засоби маршрутизації, роботи з формами, валідації даних, авторизації користувачів і взаємодії з базою даних через Active Record. Для модулів «Кадрове забезпечення» та «Статистика та звіти» це дозволяє організувати кодову базу у зрозумілий спосіб, де окремі частини функціоналу не змішуються між собою. Наприклад, логіка перевірки документів НПП може бути винесена в окремі моделі та сервіси, тоді як контролери відповідатимуть лише за приймання запитів і передавання результатів у представлення.

Важливим чинником вибору Yii2 є також наявність механізму міграцій бази даних. Під час розробки інформаційної системи структура таблиць може поступово змінюватися, оскільки додаються нові поля, уточнюються зв'язки або створюються додаткові довідники. Використання міграцій дає змогу фіксувати такі зміни у вигляді програмного коду, що спрощує розгортання системи на різних середовищах і зменшує ризик помилок під час оновлення бази даних. Для університетської інформаційної системи це особливо актуально, оскільки її

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				20
Змн.	Арк.	№ докум.	Підпис	Дата		

структура може змінюватися відповідно до нових управлінських або нормативних вимог.

З погляду безпеки обраний технологічний стек також є доцільним. Yii2 надає інструменти для перевірки вхідних даних, захисту форм, організації авторизації та контролю доступу до окремих дій. Це важливо для роботи з кадровою інформацією, оскільки система обробляє персональні дані працівників, документи, відомості про зайнятість, публікаційну активність і службову інформацію. Наявність вбудованих засобів безпеки зменшує ймовірність типових помилок під час розробки та дозволяє реалізувати єдині правила доступу до даних.

Для підтримки клієнтської частини системи можуть застосовуватися стандартні веб-технології HTML, CSS і JavaScript. Їх використання забезпечує коректне відображення інтерфейсу в браузері, побудову інтерактивних елементів, роботу фільтрів, таблиць, форм і графічних компонентів. Оскільки система орієнтована на адміністративне використання, інтерфейс має бути не лише функціональним, а й зрозумілим для користувачів без спеціальної технічної підготовки. Саме тому важливим є поєднання серверної стабільності Yii2 із достатньо простими та передбачуваними засобами відображення даних на клієнтському боці.

Доцільним є також використання системи контролю версій для організації розробки та подальшого супроводу програмного продукту. Збереження історії змін у коді дозволяє відстежувати процес реалізації функціоналу, швидко повертатися до попередніх стабільних версій і контролювати внесення змін у різні частини системи. Для кваліфікаційної роботи це забезпечує впорядкованість розробки, а для потенційного промислового використання системи - основу для командної підтримки та розвитку.

Апаратні вимоги до системи визначаються не лише кількістю користувачів, а й характером операцій, які вона виконує. Кадровий модуль працює з великою кількістю пов'язаних записів, а модуль статистики та звітів виконує агрегацію даних, фільтрацію, сортування та формування зведених

		Кукса Р.О.			ПІЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				21
Змн.	Арк.	№ докум.	Підпис	Дата		

показників. Такі операції можуть створювати значне навантаження на процесор, оперативну пам'ять і дискову підсистему, особливо під час одночасної роботи багатьох користувачів або формування великих звітів за тривалий період.

Крім основних технічних характеристик сервера, важливим є налаштування резервного копіювання та моніторингу. Регулярне створення резервних копій бази даних дозволяє відновити систему у випадку технічного збою, помилки адміністратора або пошкодження даних. Моніторинг використання процесора, оперативної пам'яті, дискового простору та часу відповіді сервера дає змогу своєчасно виявляти перевантаження й оптимізувати роботу системи. Це особливо важливо для інформаційної системи закладу вищої освіти, де втрата або недоступність даних може негативно вплинути на роботу адміністративних підрозділів.

Висновки до першого розділу

У першому розділі було проведено системний аналіз предметної області, що дозволило чітко окреслити мету дослідження та сформулювати науково-технічне обґрунтування розробки. Встановлено, що цифровізація вищої освіти вимагає відмови від розрізнених програмних рішень та застарілого паперового документообігу на користь єдиних інтегрованих середовищ. Формування коректної постановки проблеми стало відправною точкою для проєктування програмних модулів «Кадрове забезпечення» та «Статистика та звіти», які виступатимуть ключовими складовими комплексної автоматизованої системи управління закладом вищої освіти. Їх розробка дозволить ефективно автоматизувати діловодство і створити єдиний центр агрегації інституційної статистики.

Аналіз існуючих аналогів підтвердив практичну доцільність власної розробки. Популярні LMS-платформи орієнтовані переважно на навчальний процес і позбавлені адміністративного інструментарію. Водночас наявні вітчизняні комерційні системи відрізняються високою вартістю, застарілим інтерфейсом та складністю гнучкої кастомізації під конкретні потреби установи.

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		22

На основі специфіки бізнес-процесів університету обґрунтовано вибір монолітної архітектури. Через надзвичайно тісну зв'язність даних між кадровим складом, контингентом студентів та глобальною звітністю (зокрема для синхронізації з ЄДЕБО), монолітний підхід гарантує консистентність реляційної бази даних. Це дозволяє оптимізувати виконання складних аналітичних запитів і уникати мережових затримок, які були б неминучими у мікросервісній архітектурі.

Також здійснено вибір інструментальних засобів: використання мови програмування РНР та високорівневого фреймворку Yii2 забезпечує оптимальну швидкість розробки і безпеку компонентів, а СУБД MySQL гарантує надійність обробки значних обсягів інформації. Сформовані вимоги до серверного апаратного забезпечення дозволять підтримувати стабільну швидкодію під час пікових навантажень.

Отже, виконаний системний аналіз і прийняті технічні рішення формують надійну концептуальну основу для практичної реалізації зазначених модулів, що забезпечить ефективну цифрову трансформацію управлінських процесів.

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		23

РОЗДІЛ 2. ПРОЕКТУВАННЯ СИСТЕМИ ОНЛАЙН-МЕНЕДЖМЕНТУ ОСВІТНІХ ПРОЦЕСІВ

2.1. Визначення варіантів використання та структури системи

Для забезпечення надійного функціонування модулів «Кадрове забезпечення» та «Статистика та звіти» в межах інформаційної системи закладу вищої освіти сформульовано низку вимог до розробки [28]. Вони охоплюють як функціональні, так і нефункціональні характеристики, що спрямовані на підтримку високої ефективності, безпеки, масштабованості та сумісності з іншими підсистемами університету.

Бізнес-вимоги до системи визначаються потребою в комплексній автоматизації адміністративних і аналітичних процесів закладу вищої освіти. Модуль «Кадрове забезпечення» має забезпечувати централізоване управління життєвим циклом науково-педагогічних працівників: від конкурсного відбору, укладання контрактів і ведення профілів до обліку кваліфікацій, документів і табелів робочого часу. Модуль «Статистика та звіти» повинен надавати інструменти для агрегації даних щодо вступної кампанії, руху студентського контингенту, виконання розкладу та захисту кваліфікаційних робіт. Інтеграція цих компонентів у межах єдиного вебзастосунку створює достовірне інформаційне середовище для ухвалення управлінських рішень.

Для адміністрації закладу вищої освіти важливим є ведення електронних карток працівників із деталізацією відомостей про наукові ступені, вчені звання, підвищення кваліфікації, участь у проєктах, зайнятість і кадрові документи. Система також повинна підтримувати автоматизацію нарахування погодинної оплати на основі табелів, генерацію контрактів за шаблонами та супровід конкурсів на заміщення вакантних посад. Для аналітичного напрямку важливим є швидкий доступ до зведених даних про рух контингенту, результати вступних кампаній і виконання навантаження викладачами.

Власники системи потребують гнучкого налаштування ролей доступу, щоб чітко розмежувати права між відділом кадрів, деканатами, ректоратом та ІТ-відділом. Важливою умовою є ведення системного журналу, у якому

		Кукса Р.О.			ПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				24
Змн.	Арк.	№ докум.	Підпис	Дата		

фіксуються зміни в кадрових і статистичних даних, а також наявність механізму резервного копіювання бази даних. Об'єктом комп'ютеризації є процеси управління персоналом ЗВО та агрегації внутрішньої статистики, а основними користувачами виступають співробітники відділу кадрів, навчально-методичного відділу, бухгалтерії та керівний склад університету.

Функціонально система має забезпечувати створення та редагування профілів працівників, облік зайнятості, ведення реєстру публікацій і досягнень, формування табелів робочого часу, управління контрактами та супровід конкурсного відбору. Аналітична частина системи повинна забезпечувати збір і агрегацію даних про вступні кампанії, аналіз документів НПП, відстеження кваліфікаційних робіт, моніторинг контингенту та формування звітів. Спільними вимогами для обох модулів є надійна авторизація, пошук, фільтрація даних і можливість експорту результатів.

Нефункціональні вимоги стосуються зручності, надійності, продуктивності та подальшої експлуатації системи. Інтерфейс має бути інтуїтивно зрозумілим, а відгук системи на базові операції, зокрема перегляд профілю або збереження форми, не повинен перевищувати двох секунд. Для складних статистичних звітів із великими обсягами даних допускається більший час обробки, однак він не має перевищувати десяти секунд. Система повинна забезпечувати доступність не менше 99 %, підтримувати відновлення після критичного збою впродовж 15 хвилин і зберігати цілісність даних під час взаємодії кадрового модуля з іншими підсистемами. Програмний комплекс має підтримувати одночасну роботу щонайменше 1000 авторизованих користувачів, коректно працювати з великими вибірками даних і мати адаптивний інтерфейс для різних типів пристроїв.

Після детального дослідження усіх необхідних характеристик майбутнього програмного забезпечення та виокремлення ключових функцій, з якими будуть взаємодіяти актори системи, було розроблено відповідну діаграму варіантів використання (рис. 2.1):

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				25
Змн.	Арк.	№ докум.	Підпис	Дата		

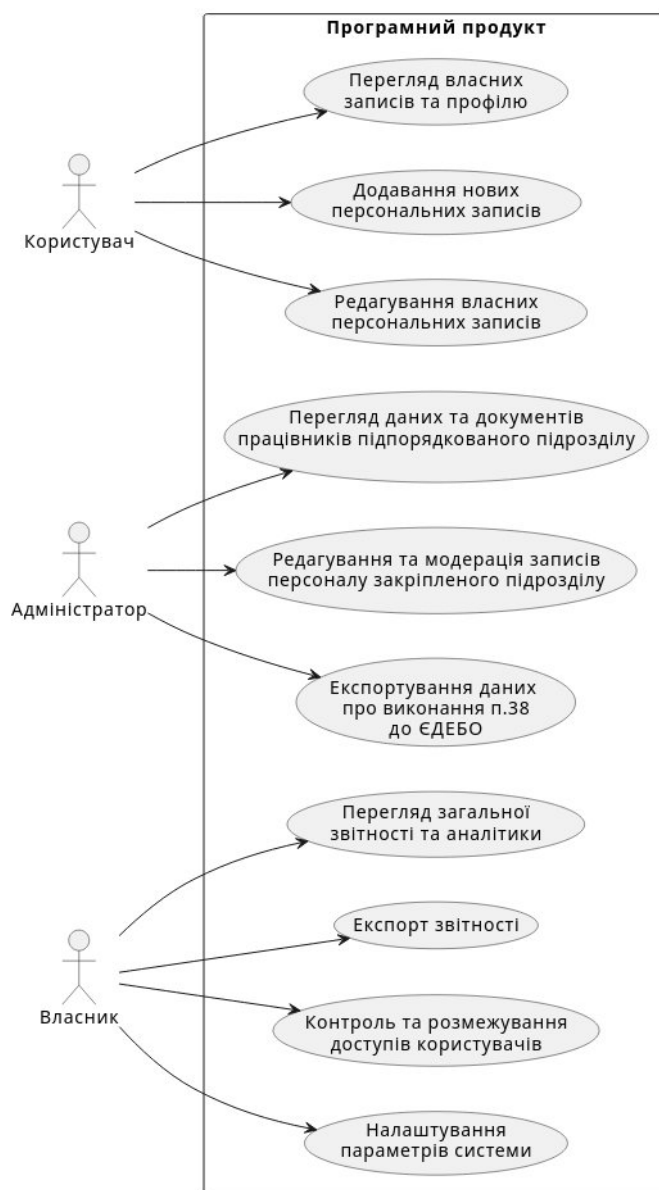


Рисунок 2.1 - Діаграма використання системи

Головними користувачами спроектованих модулів є адміністративний персонал - різноманітні групи співробітників університету (відділ кадрів, навчальна частина, деканати), які, відповідно до своїх посадових обов'язків, наділяються специфічними ролями. Ці ролі не є жорстко закодованими; архітектура вебпорталу надає супер-адміністраторам гнучкий інструментарій для самостійного створення нових ролей та налаштування деталізованих матриць доступу. Це дозволяє динамічно масштабувати зону відповідальності кожної посадової особи залежно від змін у штатній структурі.

Модулі «Кадрове забезпечення» та «Статистика та звіти» інкапсулюють величезний масив специфічного функціоналу, що стосується управління

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				26
Змн.	Арк.	№ докум.	Підпис	Дата		

життєвим циклом персоналу та агрегації глобальних університетських метрик. Актор «Власник», який представляє технічний підрозділ, несе відповідальність за підтримку життєздатності системи. Для ефективного обслуговування він потребує спеціальних інструментів: можливості тимчасового обмеження доступу користувачів на період розгортання оновлень або проведення регламентних технічних робіт та засобів для налаштування базових параметрів порталу в режимі реального часу без необхідності безпосереднього втручання у вихідний програмний код.

2.2. Обґрунтування вибору системи управління базами даних

У процесі розробки програмних модулів «Кадрове забезпечення» та «Статистика та звіти» було обґрунтовано доцільність використання реляційної системи керування базами даних MySQL [16]. Вибір СУБД є одним із ключових етапів проєктування інформаційної системи, оскільки саме база даних забезпечує збереження, цілісність, пошук, оновлення та подальше використання всієї інформації, з якою працює програмний продукт. Для системи онлайн-менеджменту освітніх процесів це має особливе значення, адже вона опрацьовує значні обсяги взаємопов'язаних даних, що стосуються кадрового складу, документів працівників, публікаційної активності, навчального навантаження, вступних кампаній, контингенту здобувачів освіти та статистичної звітності.

У межах розроблюваної системи база даних виконує не лише роль сховища інформації, а й виступає основою для взаємодії між окремими функціональними модулями. Дані, внесені в модулі «Кадрове забезпечення», можуть використовуватися під час формування аналітичних звітів, перевірки виконання ліцензійних вимог, підготовки внутрішньої статистики та створення зведених вибірок для адміністрації закладу вищої освіти. Тому обрана СУБД повинна забезпечувати стабільну роботу з реляційними зв'язками, підтримувати складні запити, гарантувати узгодженість інформації та мати достатню продуктивність для роботи з великими наборами даних.

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		27

MySQL характеризується високою продуктивністю, відмовостійкістю та гнучкістю масштабування. Ці властивості є критично важливими для системи, що працює зі значними масивами інституційних даних. Програмний продукт повинен безперебійно обробляти процеси створення профілів науково-педагогічних працівників, зберігати історії їхніх кваліфікацій, документів, контрактів і записів про зайнятість, а також оперативно генерувати аналітичні дані щодо вступних кампаній і руху контингенту студентів. У таких умовах використання надійної реляційної СУБД є доцільним, оскільки більшість об'єктів предметної області мають чітку структуру та логічні зв'язки між собою.

Важливою перевагою MySQL для розроблюваної системи є підтримка транзакцій, зовнішніх ключів, індексів і механізмів оптимізації запитів. Для кадрового модуля це має принципове значення, оскільки дані одного працівника пов'язані з багатьма сутностями: документами, публікаціями, записами про зайнятість, контрактами, табелями робочого часу та результатами перевірки ліцензійних вимог. У разі зміни одного з таких записів система повинна зберігати цілісність усієї інформаційної моделі, щоб оновлені кадрові дані коректно використовувалися в аналітичних звітах і не створювали суперечностей між різними частинами системи.

Транзакційність є особливо важливою під час виконання операцій, які передбачають одночасну зміну кількох пов'язаних записів. Наприклад, під час створення або оновлення контракту працівника можуть змінюватися дані про його посаду, зайнятість, підрозділ, період роботи та відповідні службові статуси. Якщо одна з операцій виконається некоректно, а інші зміни вже будуть збережені, це може призвести до появи неузгоджених даних у системі. Використання транзакцій дозволяє виконувати такі операції як єдиний логічний процес: або всі зміни успішно фіксуються в базі даних, або система повертається до попереднього стабільного стану. Завдяки цьому підвищується надійність програмного продукту та зменшується ризик помилок у кадровому обліку.

Зовнішні ключі забезпечують контроль логічних зв'язків між таблицями. Для інформаційної системи закладу вищої освіти це важливо, оскільки в ній не

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				28
Змн.	Арк.	№ докум.	Підпис	Дата		

повинні існувати ізольовані або некоректно пов'язані записи. Наприклад, документ НПП має бути пов'язаний із конкретним працівником, запис про зайнятість - із відповідною посадою та підрозділом, а статистичний показник - із конкретним періодом або категорією даних. Якщо такі зв'язки не контролювати на рівні бази даних, у системі можуть з'являтися помилки, які складно виявити на етапі формування звітів. Саме тому використання механізмів забезпечення цілісності даних є важливою умовою для стабільної роботи розроблюваної системи.

Для модуля «Статистика та звіти» важливою є здатність СУБД ефективно виконувати агрегатні запити, групування та фільтрацію даних за роками, факультетами, спеціальностями, статусами, категоріями працівників та іншими параметрами. Оскільки статистичні звіти формуються на основі значних обсягів історичних даних, використання індексів і правильно побудованих зв'язків між таблицями дозволяє зменшити час обробки запитів і забезпечити прийнятну швидкодію системи навіть під час пікових навантажень. Це особливо актуально в періоди підготовки звітної документації, аналізу результатів вступної кампанії або формування зведених показників щодо кадрового складу.

Індексація є одним із основних інструментів оптимізації роботи бази даних. У межах розроблюваної системи доцільним є створення індексів для полів, які часто використовуються під час пошуку, фільтрації та сортування. До таких полів можуть належати ідентифікатори працівників, структурних підрозділів, спеціальностей, статусів, дат створення записів, періодів звітності та інших параметрів. Завдяки індексам система може швидше знаходити потрібні записи без повного перегляду великих таблиць. Це безпосередньо впливає на зручність роботи користувачів, адже адміністративний персонал очікує швидкого відкриття сторінок, фільтрації списків і формування звітів.

MySQL є доцільним вибором також з погляду нормалізації даних. Для освітньої інформаційної системи важливо уникати дублювання однакових відомостей у різних таблицях, оскільки це може призвести до розбіжностей у звітах і ускладнити супровід програмного продукту. Реляційна модель дозволяє

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		29

винести повторювані значення в окремі довідники, наприклад довідники посад, структурних підрозділів, типів документів, освітніх програм, спеціальностей або статусів. Завдяки цьому основні таблиці зберігають лише посилання на відповідні записи, а не дублюють текстову інформацію в кожному рядку.

Нормалізована структура бази даних також спрощує оновлення інформації. Якщо, наприклад, змінюється назва структурного підрозділу або уточнюється тип документа, достатньо оновити відповідний запис у довіднику, після чого зміни автоматично відображатимуться в усіх пов'язаних частинах системи. Це зменшує ризик появи різних варіантів написання одних і тих самих даних, що є особливо важливим для звітності. У системі, яка використовується для управління освітніми процесами, якість і узгодженість даних безпосередньо впливають на достовірність аналітичних висновків.

Порівняно з нереляційними сховищами, MySQL краще відповідає характеру даних, які обробляються в межах модулів «Кадрове забезпечення» та «Статистика та звіти». Більшість сутностей у системі мають чітку структуру, визначені поля та стабільні зв'язки між собою. Працівник пов'язаний із посадою, кафедрою, контрактом, документами та публікаціями, а здобувач освіти - з освітньою програмою, групою, статусом і навчальними подіями. Такі залежності природно описуються засобами реляційної бази даних, де можна забезпечити контроль цілісності та логічну узгодженість записів.

Вибір MySQL як основної СУБД зумовлений також її глибокою інтеграцією з мовою програмування PHP [17] та високорівневим фреймворком Yii2, які складають основу серверної частини програмного продукту. Така технологічна сумісність дозволяє максимально ефективно використовувати вбудовані ORM-механізми, що оптимізує реалізацію логіки обміну даними та зводить до мінімуму часові витрати на виконання стандартних CRUD-операцій. Для розробки модулів, у яких передбачено велику кількість форм, таблиць, фільтрів і сторінок редагування, це є важливою перевагою.

ORM-механізми Yii2 дозволяють працювати з таблицями бази даних через моделі, що робить програмний код більш структурованим і зрозумілим. Замість

		Кукса Р.О.			ПІЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		30

безпосереднього написання великої кількості SQL-запитів розробник може використовувати об'єктно-орієнтований підхід, у якому кожна таблиця або сутність представлена відповідним класом. Це спрощує реалізацію бізнес-логіки, перевірку даних, встановлення зв'язків між моделями та повторне використання коду. Водночас фреймворк не обмежує можливість застосування складних SQL-запитів у тих випадках, коли це необхідно для оптимізації статистичних вибірок або формування складної аналітики.

Крім того, MySQL є доцільним вибором з погляду супроводу та адміністрування. Для цієї СУБД доступна велика кількість інструментів резервного копіювання, моніторингу продуктивності та аналізу запитів. Це спрощує підтримку системи після її впровадження, дозволяє своєчасно виявляти проблемні місця в роботі бази даних і забезпечує стабільність функціонування модулів у реальному освітньому середовищі. Для закладу вищої освіти цей аспект є важливим, оскільки інформаційна система повинна працювати стабільно не лише під час демонстрації, а й у процесі щоденного використання різними категоріями працівників.

Для адміністрування та конфігурації сховища даних застосовується сучасне інтегроване середовище розробки DataGrip, яке забезпечує професійний інструментарій для взаємодії з базами даних. Його використання дозволило виконувати проєктування та конфігурацію фізичної структури бази даних, зокрема створення таблиць, оптимізацію індексів, встановлення зовнішніх ключів для забезпечення суворої цілісності даних і гнучке управління політиками доступу. Також середовище спрощує перегляд структури таблиць, аналіз зв'язків між сутностями, виконання SQL-запитів і перевірку коректності внесених змін.

Застосування DataGrip є корисним не лише на етапі початкового проєктування, а й під час подальшого супроводу системи. У процесі розробки можуть виникати потреби в уточненні структури таблиць, додаванні нових полів, перевірці індексів, аналізі продуктивності запитів або пошуку помилок у даних. Наявність зручного інструмента для роботи з базою даних скорочує час на

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				31
Змн.	Арк.	№ докум.	Підпис	Дата		

виконання таких завдань і підвищує якість технічного контролю. Це важливо для системи, яка має розвиватися разом із потребами закладу вищої освіти.

На етапі концептуального проєктування сховища даних було розроблено діаграму класів, яка формалізує основні логічні одиниці предметної області та наочно демонструє ключові реляційні взаємозв'язки між компонентами системи. Така діаграма є важливим елементом проєктування, оскільки дозволяє ще до фізичної реалізації бази даних визначити, які сутності мають бути представлені в системі, які атрибути вони містять і як взаємодіють між собою. Це зменшує ризик помилок під час створення таблиць і допомагає забезпечити логічну завершеність інформаційної моделі.

Таким чином, вибір MySQL як основної системи управління базами даних є обґрунтованим з технічного, архітектурного та експлуатаційного погляду. Вона забезпечує достатню продуктивність, підтримує складні реляційні зв'язки, інтегрується з PHP та Yii2, дозволяє реалізувати контроль цілісності даних і створює стабільну основу для формування кадрової та статистичної звітності. У поєднанні з DataGrid ця СУБД дає змогу ефективно проєктувати, адмініструвати й підтримувати базу даних, що є необхідною умовою для стабільної роботи системи.

2.3. Розробка бази даних системи

Основою архітектури інформаційної системи закладу вищої освіти є надійна та структурована реляційна база даних MySQL. З огляду на значний обсяг різнорідних даних і складність бізнес-процесів, під час проєктування бази даних особливу увагу було приділено забезпеченню цілісності даних за допомогою зовнішніх ключів та оптимізації швидкодії складних аналітичних запитів засобами фреймворку Yii2.

Концептуально схема бази даних розділена на два масивні кластери пов'язаних таблиць, які обслуговують модулі «Кадрове забезпечення» та «Статистика та звіти».

		Кукса Р.О.			ПІЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		32

Проектування модуля «Кадрове забезпечення» базується на центральній сутності профілю працівника, навколо якої формується система пов'язаних таблиць. У межах кадрового обліку база даних зберігає відомості про зайнятість, посади, структурні підрозділи, ставки, погодинну оплату праці та табелі робочого часу. Таке структурування дає змогу формувати звітність для відділу кадрів і бухгалтерії без дублювання інформації в окремих файлах.

Життєвий цикл працевлаштування в системі відображено через таблиці конкурсів, поданих документів, контрактів і шаблонів контрактів. Це дозволяє поєднати процес конкурсного відбору, перевірку документів кандидатів і подальше укладання трудових угод у межах єдиної інформаційної моделі. Окремий сегмент бази даних призначено для зберігання документів НПП, що підтверджують освіту, наукові ступені, вчені звання, підвищення кваліфікації, участь у наукових проєктах, публікаційну активність, професійне визнання та роботу зі здобувачами освіти.

Така декомпозиція кадрових даних є важливою для автоматизованої перевірки виконання пункту 38 ліцензійних умов. Замість ручного збору підтверджувальних матеріалів система агрегує дані з пов'язаних таблиць і формує узагальнену картину щодо виконання кваліфікаційних вимог кожним викладачем.

Аналітичний блок бази даних охоплює таблиці, призначені для збирання, буферизації та узагальнення статистичної інформації. Дані про вступні кампанії, контингент студентів і його рух зберігаються у реляційному вигляді, що дозволяє будувати історичні звіти та відстежувати зміни показників у динаміці. Окремі структури бази даних оптимізовано для формування звітів щодо розкладу занять, захисту кваліфікаційних робіт, документів НПП і публікаційної активності працівників.

Використання індексів для полів, за якими найчастіше відбувається пошук та фільтрація (наприклад, ідентифікатори працівників, дати контрактів, статуси документів), дозволило значно прискорити час відгуку системи навіть при обробці десятків тисяч пов'язаних записів.

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		33

Під час розробки бази даних важливим етапом стало визначення основних сутностей предметної області та зв'язків між ними. Для кадрового напрямку ключовими сутностями є працівник, документ, публікація, запис про зайнятість, контракт, конкурс, підрозділ і посада. Для статистичного напрямку важливими є сутності, що описують вступну кампанію, контингент здобувачів освіти, рух студентів, освітні програми, спеціальності, кваліфікаційні роботи та розклад. Такий поділ дозволяє логічно розмежувати дані, але не ізолює їх повністю, оскільки статистичні звіти можуть використовувати кадрову інформацію та дані інших підсистем.

Особливу увагу було приділено нормалізації структури бази даних. Збереження довідникової інформації в окремих таблицях дозволяє уникнути дублювання значень і зменшити ризик появи суперечливих даних. Наприклад, назви кафедр, факультетів, посад, типів документів або статусів не повинні повторюватися у великій кількості записів. Натомість основні таблиці зберігають зовнішні ключі на відповідні довідники. Це спрощує оновлення інформації та забезпечує узгодженість даних у всіх частинах системи.

Для кадрового модуля важливим є збереження історичності частини даних. Інформація про зайнятість працівника, зміну посади, укладання контрактів або проходження конкурсного відбору не повинна просто перезаписуватися, оскільки вона може бути потрібною для звітності та перевірки попередніх періодів. Тому структура бази даних має передбачати збереження дат початку та завершення відповідних записів, а також статусів, які дозволяють відрізнити чинні записи від архівних. Такий підхід робить систему придатною не лише для поточного обліку, а й для аналізу змін у кадровому складі.

Для модуля статистики та звітів важливою є можливість формувати вибірки за різними параметрами. Саме тому під час проєктування таблиць слід передбачати поля, які дозволяють фільтрувати дані за роком, структурним підрозділом, спеціальністю, освітнім рівнем, статусом, типом події або періодом. Це значно спрощує побудову звітів, оскільки система може отримувати потрібні дані без додаткової ручної обробки. Крім того, правильно спроектована

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		34

структура таблиць дозволяє будувати не лише поточні звіти, а й аналітику в динаміці.

Важливим елементом бази даних є забезпечення цілісності зв'язків між таблицями. Зовнішні ключі дозволяють уникнути ситуацій, коли в системі залишається документ без пов'язаного працівника, запис про зайнятість без посади або статистичний показник без відповідного періоду. Такий контроль особливо важливий для адміністративних систем, де помилки в даних можуть впливати на офіційні звіти та управлінські рішення. Крім того, обмеження на рівні бази даних доповнюють перевірки, реалізовані в програмному коді.

Для підвищення надійності та прозорості роботи системи доцільним є використання службових полів у ключових таблицях. До таких полів належать дата створення запису, дата останнього оновлення, ідентифікатор користувача, який виконав зміну, а також статус активності запису. Наявність цих полів дозволяє відстежувати історію змін, спрощує аудит дій користувачів і допомагає виявляти причини можливих розбіжностей у даних. У кадровому модулі це має особливе значення, оскільки зміни в документах або статусах працівників повинні бути контрольованими.

Отже, розробка бази даних системи передбачає не лише створення набору таблиць, а й побудову цілісної інформаційної моделі, яка відображає реальні процеси закладу вищої освіти. Запропонована структура дозволяє зберігати кадрові, академічні та статистичні дані у взаємопов'язаному вигляді, забезпечує основу для швидкого пошуку, фільтрації та формування звітів, а також створює передумови для подальшого масштабування системи.

2.4. Реалізація модуля «Кадрове забезпечення»

Розробка модуля базується на архітектурному патерні MVC у межах фреймворку Yii2. Модуль охоплює десять взаємопов'язаних підсистем: облік працівників, записи про зайнятість, наукові публікації, документи працівників, виконання ліцензійних умов, погодинну оплату, статистику табелів, конкурси на посади, контракти та їхні шаблони. Оскільки базова взаємодія з базою даних

		Кукса Р.О.			ПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		35

реалізована стандартними механізмами ActiveRecord, основний акцент під час розробки зроблено на проєктуванні складної бізнес-логіки, оптимізації ресурсомістких операцій та інтеграції із зовнішніми державними реєстрами.

Окремий напрям реалізації кадрового модуля пов'язаний із бізнес-логікою обробки та модерації документів, оскільки саме цей процес забезпечує перевірку достовірності даних, які надалі використовуються для кадрового обліку та аналітичних звітів. Для уникнення дублювання коду в різних контролерах логіку роботи з фізичними файлами винесено до спеціалізованого сервісу TeacherDocumentsService. Алгоритм централізовано керує процесами генерації унікальних імен для файлів:

```
public static function saveFile($model, UploadedFile $file, string $uploadDir): array
{
    $uploadDir = dirname(Yii::getAlias('@app')) . $uploadDir;

    if (!file_exists($uploadDir)) {
        mkdir($uploadDir, 0777, true);
    }

    $uniqueId = uniqid();
    $modelPath = $model->person_id ?? 'file';
    $fileName = "{$modelPath}_{$uniqueId}.$file->extension";
    $filePath = $uploadDir . $fileName;

    if ($file->saveAs($filePath)) {
        return ['success' => true, 'filepath' => $filePath];
    }

    return ['success' => false];
}
```

Завантажені документи проходять обов'язкову модерацію співробітниками відділу кадрів. Для збереження швидкого відгуку інтерфейсу цей процес реалізовано через асинхронні запити. У разі виявлення невідповідності даних або нечіткої скан-копії адміністратор повертає документ на доопрацювання:

```
public function actionDisapprove($id): array
{
    Yii::$app->response->format = Response::FORMAT_JSON;
    $data = json_decode(Yii::$app->request->getRawBody(), true);

    $model = $this->findModel($id);
    $model->staff_document_status_id = 3;
    $model->moderated_at = date('Y-m-d H:i:s');
    $model->moderated_by = Yii::$app->user->identity->fio;
```

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				36
Змн.	Арк.	№ докум.	Підпис	Дата		

```

$model->message = $data['message'];

if ($model->save(false)) {
    return ['success' => true];
}

return ['success' => false, 'message' => 'Помилка збереження'];
}

```

Метод змінює статус документа, фіксує ідентифікатор модератора й час перевірки, а також зберігає текст причини відхилення - це формує прозорий зворотний зв'язок з користувачем і дає змогу НПП виправити конкретну помилку.

Найбільш ресурсомісткою частиною кадрового модуля є агрегація даних для перевірки виконання ліцензійних умов, адже система повинна аналізувати не окремий документ, а сукупність підтверджених досягнень працівника за кількома напрямками. Процес передбачає агрегацію даних з десятків пов'язаних таблиць за часовими критеріями. Для вирішення цього завдання розроблено клас ProfessionalAchievementBatchService.

Щоб уникнути багаторазових звернень до бази даних у циклах, сервіс виконує пакетне завантаження всіх необхідних масивів - публікацій, стажувань, сертифікатів та агрегує їх на рівні застосунку:

```

public function getAllDataBatch(): array
{
    $result = [];
    $trainings = $this->getBatchTrainings();
    $publications = $this->getBatchPublications();
    $practicalWorks = $this->getBatchPracticalWorks();

    $tree = $this->getAchievementsTree();

    foreach ($this->personIds as $personId) {
        $items = [];
        foreach ($tree as $item) {
            $items[] = $this->processAchievementItem(
                $item,
                $personId,
                [
                    'publications' => $publications[$personId] ?? [],
                    'practicalWorks' => $practicalWorks[$personId] ??
[]],
                );
        }
        $result[$personId] = ['achievements' => $items];
    }
}

```

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Морозов А.В.				37
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }
    return $result;
}

```

Метод processAchievementItem виконує роль диспетчера, викликаючи ізольовану логіку обчислення для кожного підпункту досягнень. Складність полягає в тому, що деякі критерії вимагають не просто підрахунку записів, а аналізу перетинів часових інтервалів. Зокрема, алгоритм розрахунку стажу практичної роботи за спеціальністю (пункт 20) обчислює сумарну кількість днів, що потрапляють у п'ятирічний період, ігноруючи дані поза його межами:

```

private function calcP20(array $practicalWorks): array
{
    $fiveYearsAgoTs = $this->fiveYearsAgo->getTimestamp();
    $todayTs = time();

    $calcDays = function ($items) use ($fiveYearsAgoTs, $todayTs) {
        $sum = 0;
        foreach ($items as $item) {
            $start = strtotime($item['date_start'] ?? '');
            $finish = strtotime($item['date_finish'] ?? '') ?:
$todayTs;

            if ($finish > $fiveYearsAgoTs) {
                $actualStart = max($start, $fiveYearsAgoTs);
                $actualFinish = min($finish, $todayTs);

                if ($actualFinish >= $actualStart) {
                    $sum += floor(($actualFinish - $actualStart) /
86400);
                }
            }
        }
        return $sum;
    };

    $approved = array_filter($practicalWorks, fn($p) =>
($p['staff_document_status_id'] ?? 0) == 2);
    $sumDaysApproved = $calcDays($approved);

    return [
        'isAgreed' => $sumDaysApproved >= 1825 ? true :
(count($practicalWorks) > 0 ? null : false),
        'staffPracticalWork' => array_values($practicalWorks),
    ];
}

```

Щоб уникнути повторного обчислення складних показників під час кожного відкриття сторінки, у модулі застосовано подієво-орієнтований підхід,

		Кукса Р.О.			ПІЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		38

за якого зміни в документах або досягненнях працівника автоматично ініціюють оновлення відповідного статусу.

```
public static function register(): void
{
    $events = [
        BaseActiveRecord::EVENT_AFTER_INSERT,
        BaseActiveRecord::EVENT_AFTER_UPDATE,
        BaseActiveRecord::EVENT_AFTER_DELETE,
    ];

    foreach ($events as $event) {
        Event::on(StaffTraining::class, $event, [self::class,
            'handleEventWithManyToOne']);
        Event::on(PublicationAuthor::class, $event, [self::class,
            'handleEventWithPublicationAuthor']);
        Event::on(StaffScientificDegree::class, $event, [self::class,
            'handleEventWithManyToOne']);
    }
}
```

Обробник події викликає метод `updateProfessionalAchievementStatus`, який ініціалізує оптимізований сервіс `ProfessionalAchievementIsAgreedService`. Сервіс перевіряє виконання кожного підпункту ліцензійних умов і повертає масив булевих значень:

```
public static function
updateProfessionalAchievementStatus(int|string|null $person_id): void
{
    if (!$person_id) {
        return;
    }

    $achievement =
ProfessionalAchievementStatus::findOne(['person_id' => $person_id]);

    if (!$achievement) {
        $achievement = new ProfessionalAchievementStatus();
        $achievement->person_id = $person_id;
    }

    $service = new
ProfessionalAchievementIsAgreedService($person_id);
    $data = $service->getAllIsAgreed();
    $achievement->is_completed = count(array_filter($data)) >= 4;
    $achievement->save(false);
}
```

Результат обчислення зберігається у кеш-таблиці `professional_achievement_status`, що дає змогу миттєво отримувати актуальну статистику без повторного виконання складних SQL-запитів. Для поглибленої

		Кукса Р.О.			ПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				39
Змн.	Арк.	№ докум.	Підпис	Дата		

аналітики та масового експорту до державних баз використовується ProfessionalAchievementBatchService, який агрегує повний перелік підтверджуючих документів.

Агреговані результати потребують інформативної візуалізації для співробітників відділу кадрів. У представленні реалізовано динамічну генерацію статусу виконання ліцензійних умов. Алгоритм перевіряє винятки, за яких вимоги до співробітника не застосовуються, а в іншому разі генерує мікро-дашборд:

```
$service = new
ProfessionalAchievementIsAgreedService($model->staff->person_id);
$completionList = $service->getAllIsAgreed();
$isCompleted = count(array_filter($completionList)) >= 4;

$cells = '';
for ($i = 0; $i <= 20; $i++) {
    if ($i > 15 && $i < 19) continue;

    $content = $i === 0 ? "ПК" : $i;
    $classList = "completion-table-cell" .
(is_null($completionList[$i]) ? " bg-half" : ($completionList[$i] ? "
bg-success" : ""));
    $cells .= "<div class='$classList'>$content</div>";
}

return $isCompleted
? "<span class='text-success'>Виконано</span><div
class='completion-table'>$cells</div>"
: "<span class='text-danger'>Не виконано</span><div
class='completion-table'>$cells</div>";
```

Підхід забезпечує наочну кольорову індикацію кожного з 20 підпунктів безпосередньо у загальній таблиці працівників.

Фінальним етапом обробки кадрових даних є забезпечення їх синхронізації з ЄДЕБО. Оскільки формування великих масивів даних може перевищувати ліміти часу виконання PHP-скрипта, експорт реалізовано з урахуванням можливості масової відправки. За цю логіку відповідає контролер ProfessionalAchievementController:

```
private function exportToEdebo(string $type, bool $all = false):
array
{
    Yii::$app->response->format = Response::FORMAT_JSON;

    try {
```

		Кукса Р.О.			ПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				40
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        if ($all) {
            $searchModel = new ProfessionalAchievementSearch();
            $dataProvider =
$searchModel->search(Yii::$app->request->queryParams);
            $dataProvider->setPagination(false);

            $models = $dataProvider->getModels();
            $personIds = array_filter(array_unique(array_map(fn($m)
=> $m->staff->person_id ?? null, $models)));
        } else {
            $personIds = [Yii::$app->request->get('personId')];
        }

        $exportData = $this->exportStaffAchievements($personIds,
$type);

        $service = new EdeboNodeService();
        $res = $service->sendTeacherToEdebo($exportData);

        return [
            'success' => true,
            'message' => is_scalar($res) ? (string)$res :
json_encode($res, JSON_UNESCAPED_UNICODE)
        ];

    } catch (\Throwable $e) {
        Yii::$app->response->setStatusCode(500);
        return ['success' => false, 'message' => $e->getMessage()];
    }
}

```

2.5. Реалізація модуля «Статистика та звіти»

Модуль «Статистика та звіти» є ключовим аналітичним інструментом системи, який забезпечує трансформацію сирих первинних даних у структуровану інформацію для прийняття управлінських рішень. Реалізація модуля базується на трирівневій моделі обробки: рівень сховища, рівень бізнес-логіки та рівень представлення.

Робота модуля статистики починається з визначення часових меж аналізу, оскільки різні види звітності потребують різного періоду охоплення історичних даних. Враховуючи великий обсяг історичних даних, що накопичувалися в університеті роками, виникла потреба в механізмі гнучкого налаштування «точки відліку» для кожного типу звітності. Для цього реалізовано сервіс `UniversitySettingsStatisticsService`, який взаємодіє з денормалізованою таблицею налаштувань.

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				41
Змн.	Арк.	№ докум.	Підпис	Дата		

Модель Statistics інкапсулює параметри для різних напрямків аналітики:

```
class Statistics extends ActiveRecord
{
    public static function tableName(): string
    {
        return 'university_settings_statistics';
    }

    public function rules(): array
    {
        return [
            [['start_year_admission_campaign',
            'start_year_contingent', 'start_year_contingent_movement'],
            'required'],
            [['start_year_admission_campaign',
            'start_year_contingent', 'start_year_contingent_movement'],
            'integer', 'min' => 2000, 'max' => 2099],
            [['top_specialties_limit'], 'integer', 'min' => 1]
        ];
    }
}
```

Використання статичного методу `getStatisticField` дозволяє будь-якому компоненту системи отримати актуальний рік початку розрахунків без необхідності повторного створення екземплярів об'єктів. Це забезпечує єдину точку конфігурації для всього аналітичного модуля.

Під час реалізації статистики вступної кампанії особливу увагу приділено коректній обробці спеціальностей і спеціалізацій, оскільки дані з першоджерел можуть мати неоднорідну структуру та потребують додаткового перетворення перед групуванням. Для забезпечення коректного групування на рівні бази даних у `SpecialitiesStatisticsRepository` реалізовано логіку парсингу рядків за допомогою SQL-функцій.

Особливий інтерес становить алгоритм визначення коду освітньої програми, який використовує пріоритет спеціалізації над загальною спеціальністю:

```
public function getBaseQuery(): Query
{
    return (new Query())
        ->select([
            'year' => new Expression('YEAR(eseh.date_begin)'),
            'qualification_group_name' =>
            'eseh.qualification_group_name',
            'speciality_code' => new
            Expression("SUBSTRING_INDEX(eseh.speciality_name, ' ', 1)"),
```

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Морозов А.В.				42
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        'education_code' => new Expression("
            COALESCE(
                NULLIF(SUBSTRING_INDEX(eseh.specialization_name,
' ', 1), ''),
                SUBSTRING_INDEX(eseh.speciality_name, ' ', 1)
            )
        "),
        'students_count' => new Expression('COUNT(*)'),
    ])
->from(['eseh' => '$tableHistories'])
->leftJoin(['eus' => '$tableStructurs'], 'eseh.faculty_id =
eus.id_university_facultet')
->where(['eseh.history_type_id' => 10])
->groupBy(['year', 'speciality_name', 'specialization_name']);
}

```

Функція SUBSTRING_INDEX виділяє першу частину рядка до першого пробілу, а COALESCE у поєднанні з NULLIF дозволяє динамічно підставляти код спеціалізації, якщо він заповнений. Це рішення дозволяє будувати точні звіти «ТОП-спеціальностей» навіть у випадках, коли структура даних у першоджерелі є неоднорідною.

Для аналізу контингенту студентів було реалізовано механізм формування історичних зрізів, який дає змогу відновити стан навчання здобувачів освіти на конкретні дати та використовувати ці дані для побудови динамічних звітів.

```

WITH RECURSIVE months AS (
    SELECT DATE('2015-01-01') AS month_date
    UNION ALL
    SELECT DATE_ADD(month_date, INTERVAL 1 MONTH)
    FROM months
    WHERE month_date < DATE_FORMAT(CURDATE(), '%Y-%m-01')
),
hist_intervals AS (
    SELECT eseh.*,
        LEAD(eseh.date_begin) OVER (
            PARTITION BY eseh.education_id
            ORDER BY eseh.date_begin
        ) AS next_date_begin
    FROM $tableHistories eseh
    WHERE eseh.qualification_group_id IN (1, 2, 7)
)
SELECT YEAR(m.month_date), MONTH(m.month_date), m.month_date,
    h.qualification_group_name, COUNT(*) ...
FROM months m
JOIN hist_intervals h
    ON m.month_date >= h.date_begin
    AND (h.next_date_begin IS NULL
        OR m.month_date < h.next_date_begin)
    AND h.history_type_id < 100

```

		Кукса Р.О.			ІІЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Морозов А.В.				43
Змн.	Арк.	№ докум.	Підпис	Дата		

```
GROUP BY m.month_date, h.qualification_group_name, ...;
```

Рекурсивна частина months генерує послідовність першого дня кожного місяця від 2015 року до поточного. Підзапит hist_intervals застосовує віконну функцію LEAD() з секцією PARTITION BY education_id: для кожного запису про зміну статусу визначається дата початку наступного запису тієї самої особи. Умова JOIN перевіряє, чи потрапляє місячна мітка в інтервал між date_begin поточного та next_date_begin наступного записів; значення NULL для next_date_begin означає останній відомий статус, тобто студент вважається активним. Фільтрація записів забезпечує відбір лише актуальних подій, необхідних для формування статистичних показників контингенту студентів. Отриманий результат зберігається в таблиці student_education_snapshots, яка надалі використовується як джерело даних для всіх аналітичних запитів.

Базовий запит для читання знімкової таблиці формується з урахуванням прав доступу поточного користувача:

```
public function getBaseQuery(): Query
{
    return (new Query())
        ->select([
            'snapshot_date',
            'COALESCE(university_facultet_short_name,
                faculty_name,
                "Не відомий факультет") as faculty',
            'SUM(students_count) as students_count'
        ])
        ->from('student_education_snapshots')
        ->where(['university_facultet_short_name'
            => $this->getAllowedFaculties()])
        ->andWhere(['>=', 'year', $this->startYearContingent])
        ->groupBy(['snapshot_date', 'faculty'])
        ->orderBy(['snapshot_date' => SORT_ASC]);
}
```

Функція COALESCE() послідовно перевіряє три варіанти назви факультету - це зумовлено неоднорідністю даних ЄДЕБО, де назва може зберігатися у різних полях залежно від типу запису. Метод getAllowedFaculties() трейту FacultyFilterTrait повертає перелік підрозділів, до яких має доступ поточний користувач відповідно до його ролі. Поле startYearContingent визначає нижню межу вибірки.

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		44

Сервіс StudentStatisticService перетворює сирі рядки запиту на три структури для Chart.js:

```
private function groupChartData(array $rowData): array
{
    foreach ($rowData as $row) {
        $date      = $row['snapshot_date'];
        $faculty   = $row['faculty'] ?? 'Інше';
        $count     = (int)$row['students_count'];

        $totalGrouped[$date] =
            ($totalGrouped[$date] ?? 0) + $count;
        $facultyDateMap[$faculty][$date] = $count;
    }
    // ...
    return [
        'total'      => [/* загальна динаміка */],
        'comparison' => [/* порівняння факультетів */],
        'individual' => $individualGrouped
    ];
}
```

Структура total містить одновимірний масив для лінійного графіка загальної чисельності. Структура comparison формує набір рядів datasets у форматі Chart.js - по одному на кожен факультет, що дає змогу накласти їх на спільний порівняльний графік. Структура individual зберігає незалежні масиви міток та значень для кожного факультету окремо. Масиви дат сортуються перед поверненням, гарантуючи хронологічний порядок по осі X незалежно від порядку рядків у запиті.

Статистика «Рух контингенту» відображає зведену статистику зарахувань та відрахувань по роках. Репозиторій StudentStatusRepository формує запит, що агрегує різні типи подій навчання в одному SELECT:

```
public function getBaseQuery(): Query
{
    return (new Query())
        ->select([
            'year' => new Expression('YEAR(eseh.date_begin)'),
            'total_enrollment' => new Expression(
                'SUM(eseh.history_type_id IN (10,20,21,13))'),
            'total_deduction' => new Expression(
                'SUM(eseh.history_type_id IN (110,100,111))'),
            'enrolled' => new Expression(
                'SUM(eseh.history_type_id = 10)'),
            'renewed_same_university' => new Expression(
                'SUM(eseh.history_type_id = 20)'),
            'expelled' => new Expression(
                'SUM(eseh.history_type_id = 110)'),
        ])
```

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				45
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        'graduated' => new Expression(
            'SUM(eseh.history_type_id = 100)'),
    ])
    ->from(['eseh' => $tableHistories])
    ->andWhere(['eseh.history_type_id' =>
[10,20,21,13,110,100,111]])
    ->groupBy(new Expression('YEAR(eseh.date_begin)'))
    ->orderBy(['year' => SORT_ASC]);
}

```

Конструкція SUM(вираз = значення) є специфічною для MySQL: булева рівність повертає значення 1 або 0, тому функція SUM() фактично підраховує кількість рядків, для яких виконується задана умова. Це дозволяє за один прохід по таблиці обчислити кілька статистичних показників без використання підзапитів або додаткових з'єднань. Розрахунки охоплюють основні події руху контингенту студентів, зокрема поновлення, зарахування, переведення, завершення навчання, відрахування та анулювання записів. Підсумкові поля агрегують відповідні групи подій, формуючи узагальнені показники поряд із деталізованою статистикою.

Метод `getFilterOptionsQuery()` репозиторію формує унікальні значення для фільтрів:

```

public function getFilterOptionsQuery(array $fields): Query
{
    $query = (new Query())
        ->select($fields)
        ->from('$tableHistories eseh')
        ->where([
            'eseh.qualification_group_id' => [1, 2, 7],
            'eseh.history_type_id' => [10,20,21,13,110,100,111],
        ])
        ->andWhere(['>=', 'eseh.date_begin',
            "{$this->startYearContingentMovement}-01-01"]])
        ->leftJoin(['eus' => '$tableStructures'],
            'eseh.faculty_id = eus.id_university_facultet')
        ->distinct();
    $this->applyFacultyFilter($query,
        'university_facultet_short_name');
    return $query;
}

```

Метод приймає масив полів і будує єдиний запит для отримання унікальних значень усіх фільтрів одночасно, уникаючи серії окремих запитів.

Сервіс `StudentStatusStatisticService` розбирає результат по полях:

```

public function getFilterOption(): array
{

```

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		46

```

        $fields = ['qualification_group_name', 'base_qualification_name',
                    'education_form_name',
                    'university_facultet_short_name'];
        $rows = $this->repository->getFilterOptionsQuery($fields)->all();
        $result = array_fill_keys($fields, []);
        foreach ($rows as $row) {
            foreach ($fields as $field) {
                $value = $row[$field] ?? null;
                if ($value !== null && $value !== '') {
                    $result[$field][$value] = $value;
                }
            }
        }
        return $result;
    }
}

```

Функція `array_fill_keys()` ініціалізує масив результату з порожніми масивами для кожного поля, що гарантує наявність усіх ключів навіть якщо для певного поля не знайдено жодного значення. Ітерація по рядках та полях одночасно виконується без групування на рівні SQL - рядки можуть містити різні комбінації значень, і цикл розподіляє їх по відповідних масивах. Перевірка на цілісність даних виключає незаповнені рядки, які виникають при відсутності даних в окремих записах ЄДЕБО.

Статистики документів НПП побудована на патерні `Template Method`. Абстрактний клас `BaseStaffStatisticService` визначає незмінний алгоритм формування відповіді через публічні методи `getSummaryPayload()` та `getBreakdownsPayload()`, залишаючи конкретні вибірки нащадкам через абстрактні методи:

```

abstract class BaseStaffStatisticService
{
    public function getSummaryPayload(): array
    {
        return [
            'selectedTab' => $this->getSelectedTabName(),
            'tabs'        => StaffDocumentsEnum::getLabels(),
            'metrics'     => $this->formatSummary(
                            $this->fetchSummaryData()),
            'availableYears' => $this->formatAvailableYears(
                            $this->fetchAvailableYears()),
        ];
    }

    abstract protected function fetchSummaryData(): array;
    abstract protected function fetchAvailableYears(): array;
    abstract protected function fetchBreakdownsData(?string $year):
array;

```

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Морозов А.В.				47
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        abstract protected function getBreakdownsConfig(): array;
    }

```

Клас-нащадок, наприклад `StaffAcademicDegreeTabService` для статистики вченого звання, реалізує лише чотири методи, вказуючи назву таблиці та датове поле. Вся логіка форматування залишається у базовому класі:

```

class StaffAcademicDegreeTabService extends BaseStaffStatisticService
{
    protected function fetchSummaryData(): array
    {
        return $this->repository
            ->getGeneralTabStatisticsQuery('academic_degree')
            ->all();
    }

    protected function fetchBreakdownsData(?string $year): array
    {
        $query = $this->repository->getCategoryTabStatisticsQuery(
            'academic_degree',
            'dictionary_academic_degree',
            'academic_degree_id', 'name'
        );
        $this->applyYearFilter($query, 'diploma_issue_date', $year);
        return [$query->all()];
    }

    protected function getBreakdownsConfig(): array
    {
        return [['title' => 'Розподіл за типом',
            'columnName' => 'Тип / Категорія']];
    }
}

```

Передача назви таблиці рядком дозволяє використовувати один клас `GeneralStaffStatisticsRepository` для всіх дев'ятнадцяти типів документів без дублювання коду. Метод `getCategoryTabStatisticsQuery()` приймає чотири параметри: назву таблиці документів, назву пов'язаного словника, зовнішній ключ та поле для відображення — цього достатньо для побудови будь-якого запиту з розбивкою по категоріях.

Базовий клас форматує список доступних років для фільтрації таким чином, щоб архівні дані не перевантажували інтерфейс:

```

protected function formatAvailableYears(array $rawYears): array
{
    $currentYear      = (int)date('Y');
    $oldestSpecific   = $currentYear - 5;
    $lessOrEqualYear  = $oldestSpecific - 1;

    $formattedYears = ['all' => 'Yci'];
}

```

		Кукса Р.О.			ІІЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Морозов А.В.				48
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    for ($i = 0; $i <= 5; $i++) {
        $formattedYears[$currentYear - $i] = (string)($currentYear -
    $i);
    }

    $hasOlderData = false;
    foreach ($rawYears as $year) {
        if (is_numeric($year) && $year <= $lessOrEqualYear) {
            $hasOlderData = true; break;
        }
    }
    if ($hasOlderData) {
        $formattedYears['<= ' . $lessOrEqualYear] =
            '<= ' . $lessOrEqualYear;
    }
    return $formattedYears;
}

```

Метод завжди відображає поточний рік та п'ять попередніх як окремі пункти. Якщо в базі даних є записи старші шести років — вони групуються в один пункт «<= рік». Це запобігає розростанню списку фільтрів до десятків позицій при наявності великого архіву документів. Фільтрація за таким спеціальним рядком обробляється методом `applyYearFilter()`:

```

protected function applyYearFilter(
    Query $query, string $dateField, ?string $year): void
{
    if (empty($year) || $year === 'all') { return; }
    if (str_starts_with($year, '<=')) {
        $yearValue = (int)trim(str_replace('<=', '', $year));
        $query->andFilterWhere(
            ['<=', "YEAR($dateField)", $yearValue]);
    } else {
        $query->andFilterWhere(
            ['=', "YEAR($dateField)", (int)$year]);
    }
}

```

Перевірка `str_starts_with()` визначає формат значення фільтра та буде відповідну умову SQL. Виклик `andFilterWhere()` на відміну від `andWhere()` автоматично ігнорує порожні значення, що захищає від випадкового фільтрування при `null`-значеннях. Зведена статистика по всіх типах документів одночасно отримується запитом з `UNION ALL`, що об'єднує вибірки з дев'ятнадцяти таблиць в один результат без дублювання.

Для оптимізації швидкодії клієнтської частини застосовано технологію PJAX. Зміна параметрів фільтрації ініціює фоновий запит без повного перезавантаження сторінки:

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				49
Змн.	Арк.	№ докум.	Підпис	Дата		

```
function reloadStatistic() {
    const url = new URL(window.location.href);
    url.search = $('#filter-form').serialize();

    const $statisticContainer = $('#pjax-statistic-container');
    createLoader($statisticContainer[0], { replace: false }).show();

    $.pjax.reload({
        container: "#pjax-statistic-container",
        url: url.toString(),
        timeout: 0,
    });
}
```

Алгоритм серіалізує дані форми, оновлює URL-адресу для збереження історії навігації та отримує від сервера лише оновлений HTML-контейнер. Це суттєво зменшує обсяг переданого трафіку та навантаження на сервер рендерингу.

Для наочного представлення показників інтегровано бібліотеку Chart.js. Кожен тип графіка (лінійний, гістограма, кільцевий) винесено в окремий модульний РНР-компонент. Рендеринг графіків на сторінках відбувається динамічно: необхідні набори даних та конфігураційні параметри передаються компоненту як аргументи, що дозволяє стандартизувати виклик графіків у системі:

```
<?= $this->render('../_shared/_line-chart', [
    'labels' => $years,
    'datasets' => [
        [
            "data" => array_column($data, 'total_enrollment'),
            "borderColor" => "#28a745",
            "fill" => true,
        ],
    ],
]) ?>
```

Усередині компонент автоматично ініціалізує екземпляр Chart.js, трансліюючи РНР-масиви у JS-об'єкт через Json::encode(). Такий підхід забезпечує єдиний стиль візуалізації та швидке додавання нових датасетів.

Генерацію масивних звітів у форматі Excel делеговано на клієнтський рівень за допомогою бібліотеки SheetJS. Це архітектурне рішення усуває ризик переповнення оперативної пам'яті сервера при експорті десятків тисяч записів.

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		50

Алгоритм парсить відфільтровані дані безпосередньо з віртуального DOM-дерева HTML-таблиці, очищуючи їх від несумісних веб-елементів:

```
const rows = [];  
table.querySelectorAll('tbody tr').forEach(tr => {  
  const row = [];  
  tr.querySelectorAll('td').forEach(td => {  
    const tmp = document.createElement('div');  
    tmp.innerHTML = td.innerHTML;  
    tmp.querySelectorAll('i, svg').forEach(el => el.remove());  
  
    const text = tmp.textContent.trim();  
    if (text.endsWith('%')) {  
      row.push(text);  
    } else {  
      const numeric = parseFloat(text.replace(/\s/g,  
'').replace(',', ' '));  
      row.push(!isNaN(numeric) && text !== ' ' ? numeric :  
text);  
    }  
  });  
  rows.push(row);  
});
```

Ключовим етапом парсингу є приведення типів даних. Регулярні вирази очищують числові показники та форматують їх у тип Float. Завдяки цьому згенерований Excel-документ розпізнає їх як числа, що дозволяє користувачам одразу застосовувати до них математичні формули без ручної конвертації комірок. Після парсингу скрипт задає базові параметри друку та ініціює завантаження готового бінарного файлу.

Висновки до другого розділу

У межах другого розділу було проведено комплексне проектування системи автоматизації управління освітніми процесами, під час якого сформовано детальний перелік функціональних та нефункціональних вимог. Визначені цілі розробки орієнтовані на повну цифровізацію ключових адміністративних та аналітичних завдань закладу вищої освіти. Запропонована архітектурна концепція забезпечує необхідну гнучкість, масштабованість та високу швидкість обробки інформації, що є критично важливим при роботі з великими обсягами даних у реальному часі.

На основі аналізу предметної області створено логічну інформаційну модель, яка охоплює всі значущі об'єкти системи та їхні взаємозв'язки.

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				51
Змн.	Арк.	№ докум.	Підпис	Дата		

Проектування структури сховища даних було спрямовано на забезпечення цілісності інформації, усунення її дублювання та оптимізацію доступу до звітних показників. Впроваджені методи розмежування прав доступу та механізми ведення системних журналів гарантують високий рівень безпеки конфіденційних даних та надійність їх зберігання.

Практична реалізація охоплює повний цикл автоматизації: від ведення деталізованих профілів персоналу до інтелектуального аналізу динаміки контингенту здобувачів освіти. Розроблені сервісні компоненти дозволяють автоматично контролювати відповідність кадрового складу нормативним критеріям, мінімізуючи потребу в ручному опрацюванні документів. Завдяки впровадженню інтерактивних засобів візуалізації та ефективних механізмів формування звітності у популярних офісних форматах, користувачі отримали потужний інструмент для прийняття стратегічних управлінських рішень. Таким чином, спроєктований програмний продукт повністю відповідає поставленим завданням та є ефективним рішенням для цифрової трансформації діяльності університету.

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		52

РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З СИСТЕМОЮ ОНЛАЙН-МЕНЕДЖМЕНТУ ОСВІТНІХ ПРОЦЕСІВ

3.1. Порядок встановлення та налаштування параметрів додатку

Перед запуском проєкту на реальних машинах було розроблено діаграму розгортання [27], що наочно демонструє зв'язки між різними компонентами програмної системи (рис. 3.1):

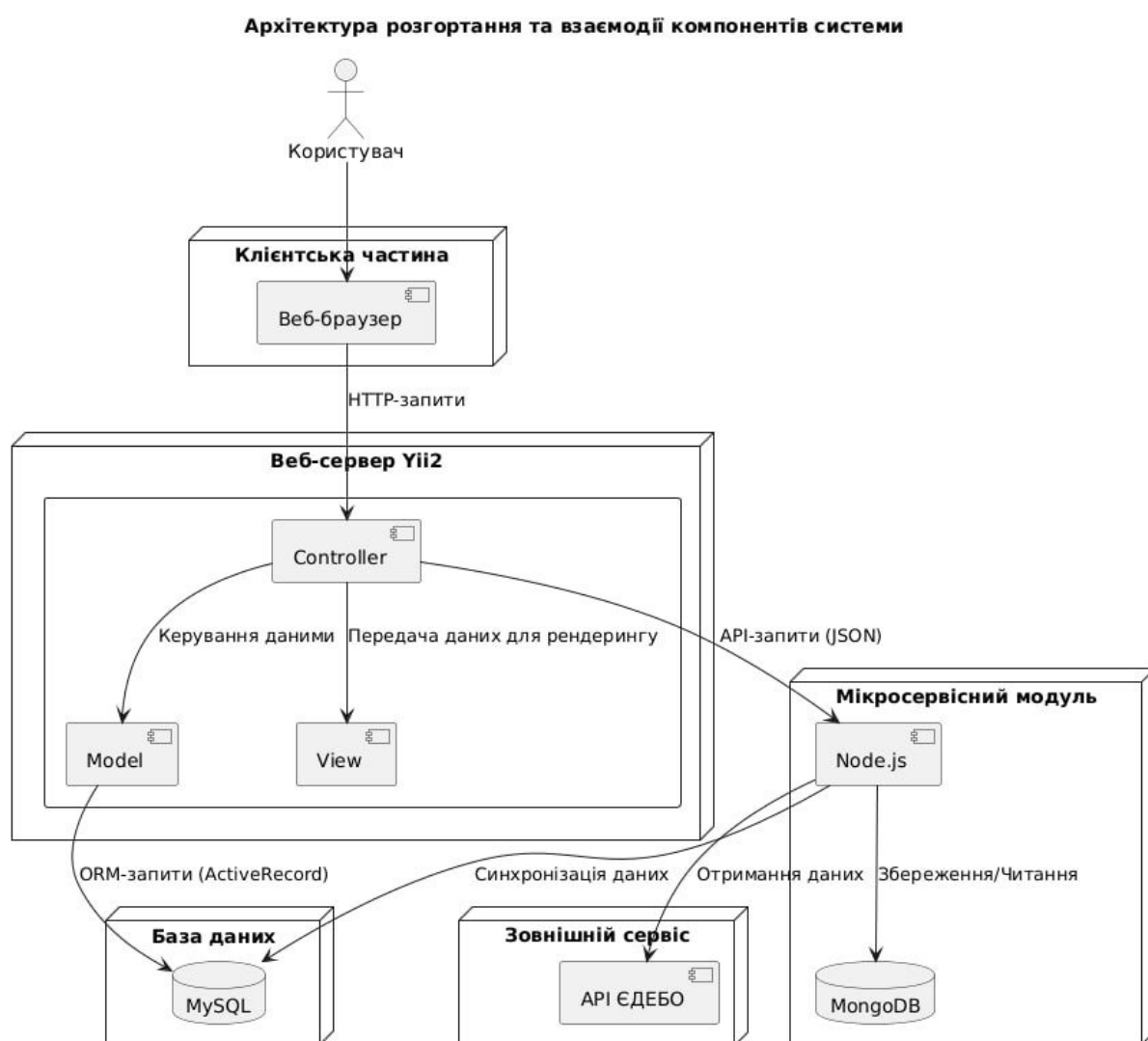


Рисунок 3.1 - Діаграма розгортання додатку

Схема відображає структурні особливості комплексу, орієнтованого на автоматизацію освітнього менеджменту. В основу архітектури покладено комбінований принцип: базовий функціонал функціонує у вигляді монолітного

Yii2-додатка із генерацією інтерфейсу на стороні сервера, тоді як супутні завдання покладені на окремий мікросервіс, створений за допомогою Node.js. Усі модулі розподілені по автономних фізичних та логічних вузлах.

Взаємодія кінцевого користувача із платформою відбувається через стандартний веббраузер. Направлені запити надходять до серверного вузла, де розгорнуте середовище Yii2 обробляє їх відповідно до концепції MVC.

До ключових елементів вебсервера належать контролери, моделі та представлення, які разом забезпечують реалізацію архітектурного підходу MVC. Контролери приймають вхідні HTTP-запити, визначають подальший порядок їх обробки та координують роботу допоміжних служб. Моделі зосереджують правила бізнес-логіки та забезпечують взаємодію з реляційним сховищем MySQL через інструментарій Active Record ORM. Представлення формують кінцевий HTML-код, який відображається у браузері користувача на основі даних, переданих контролерами.

Головний вебпортал має пряме з'єднання з базою даних MySQL. У цьому сховищі акумулюються відомості про студентський контингент, викладацький склад, навчальні плани, а також показники успішності.

Паралельно працює додатковий мікросервіс на базі Node.js. Його призначення полягає у виконанні таких процедур:

- Робота з неструктурованими чи тимчасовими масивами даних через підключення до бази MongoDB;
- Комунікація із зовнішнім державним шлюзом ЄДЕБО для верифікації та імпорту та експорту даних відомостей про вступників, студентів та викладацький склад;
- Підтримка актуальності даних шляхом синхронізації з головною базою MySQL.

Обмін інформацією між сервісами побудований на використанні API-запитів із застосуванням формату JSON. Такий підхід закладає підґрунтя для подальшого масштабування системи, розподілу обчислювальних потужностей або майбутнього відокремлення функціональних блоків.

		Кукса Р.О.			ПІЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		54

Технічне розгортання платформи здійснюється на серверних потужностях, які задовольняють вимоги щодо відмововитривалості та швидкодії. Повноцінна робота вебзастосунку підтримується у всіх поширених браузерах (зокрема Google Chrome версії 45 і вище, Mozilla Firefox від версії 38, а також Safari та Microsoft Edge). Це забезпечує стабільну кросбраузерну роботу [29] на комп'ютерах і мобільних гаджетах без потреби інсталивати будь-який локальний софт. Описана конфігурація дозволяє зберегти гнучкість архітектури, спрощує її подальший розвиток та технічну підтримку.

Процес встановлення системи передбачає послідовну підготовку серверного середовища, розгортання програмного коду, налаштування підключення до бази даних і перевірку взаємодії всіх основних компонентів. На першому етапі адміністратор повинен підготувати сервер, встановити необхідні компоненти середовища виконання, налаштувати вебсервер і забезпечити доступ до СУБД MySQL. Після цього розгортається кодова база застосунку, встановлюються залежності, виконуються міграції бази даних і задаються параметри підключення до зовнішніх сервісів.

Важливою умовою коректного запуску є налаштування конфігураційних файлів. У них визначаються параметри підключення до бази даних, адреси сервісів, ключі доступу, параметри поштових повідомлень, режим роботи застосунку та інші службові налаштування. Зберігання таких параметрів окремо від основного програмного коду є доцільним, оскільки дозволяє використовувати одну кодову базу для різних середовищ: локального, тестового та робочого. Це також підвищує безпеку, адже конфіденційні дані не повинні безпосередньо зберігатися у відкритих частинах проєкту.

Після налаштування середовища необхідно виконати міграції бази даних, які створюють потрібні таблиці, зв'язки, індекси та початкові довідникові записи. Механізм міграцій дозволяє уникнути ручного створення структури бази даних і зменшує ризик помилок під час розгортання. Якщо система переноситься на інший сервер або оновлюється до нової версії, міграції допомагають привести

		Кукса Р.О.			ПІЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		55

структуру бази даних до актуального стану без повного ручного втручання адміністратора.

Після завершення основного налаштування важливо перевірити коректність роботи системи в умовах, наближених до реальної експлуатації. Для цього необхідно протестувати відкриття основних сторінок, авторизацію користувачів, доступ до модулів відповідно до ролей, збереження форм, завантаження документів і формування звітів. Така перевірка дозволяє переконатися, що програмний продукт не лише успішно розгорнутий на сервері, а й готовий до практичного використання працівниками закладу вищої освіти.

Окрему увагу під час налаштування слід приділити параметрам доступу та захисту даних. Оскільки система працює з кадровою, академічною та статистичною інформацією, кожна категорія користувачів повинна мати доступ лише до тих функцій, які відповідають її повноваженням. Правильне налаштування ролей і прав доступу зменшує ризик несанкціонованого перегляду або зміни службової інформації, а також забезпечує більш контрольовану роботу з документами, профілями працівників і звітними даними.

Таким чином, встановлення та налаштування системи охоплює не лише розміщення програмного коду на сервері, а й підготовку середовища, перевірку підключення до бази даних, налаштування конфігураційних параметрів, контроль доступу та тестування основних функцій. Дотримання цих етапів забезпечує стабільний запуск інформаційної системи та створює умови для її подальшої експлуатації в адміністративному середовищі закладу вищої освіти.

3.2. Структура інтерфейсу додатку

Створена система онлайн-керування освітніми процесами має інтуїтивно зрозумілий користувацький інтерфейс і повністю адаптована для роботи на різних типах пристроїв.

Вхід до системи є повністю персоналізованим. Щоб потрапити в робочий простір, користувачеві необхідно заповнити стандартну форму авторизації (рис. 3.2). Після перевірки введених даних система відкриває доступ саме до тих

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		56

інструментів та розділів, які відповідають посадовим обов’язкам або правам конкретного співробітника

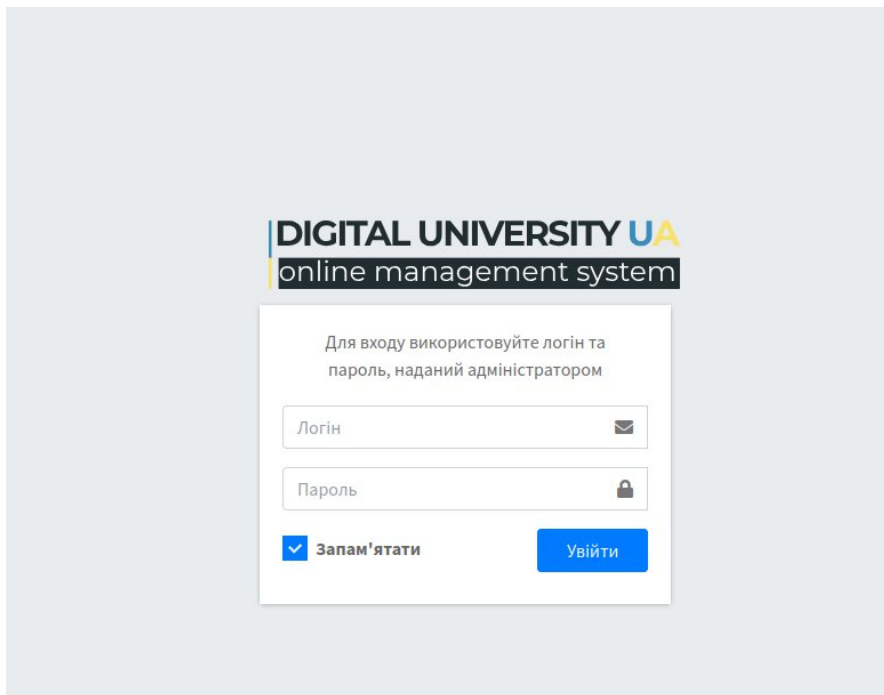


Рисунок 3.2 – Інтерфейс авторизації користувача

Для захисту внутрішньої інформації ЗВО від стороннього втручання, відкрита реєстрація на сайті заблокована. Створенням та видачею облікових записів займається виключно відповідальний підрозділ університету. Такий підхід до формування бази користувачів дозволяє повністю контролювати доступ до внутрішнього середовища університету та нівелює ризик появи фейкових чи несанкціонованих профілів.

Після успішної автентифікації користувача система автоматично відкриває головну сторінку платформи. Структура робочого простору спроектована так, щоб мінімізувати кількість кліків для пошуку потрібної інформації. Всі доступні інструменти, модулі та аналітичні блоки розробники об'єднали в єдине вертикальне меню навігації, яке зафіксоване в лівій частині робочого столу (рис. 3.3).

		Кукса Р.О.			ПІЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		57

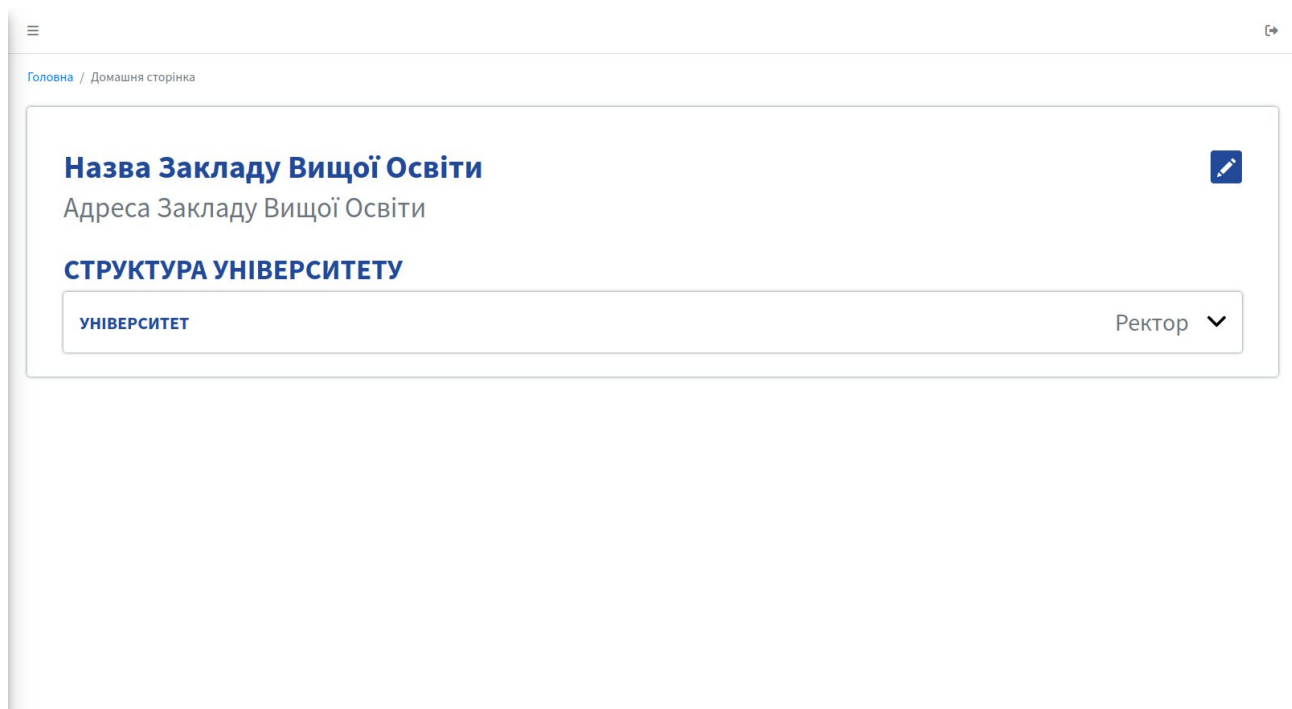


Рисунок 3.3 - Головна сторінка

У межах розробленої платформи ключова увага приділяється взаємодії з двома модулями – «Кадрове забезпечення» та «Статистика та звіти». Саме ці компоненти відповідають за накопичення, систематизацію та аналітичну обробку інформації про викладацький склад університету та рух студентського контингенту. Головним елементом навігації для переходу до робочих областей цих модулів є ліва панель керування (рис. 3.4 та рис. 3.5)

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		58

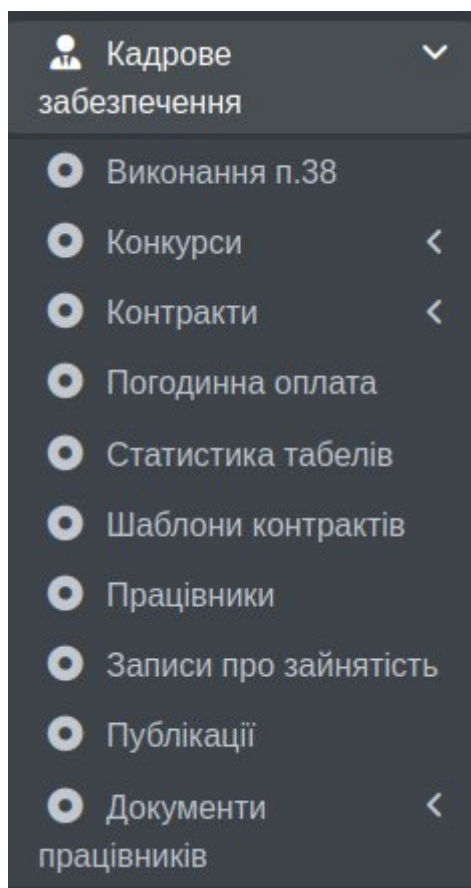


Рисунок 3.4 – Кадрове забезпечення

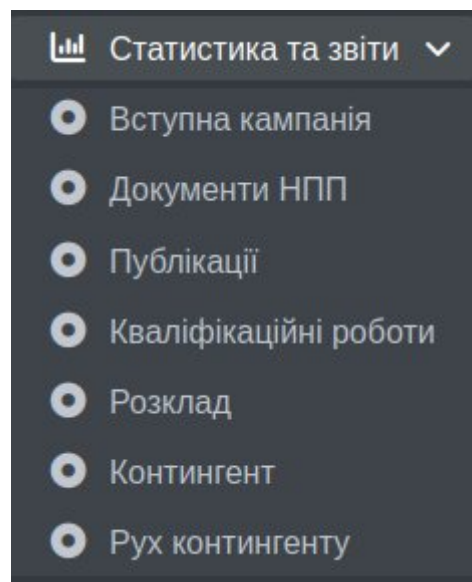


Рисунок 3.5 – Статистика та звіти

Подальший огляд функціональних можливостей інтерфейсу виконуватиметься з позиції користувача, який володіє правами адміністратора. Такий підхід обумовлений тим, що користувачі з правами адміністратора мають доступ до розширеного функціоналу системи, зокрема до засобів модерування, редагування даних, управління кадровим складом підрозділів, верифікації документів, налаштування параметрів звітності та запуску ресурсоємних аналітичних процесів.

Після опису загальної структури робочого простору доцільно перейти до інтерфейсу модуля «Кадрове забезпечення», оскільки саме він забезпечує основну взаємодію з кадровими даними, документами працівників і засобами перевірки виконання ліцензійних вимог.

Основне вікно цього блоку – сторінка «Працівники» (рис. 3.6), яка містить повний список викладачів та персоналу університету. Кнопка «+» вгорі дозволяє швидко створити нову картку, а за допомогою полів під заголовками колонок можна миттєво відфільтрувати список, наприклад, за ПІБ чи статусом

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				59
Змн.	Арк.	№ докум.	Підпис	Дата		

активності. У першій колонці виведені кнопки для швидкого редагування, видалення або переходу до документів конкретної особи. Окремо реалізовано стовпець «Записи про зайнятість», де через іконку ока можна відразу переглянути трудову історію працівника.

Головна / Кадрове забезпечення / Працівники

Працівники +
Показані 1-15 із 1 957 записів.

#	ІД фізичної особи	ПІБ	Записи про зайнятість	Стать	Дата народження	Чи активний	ПІН	Логін користувача
						Активні		
1		1234567890	Працівник -1				11111100-1	
2		1432662880	Працівник 1567		Жіноча		1111111567	
3		1844068834	Працівник 1568		Чоловіча		1111111568	
4		1352038614	Працівник 1569		Жіноча		1111111569	
5		1517112492	Працівник 1570		Жіноча		1111111570	
6		1764202381	Працівник 1571		Жіноча		1111111571	
7		1479849058	Працівник 1572		Чоловіча		1111111572	
8		1573990223	Працівник 1573		Чоловіча		1111111573	
9		1517070444	Працівник 1574		Жіноча		1111111574	
10		1917259583	Працівник 1575		Чоловіча		1111111575	
11		1516895463	Працівник 1576		Чоловіча		1111111576	
12		1262237007	Працівник 1577		Жіноча		1111111577	
13		1559374966	Працівник 1578		Чоловіча		1111111578	
14		1519200438	Працівник 1579		Жіноча		1111111579	
15		1723514225	Працівник 1580		Жіноча		1111111580	

« 1 2 3 4 5 »

Рисунок 3.6 – Інтерфейс реєстру працівників

Для детального моніторингу умов праці та розподілу навантаження використовується сторінка «Записи про зайнятість» (рис. 3.7). Тут у вигляді таблиці виводяться посади, структурні підрозділи, типи працевлаштувань, а також точний обсяг ставки та годин. Над самою таблицею є випадаючий список для швидкого відсіювання лише тих, хто працює на даний момент. Кожен стовпець має власне поле пошуку та кнопку швидкого скидання фільтра, що спрощує роботу з великою базою даних.

Головна / Кадрове забезпечення / Записи про зайнятість

Записи про зайнятість

Показані 1-15 із 1 904 записів.

Працюють												
#		П.І.Б.	Дата народження	ID фізичної особи	Чи зовнішній сумісник	Посада	Тип працевлаштування	Ставка	К-сть годин	Структурний підрозділ	Дата початку роботи	Дата завершення роботи
1		Працівник 3488		1234567890		Викладач	Сумісництво	0.1		Кафедра біоінформатики та аерокосмічної медицини	02.02.2026	
2		Працівник 3555	26.01.1988	1522740762		Викладач	Сумісництво	0.1		Кафедра біоінформатики та аерокосмічної медицини	02.02.2026	
3		Працівник 3487		1234567890		Професор	Сумісництво	0.1		Кафедра біоінформатики та аерокосмічної медицини	02.02.2026	
4		Працівник 3601		1234567890		Асистент викладача	Сумісництво	0.1		Кафедра логістики	26.01.2026	30.06.2026
5		Працівник 2499	26.04.1956	1808782878		Асистент викладача	Штатна посада	0.1	40	Кафедра аеродинаміки та безпеки польотів літальних апаратів	01.07.2025	
6		Працівник 1803	13.08.1999	1622571054		Асистент викладача	Сумісництво	0.1		Кафедра логістики	26.01.2026	30.06.2026
7		Працівник 3322	15.06.1982	1523864970		Асистент викладача	Сумісництво	0.1		Кафедра логістики	26.01.2026	30.06.2026
8		Працівник 2084	15.07.1964	1523649482		Професор	Сумісництво	0.1	120	Кафедра інтелектуальних кібернетичних систем	19.01.2026	30.06.2026
9		Працівник 1839	22.03.1970	1521728442		Професор	Сумісництво	0.1		Кафедра теорії держави і права та конституціоналізму	19.01.2026	30.06.2026
10		Працівник 2310	17.09.1978	1520994362		Доцент	Сумісництво	0.1		Кафедра маркетингу	04.02.2026	30.06.2026
11		Працівник 2314	01.05.1989	1520981264		Професор	Сумісництво	0.1		Кафедра маркетингу	19.01.2026	30.06.2026
12		Працівник 2308	14.10.1990	1365706398		Завідувач кафедри	Штатна посада	0.1	276	Кафедра маркетингу	22.12.2025	30.06.2026
13		Працівник 1720	25.11.1981	1521010716		Доцент	Сумісництво	0.1		Кафедра маркетингу	19.01.2026	
14		Працівник 3600		1234567890		Асистент	Штатна посада	0.1	55	Кафедра телекомунікаційних та радіоелектронних систем	19.01.2026	30.06.2026
15		Працівник 3597		1234567890		Доцент	Штатна посада	0.1		Кафедра англійської філології і перекладу	19.01.2026	30.06.2026

« 1 2 3 4 5 »

Рисунок 3.7. – Інтерфейс записів працівників про зайнятість

Для накопичення та систематизації даних про наукову діяльність викладачів у системі функціонує сторінка «Публікації» (рис. 3.8):

Головна / Кадрове забезпечення / Публікації

Публікації

Показані 1-20 із 13 564 записів.

Експорт

Всі статуси

#		Тип публікації	Підрозділ	Бібліографічний опис	Дата публікації	Фахове
1		Мистецький твір (живопис, архітектура, дизайн, фото, скульптура, графіка)	Кафедра авіаційної психології	Прізвище І.М., Прізвище І.М., Прізвище І.М. Тема публікації. Предмет. Предмет. Предмет. Журнал. Стаття. К. 2020. №123/321 р., С. 10-15	01.04.2024	
2		Мистецький твір (живопис, архітектура, дизайн, фото, скульптура, графіка)	Кафедра фінансів, обліку та оподаткування	Прізвище І.М., Прізвище І.М., Прізвище І.М. Тема публікації. Предмет. Предмет. Предмет. Журнал. Стаття. К. 2020. №123/321 р., С. 10-15	20.02.2022	
3		Мистецький твір (живопис, архітектура, дизайн, фото, скульптура, графіка)	Кафедра фінансів, обліку та оподаткування	Прізвище І.М., Прізвище І.М., Прізвище І.М. Тема публікації. Предмет. Предмет. Предмет. Журнал. Стаття. К. 2020. №123/321 р., С. 10-15	20.01.2022	
4		Мистецький твір (живопис, архітектура, дизайн, фото, скульптура, графіка)	Кафедра фінансів, обліку та оподаткування	Прізвище І.М., Прізвище І.М., Прізвище І.М. Тема публікації. Предмет. Предмет. Предмет. Журнал. Стаття. К. 2020. №123/321 р., С. 10-15	16.09.2020	
5		Мистецький твір (живопис, архітектура, дизайн, фото, скульптура, графіка)	Кафедра фінансів, обліку та оподаткування	Прізвище І.М., Прізвище І.М., Прізвище І.М. Тема публікації. Предмет. Предмет. Предмет. Журнал. Стаття. К. 2020. №123/321 р., С. 10-15	20.08.2022	
6		Мистецький твір (живопис, архітектура, дизайн, фото, скульптура, графіка)	Кафедра фінансів, обліку та оподаткування	Прізвище І.М., Прізвище І.М., Прізвище І.М. Тема публікації. Предмет. Предмет. Предмет. Журнал. Стаття. К. 2020. №123/321 р., С. 10-15	10.02.2022	
7		Мистецький твір (живопис, архітектура, дизайн, фото, скульптура, графіка)	Кафедра фінансів, обліку та оподаткування	Прізвище І.М., Прізвище І.М., Прізвище І.М. Тема публікації. Предмет. Предмет. Предмет. Журнал. Стаття. К. 2020. №123/321 р., С. 10-15	20.04.2021	
8		Стаття	Кафедра фінансів, обліку та оподаткування	Прізвище І.М., Прізвище І.М., Прізвище І.М. Тема публікації. Предмет. Предмет. Предмет. Журнал. Стаття. К. 2020. №123/321 р., С. 10-15	10.09.2020	
9		Стаття	Кафедра фінансів, обліку та оподаткування	Прізвище І.М., Прізвище І.М., Прізвище І.М. Тема публікації. Предмет. Предмет. Предмет. Журнал. Стаття. К. 2020. №123/321 р., С. 10-15	14.10.2020	
10		Патент на корисну модель	Кафедра логістики	Прізвище І.М., Прізвище І.М., Прізвище І.М. Тема публікації. Предмет. Предмет. Предмет. Журнал. Стаття. К. 2020. №123/321 р., С. 10-15	18.12.2024	

Рисунок 3.8 – Інтерфейс реєстру наукових публікацій працівників

Цей інтерфейс відображає параметри наукових праць, які автори самостійно заносять до бази даних. У таблиці реєстру виводяться відомості про тип публікації, бібліографічний опис, дату видання, а також індексацію в міжнародних наукометричних базах Scopus чи Web of Science. Крім перегляду

		Кукса Р.О.			ПІЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				61
Змн.	Арк.	№ докум.	Підпис	Дата		

супровідних даних, мови та країни видання, інтерфейс відображає пряме електронне посилання на джерело і повний перелік співавторів.

Подальший облік кваліфікаційних документів здійснюється через блок «Документи працівників», для прикладу було взято сторінку «Підвищення кваліфікації викладачів» (рис. 3.9):

Головна / Документи працівників / Підвищення кваліфікації викладачів

Підвищення кваліфікації викладачів +

Показані 1-20 із 2 922 записів.

Усі записи

#	Дії	Документ	Статус документів	ПІБ	Місце проходження (організація)	Тема підвищення кваліфікації	Вид документа про підвищення кваліфікації
1				Працівник 1905	Центр підвищення кваліфікації, професійного розвитку освітньої діяльності науково-педагогічних працівників	Сучасні методи дистанційного навчання	довідка
2				Працівник 1927	Центр підвищення кваліфікації, професійного розвитку освітньої діяльності науково-педагогічних працівників	Сучасні методи дистанційного навчання	сертифікат
3				Працівник 1927	Центр підвищення кваліфікації, професійного розвитку освітньої діяльності науково-педагогічних працівників	Сучасні методи дистанційного навчання	сертифікат
4				Працівник 2742	Центр підвищення кваліфікації, професійного розвитку освітньої діяльності науково-педагогічних працівників	Сучасні методи дистанційного навчання	сертифікат
5				Працівник 2716	Центр підвищення кваліфікації, професійного розвитку освітньої діяльності науково-педагогічних працівників	Сучасні методи дистанційного навчання	сертифікат
6				Працівник 1905	Центр підвищення кваліфікації, професійного розвитку освітньої діяльності науково-педагогічних працівників	Сучасні методи дистанційного навчання	довідка
7				Працівник 2499	Центр підвищення кваліфікації, професійного розвитку освітньої діяльності науково-педагогічних працівників	Сучасні методи дистанційного навчання	сертифікат
8				Працівник 2499	Центр підвищення кваліфікації, професійного розвитку освітньої діяльності науково-педагогічних працівників	Сучасні методи дистанційного навчання	сертифікат
9				Працівник 1865	Центр підвищення кваліфікації, професійного розвитку освітньої діяльності науково-педагогічних працівників	Сучасні методи дистанційного навчання	сертифікат
10				Працівник 1638	Центр підвищення кваліфікації, професійного розвитку освітньої діяльності науково-педагогічних працівників	Сучасні методи дистанційного навчання	сертифікат
11				Працівник 2680	Центр підвищення кваліфікації, професійного розвитку освітньої діяльності науково-педагогічних працівників	Сучасні методи дистанційного навчання	сертифікат
12				Працівник 3077	Центр підвищення кваліфікації, професійного розвитку освітньої діяльності науково-педагогічних працівників	Сучасні методи дистанційного навчання	сертифікат
13				Працівник 1702	Центр підвищення кваліфікації, професійного розвитку освітньої діяльності науково-педагогічних працівників	Сучасні методи дистанційного навчання	довідка
14				Працівник 2569	Центр підвищення кваліфікації, професійного розвитку освітньої діяльності науково-педагогічних працівників	Сучасні методи дистанційного навчання	сертифікат
15				Працівник 1652	Центр підвищення кваліфікації, професійного розвитку освітньої діяльності науково-педагогічних працівників	Сучасні методи дистанційного навчання	сертифікат
16				Працівник 1652	Центр підвищення кваліфікації, професійного розвитку освітньої діяльності науково-педагогічних працівників	Сучасні методи дистанційного навчання	сертифікат
17				Працівник 1652	Центр підвищення кваліфікації, професійного розвитку освітньої діяльності науково-педагогічних працівників	Сучасні методи дистанційного навчання	сертифікат
18				Працівник 1649	Центр підвищення кваліфікації, професійного розвитку освітньої діяльності науково-педагогічних працівників	Сучасні методи дистанційного навчання	сертифікат
19				Працівник 1953	Центр підвищення кваліфікації, професійного розвитку освітньої діяльності науково-педагогічних працівників	Сучасні методи дистанційного навчання	сертифікат
20				Працівник 3230	Центр підвищення кваліфікації, професійного розвитку освітньої діяльності науково-педагогічних працівників	Сучасні методи дистанційного навчання	сертифікат

Рисунок 3.9 – Інтерфейс обліку документів про підвищення кваліфікації

На цій сторінці акумулюються відомості про проходження науково-педагогічними працівниками відповідних курсів та стажувань. Користувачі вносять до системи назву організації, на базі якої здійснювалося навчання, тему курсу, а також вид отриманого документа, що підтверджує здобуття нових навичок.

Усі внесені публікації та завантажені документи підлягають обов'язковій перевірці. Уповноважений модератор здійснює верифікацію даних із можливістю їх затвердження або повернення на доопрацювання, при цьому в системі реалізовано функціонал обов'язкового зазначення причини відхилення (рис. 3.10):

Причина повернення

×

Не вказана тема публікації

Закрити

Рисунок 3.10 – Модальне вікно з причиною повернення документа

Інтерфейсне вікно у вигляді модальної форми виводить внесену перевіряючим причину для відхилення. Це формує прозорий зворотний зв'язок і дозволяє користувачеві оперативно виправити конкретну помилку для повторного надсилання документа.

Для комплексної оцінки відповідності всього кадрового складу нормативним критеріям МОН розроблено сторінку «Виконання п.38» (рис. 3.11):

Виконання п.38							
Показані 1-15 із 1980 записів.							
Експортувати усі (Досягнення)		Експортувати усі (Підвищення кваліфікації)		Науково-педагогічні пра...	Активні працівники	Усі	
#	П.І.Б	Виконання п.38		Структурний підрозділ	Посада	Тип працевлаштування	Чи зовнішній сумісник
		X Не виконано				Усі	
1	Працівник 1633	ПК 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 19 20		Факультет авіонавігації, електроніки та телекомунікацій	Заступник декана	Штатна посада	Ні
2	Працівник 1630	ПК 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 19 20		Факультет авіонавігації, електроніки та телекомунікацій	Заступник декана	Штатна посада	Ні
3	Працівник 1637	ПК 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 19 20		Факультет авіонавігації, електроніки та телекомунікацій	Заступник декана	Штатна посада	Ні
4	Працівник 2118	ПК 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 19 20		Факультет авіонавігації, електроніки та телекомунікацій	Декан факультету	Штатна посада	Ні
5	Працівник 2476	ПК 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 19 20		Факультет екологічної безпеки, інженерії та технологій	Заступник декана	Штатна посада	Ні
6	Працівник 2463	ПК 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 19 20		Факультет екологічної безпеки, інженерії та технологій	Декан факультету	Штатна посада	Ні
7	Працівник 2475	ПК 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 19 20		Факультет екологічної безпеки, інженерії та технологій	Заступник декана	Штатна посада	Ні
		X Не виконано					

Рисунок 3.11 – Загальний інтерфейс моніторингу виконання п. 38

Інтерфейс відображає зведений список співробітників, їхній структурний підрозділ, посаду, тип працевлаштування та ознаку зовнішнього сумісництва. У

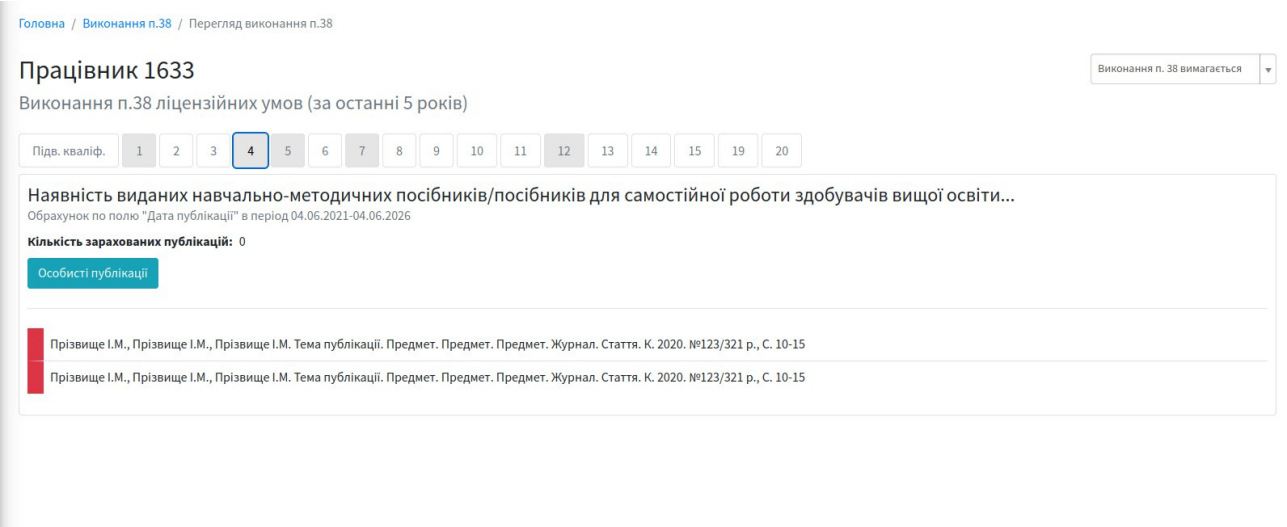
верхній частині екрана реалізовано блок фільтрації, призначений для сортування та швидкого пошуку працівників за визначеними критеріями.

Візуалізація стану виконання підпунктів ліцензійних вимог реалізована за допомогою автоматичного розрахунку трьох базових статусів, наочно продемонстрованих на рисунках 3.12, 3.13 та 3.14:



Усі пункти мають чітке кольорове кодування: зелений колір індикатора свідчить про успішне виконання вимоги та підтвердження даних модератором; сірий колір означає, що внесені дані є недостатніми для зарахування або наразі очікують на перевірку.

Щоб дізнатися більше інформації про конкретного співробітника, є можливість перейти на його персональну сторінку моніторингу ліцензійних умов (рис. 3.15):



Робочий простір тут розподілено за допомогою пронумерованих тематичних вкладок, кожна з яких відповідає конкретному пункту, включно з вкладкою підвищення кваліфікації. Натискання на відповідну табу відкриває

внутрішній інтерфейс, який містить текстове пояснення критерію та табличні дані з деталізацією всіх підтверджених досягнень викладача за цим напрямом.

Наступною складовою інтерфейсу є модуль «Статистика та звіти», який призначений для збору та перетворення первинних даних у наочні зведені звіти за кількома основними напрямами. Модуль дозволяє детально аналізувати показники вступної кампанії, документи науково-педагогічних працівників, а також загальний контингент студентів та його рух. Робота з аналітикою починається з налаштування часових меж, де користувач може самостійно обрати базовий рік, з якого система розпочне розрахунок усіх статистичних показників (рис. 3.16):

Рисунок 3.16 – Форма ведення року для розрахунку аналітики

Після встановлення початкового року перейдемо до перегляду першої статистики, яка відображає хід вступної кампанії. Весь робочий простір цієї сторінки розділений на п'ять функціональних вкладок: «Динаміка зарахування», «Факультети», «Спеціальності», «Гендерна структура» та «Вікова структура». Для прикладу розглянемо вкладку «Спеціальності». На рисунку 3.17 показано верхню частину екрана, де розміщено інформаційні блоки з кількістю зарахованих осіб та розгорнуту панель фільтрів:

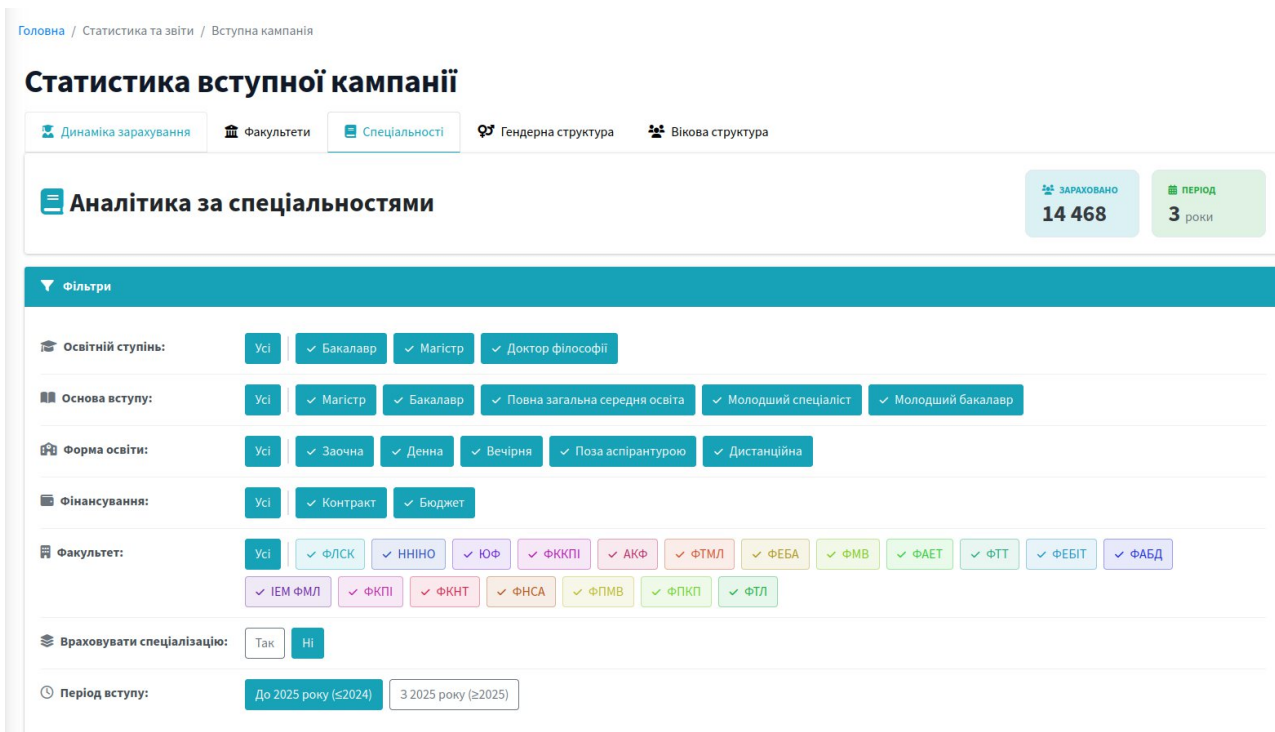


Рисунок 3.17 – Панель фільтрації

За допомогою цих фільтрів можна гнучко налаштовувати відображення даних за освітніми ступенями, формою освіти, фінансуванням чи конкретними факультетами. Наступний блок сторінки містить елементи вибору кодів спеціальностей та лінійний графік динаміки зарахування студентів (рис. 3.18):

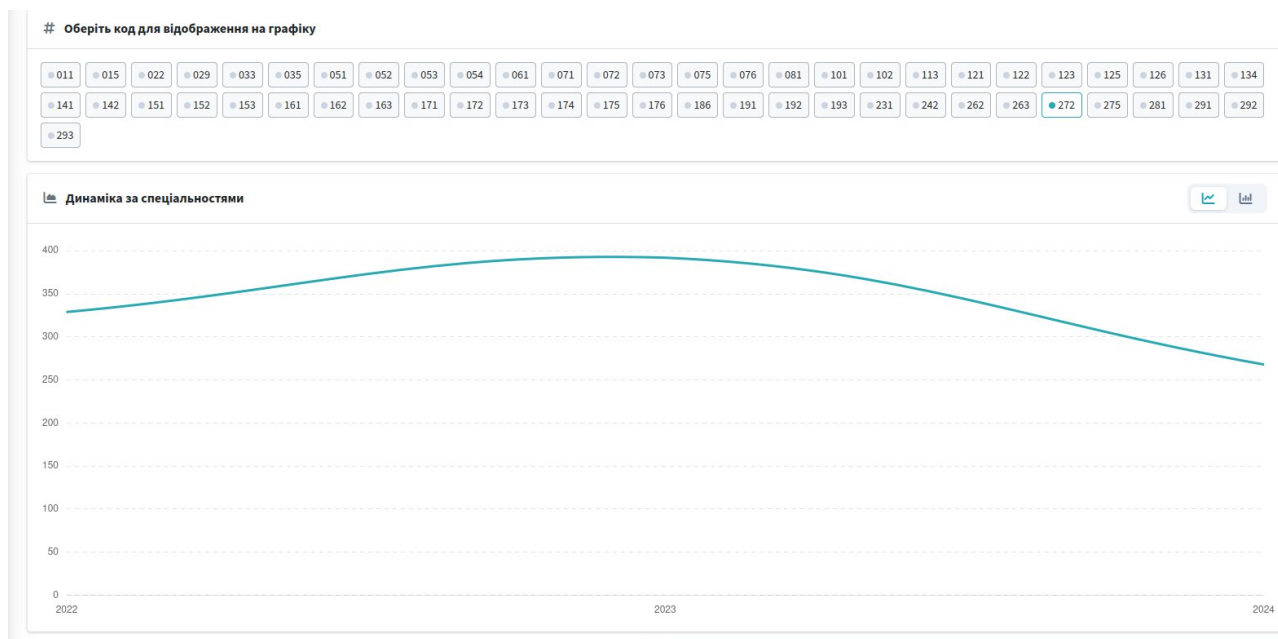


Рисунок 3.18 – Графік динаміки зарахування за обраною спеціальністю

Нижня частина сторінки містить аналітичні таблиці: рейтинги ТОП-10 спеціальностей з найбільшим та найменшим набором, а також зведену інформацію про кількість людей на кожній спеціальності за роками (рис. 3.19):

ТОП-10 (Найбільший набір)

код	назва	осіб
272	272 Авіаційний транспорт	989
073	073 Менеджмент	901
053	053 Психологія	719
275	275 Транспортні технології	711
125	125 Кібербезпека та захист інформації	703
122	122 Комп'ютерні науки	692
121	121 Інженерія програмного забезпечення	663
123	123 Комп'ютерна інженерія	621
081	081 Право	598
125	125 Кібербезпека	482

ТОП-10 (Найменший набір)

код	назва	осіб
102	102 Хімія	2
175	175 Інформаційно-вимірвальні технології	5
011	011 Освітні, педагогічні науки	7
015	015 Професійна освіта	11
152	152 Метрологія та інформаційно-вимірвальна техніка	16
033	033 Філософія	19
052	052 Політологія	29
176	176 Мікро- та наносистемна техніка	32
153	153 Мікро- та наносистемна техніка	40
071	071 Облік і оподаткування	42

Загальна статистика по спеціальностях

Експорт Excel

рік	всього	011	015	022	029	033	035	051	052	053	054	061	071	072	073	075	076	081	101	102	113	121	122	123	125	126	131	134	141	1
2022	5 277	7	4	189	36	8	117	96	13	240	23	79	28	65	285	122	65	256	44	1	31	220	290	293	482	38	2	74	83	4
2023	5 451	-	5	150	18	5	101	106	6	286	22	97	12	67	382	173	85	228	47	1	15	238	233	217	424	55	46	88	72	2
2024	3 740	-	2	87	18	6	109	94	10	193	12	47	2	53	234	112	56	114	37	-	14	205	169	111	279	28	20	75	56	2

Рисунок 3.19 – Рейтинги спеціальностей та загальна звітна таблиця

Також над фінальною таблицею розміщено кнопку «Експорт Excel», яка дозволяє миттєво вивантажити всі ці дані у файл.

Наступна статистика відображає стан та обсяги документів науково-педагогічних працівників. При переході до цього розділу відкривається сторінка, де показується загальна статистика по всіх наявних у базі типах документів НПП (рис. 3.20):

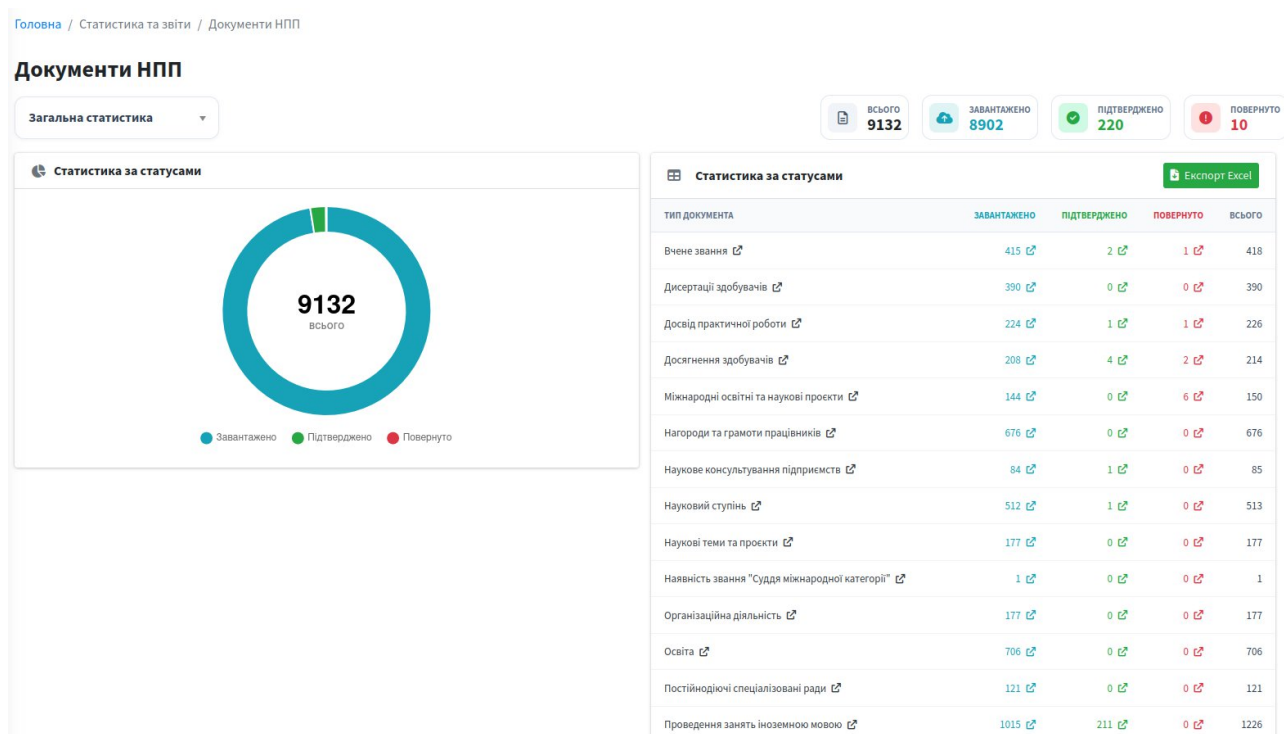


Рисунок 3.20 – Головна сторінка статистики документів НПП

Для більш детального аналізу користувач може вибрати конкретний вид кваліфікаційного документа. Вибір здійснюється за допомогою зручного випадаючого списку у верхній частині робочої панелі (рис. 3.21):

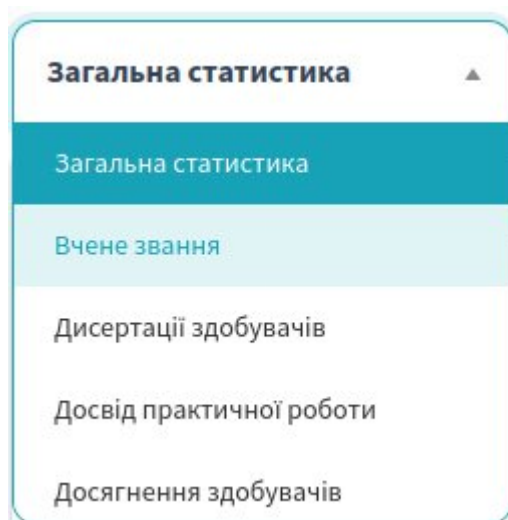


Рисунок 3.21 – Випадаючий список для вибору типу документа

Для прикладу розглянемо статистику за документом «Науковий ступінь». Після його вибору система виводить загальну кількість зафіксованих документів цього типу та інформаційні картки (рис. 3.22):

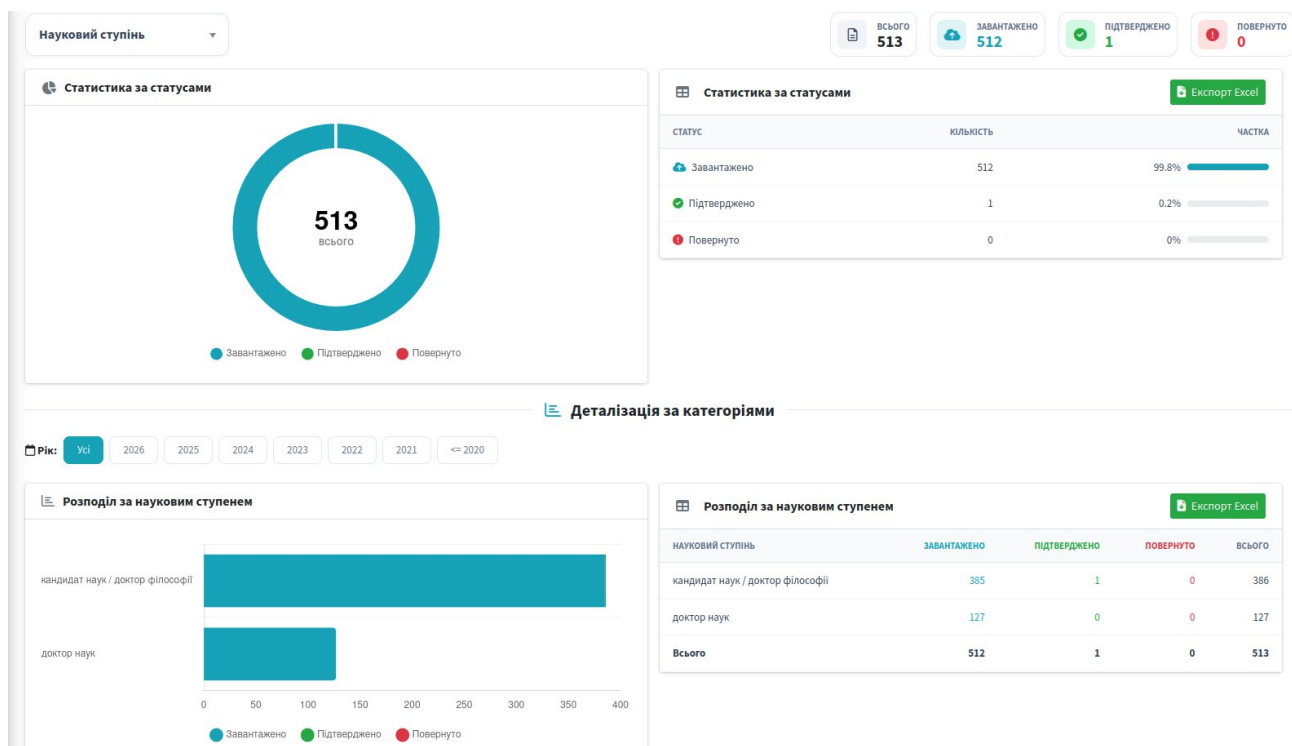


Рисунок 3.22 – Статистика за відповідним документом

Нижче на сторінці відображається деталізація даних за категоріями. У вигляді таблиці та графічних елементів система показує розподіл працівників, наприклад, за конкретними науковими ступенями чи галузями науки (рис. 3.23):

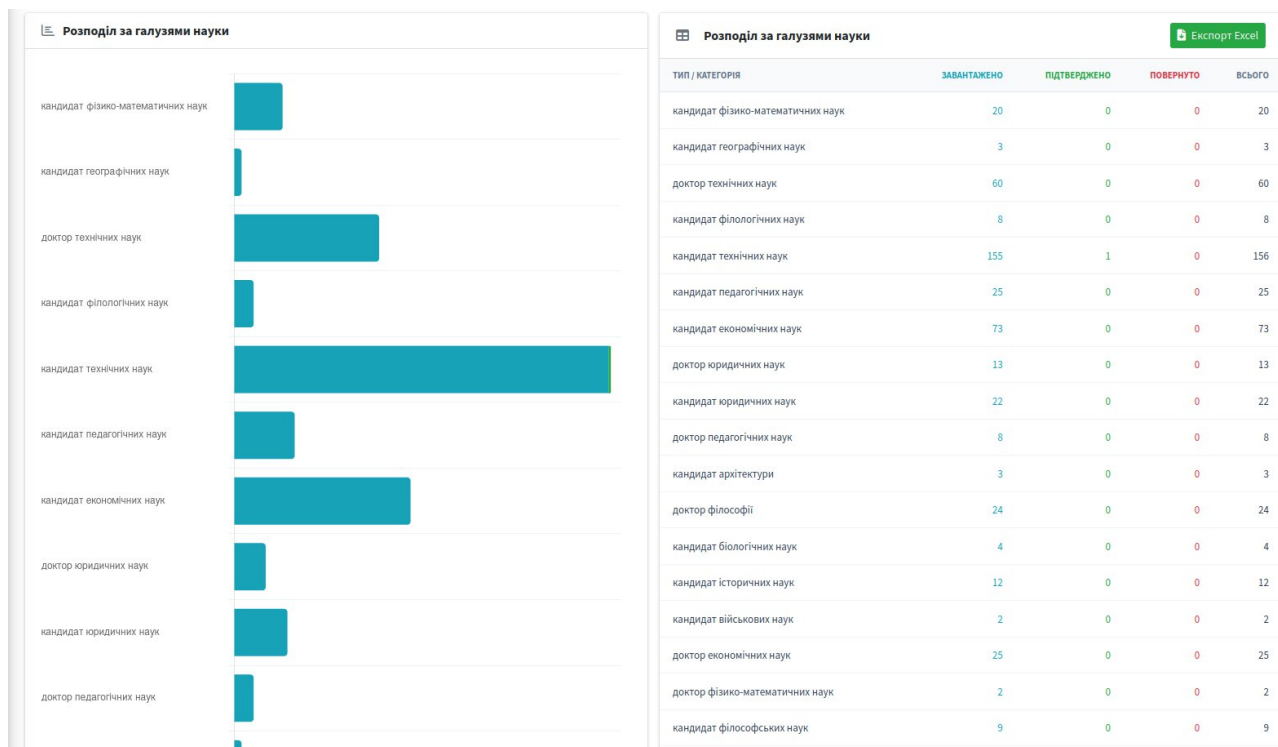


Рисунок 3.23 – Деталізація статистичних даних за категоріями документів

Наступний блок аналітики відображає статистичні дані щодо контингенту здобувачів освіти. Робота з цим розділом починається з панелі налаштувань, де розміщено елементи фільтрації для вибірки необхідної інформації (рис. 3.24):

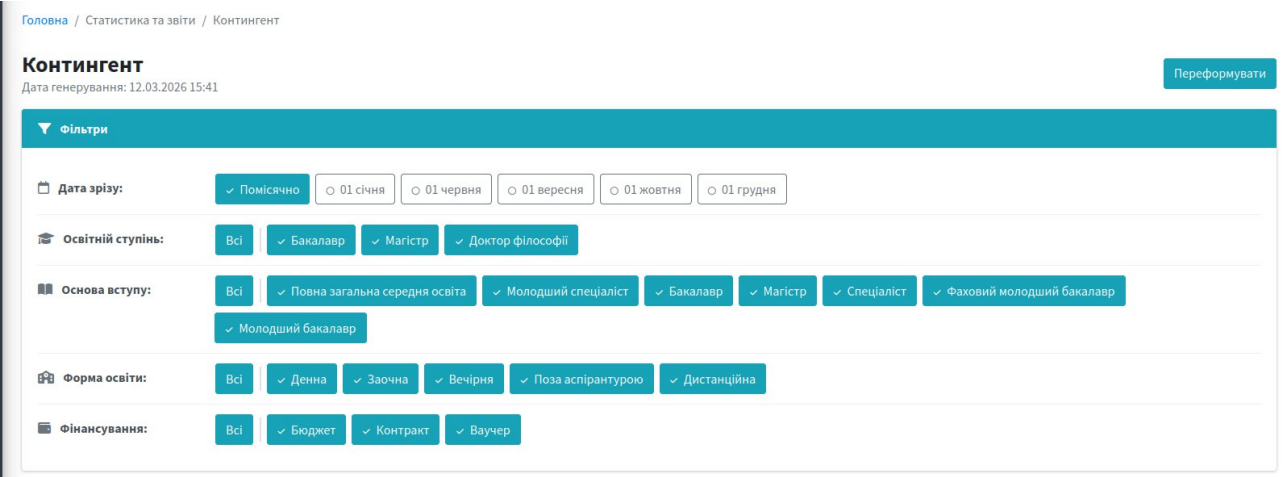


Рисунок 3.24 – Інтерфейс фільтрації параметрів контингенту студентів

За допомогою цих фільтрів користувач може задавати параметри відображення за формою навчання, фінансуванням, освітнім ступенем, основою вступу та конкретною датою зрізу даних.

Після застосування обраних параметрів система формує графічні звіти. На сторінці відображається лінійний графік загальної чисельності студентів та діаграми для порівняння показників між різними факультетами (рис. 3.25):

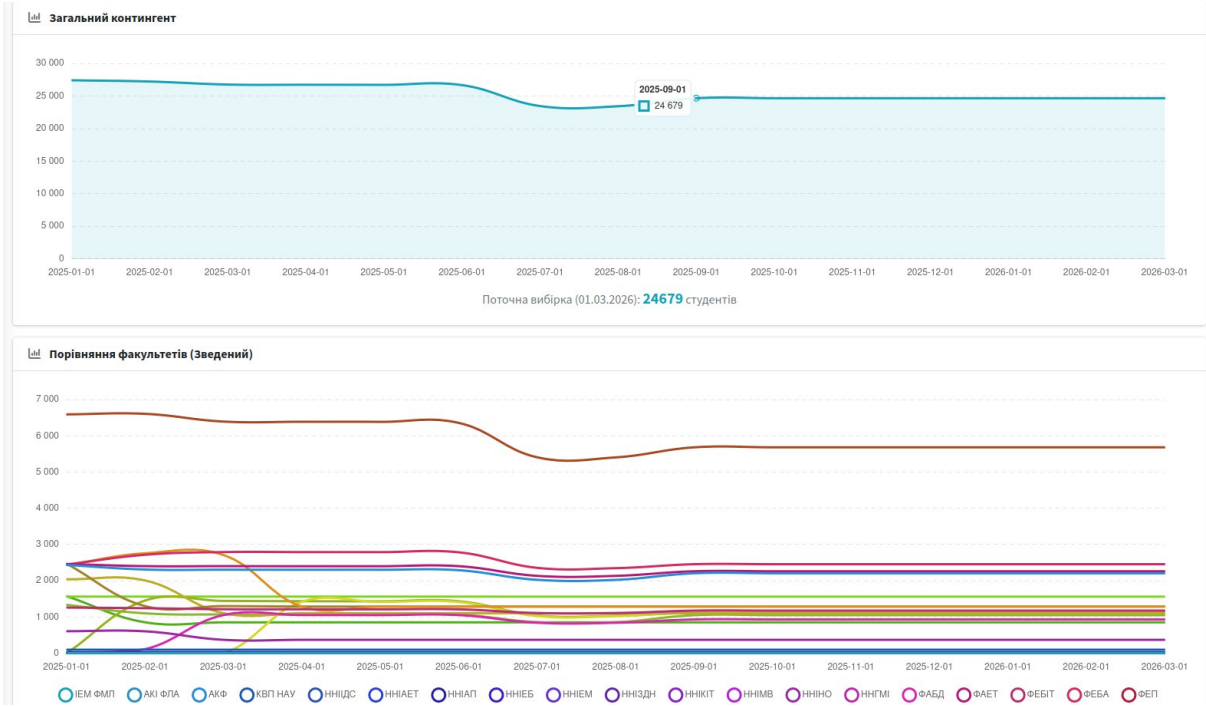


Рисунок 3.25 – Графіки загального контингенту та порівняння факультетів

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				70
Змн.	Арк.	№ докум.	Підпис	Дата		

У нижній частині сторінки розміщено блоки з детальними цифровими показниками. В них наведена розгорнута інформація про кількість студентів, окремо по кожному факультету (рис. 3.26):

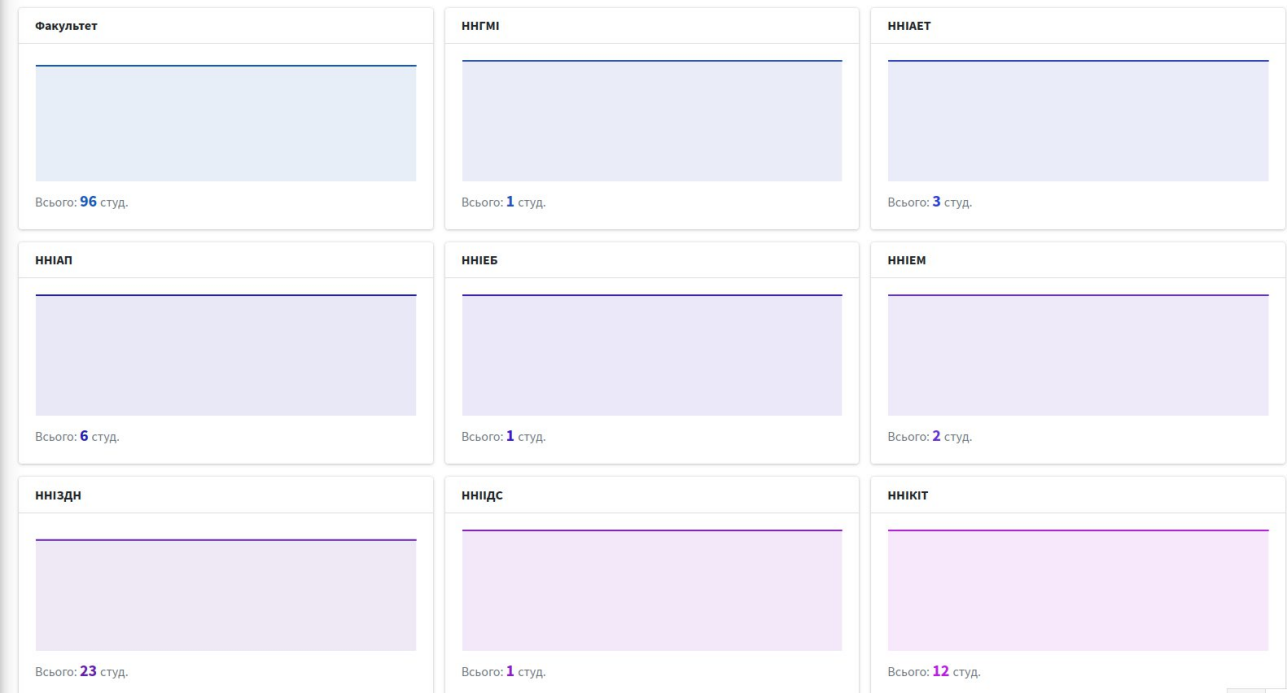


Рисунок 3.26 – Деталізація контингенту за факультетами

Останнім компонентом аналітичного модуля є статистика руху контингенту студентів, яка дозволяє відстежувати процеси зарахування, переведення та відрахування здобувачів освіти. Верхня частина цієї сторінки містить панель із фільтрами для налаштування відображення даних (рис. 3.27):



Рисунок 3.27 – Панель фільтрів статистики руху контингенту

Нижче на сторінці розміщено графік «Зведений баланс (Прибуло – Вибуло)» (рис. 3.28). Ця діаграма наочно демонструє динаміку змін у чисельності студентів, відображаючи співвідношення між кількістю прийнятих та вибулих осіб за роками.

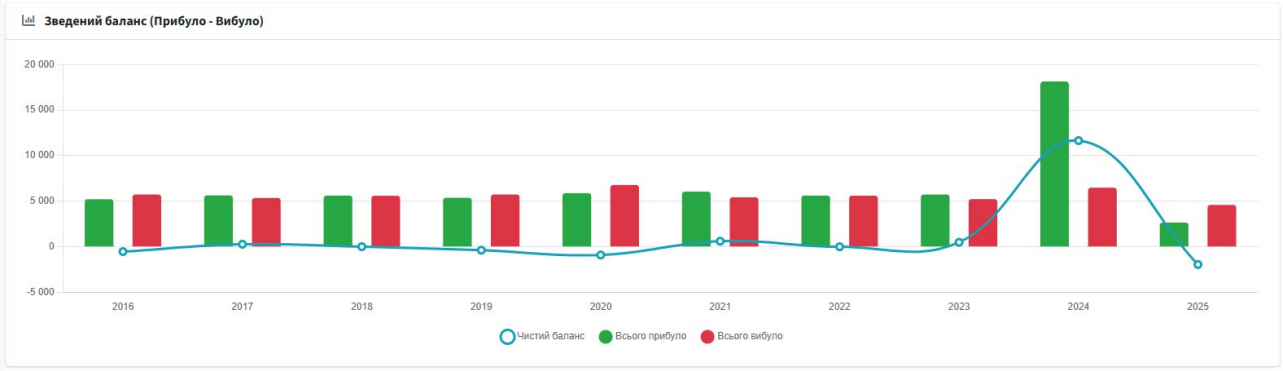


Рисунок 3.28 – Динаміка змін чисельності студентів

Наступним елементом сторінки є блоки, що деталізують статуси руху контингенту (рис. 3.29). Ці картки розподіляють дані за конкретними статусами, такими як переведення на наступний курс, поновлення на навчання, відрахування або надання академічної відпустки.

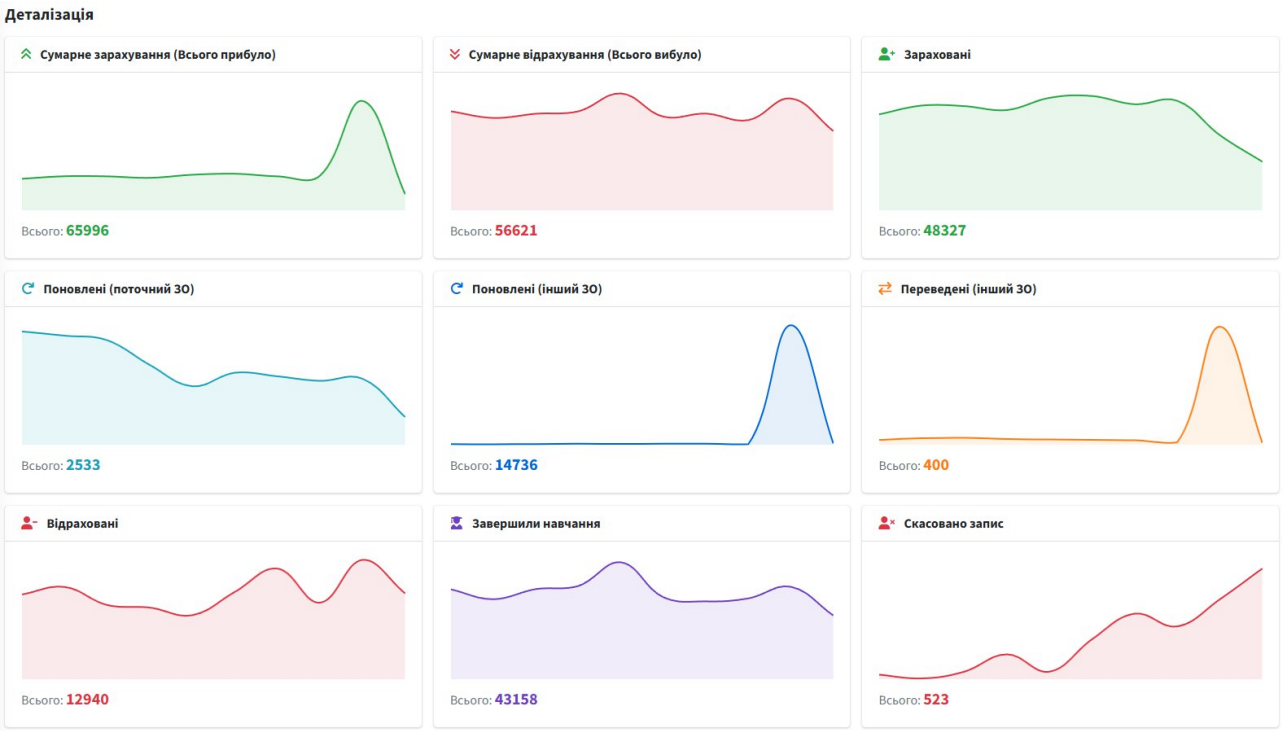


Рисунок 3.29 – Інформаційні блоки деталізації статусів руху контингенту

У самому низу сторінки знаходиться зведена таблиця, у якій показники за кожним статусом руху контингенту розподілені по роках (рис. 3.30). Вона

дозволяє порівнювати кількісні показники за різні періоди. Також є можливість експортування даних у формат Excel

Зведена таблиця									Експорт Excel
РІК	БАЛАНС	ЗАРАХОВАНІ	ПОНОВЛЕНІ (ПОТОЧНИЙ ЗО)	ПОНОВЛЕНІ (ІНШИЙ ЗО)	ПЕРЕВЕДЕНІ (ІНШИЙ ЗО)	ВІДРАХОВАНІ	ЗАВЕРШИЛИ НАВЧАННЯ	СКАСОВАНО ЗАПИС	
2016	-519	4774	376	63	12	1265	4473	6	
2017	284	5213	362	54	16	1378	3982	1	
2018	18	5185	347	75	17	1108	4488	10	
2019	-365	4986	265	116	14	1070	4643	33	
2020	-896	5598	194	84	13	956	5819	10	
2021	619	5687	239	118	12	1304	4080	53	
2022	13	5277	227	112	11	1652	3875	87	
2023	504	5451	212	71	6	1143	4023	70	
2024	11654	3740	219	13891	294	1783	4601	106	
2025	-1937	2416	92	152	5	1281	3174	147	

Рисунок 3.30 – Зведена таблиця статусів руху контингенту за роками

3.3. Тестування роботи розробленого додатку

У межах розробки інформаційної системи онлайн-менеджменту освітніх процесів було проведено ручне функціональне тестування. Цей підхід передбачав послідовну перевірку всіх користувацьких сценаріїв, елементів графічного інтерфейсу та логіки навігації без застосування автоматизованих скриптів. Емуляція дій кінцевого користувача дозволила комплексно оцінити стабільність системи, правильність відображення даних та зручність інтерфейсу в умовах, максимально наближених до реальної експлуатації.

Головною метою тестування було підтвердження безпомилкового виконання бізнес-логіки додатку, забезпечення цілісності інформації при взаємодії з інтерфейсом та перевірка відповідності реалізованих функцій вимогам адміністрації закладу вищої освіти.

Під час тестування увага приділялася не лише факту виконання окремих операцій, а й логіці переходів між сторінками, коректності повідомлень системи, стабільності роботи фільтрів і відповідності результатів очікуваній поведінці користувача. Такий підхід дозволив оцінити систему комплексно, оскільки для адміністративного програмного забезпечення важливою є не тільки наявність функцій, а й передбачуваність їхньої роботи в різних умовах.

		Кукса Р.О.			ПІЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		73

Перевірка проводилася з позиції основних груп користувачів: працівника відділу кадрів, адміністратора та користувача, який переглядає статистичні дані. Для кожної ролі аналізувалося, чи доступні лише дозволені дії, чи правильно відображаються дані після зміни статусів, чи не виникає помилок під час переходу між модулями. Окремо перевірялася коректність роботи з великими наборами даних, оскільки саме такі ситуації є типовими під час формування статистичних звітів або перегляду зведеної інформації за кілька років.

Першим перевіреним напрямом була модерація та перегляд документів НПП. У межах тестування було відтворено повний цикл обробки кадрових документів: завантаження копій, відкриття документа відповідальним працівником, підтвердження коректних даних або повернення документа на доопрацювання із зазначенням причини. Перевірка показала, що система коректно змінює статус документа, оновлює інформацію в інтерфейсі без зайвих затримок і відображає користувачеві зрозумілі повідомлення про причини відхилення.

Окремо було протестовано механізм перевірки виконання ліцензійних умов за пунктом 38. Під час перевірки аналізувалося, чи правильно система враховує підтверджені досягнення викладача, чи відсікає застарілі дані за п'ятирічний період і чи оновлює кольорову індикацію після внесення змін до профілю працівника. Результати тестування підтвердили, що загальний статус і маркери окремих підпунктів змінюються автоматично відповідно до актуального стану даних.

Також було перевірено роботу аналітичних панелей, призначених для моніторингу чисельності здобувачів освіти та руху контингенту. Тестування охоплювало побудову графіків зарахувань, відрахувань, поновлень і переведень, а також застосування фільтрів для уточнення вибірки. Система коректно перебудовує графічні елементи після зміни параметрів і забезпечує часткове оновлення контенту без повного перезавантаження сторінки.

Фінальним напрямом перевірки стало тестування експорту статистичних звітів у формат Excel. Було перевірено формування зведених таблиць щодо

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Морозов А.В.				74
Змн.	Арк.	№ докум.	Підпис	Дата		

контингенту та кадрового складу, відкриття згенерованих файлів у табличних процесорах і коректність розпізнавання числових показників. Результати тестування засвідчили, що експортовані файли мають правильну структуру, числові дані відображаються коректно та можуть використовуватися для подальшої обробки або друку.

Додатково було перевірено поведінку системи у випадках некоректного введення даних. Зокрема, тестувалися порожні обов'язкові поля, неправильні формати дат, спроби збереження неповних форм і введення значень, які не відповідають очікуваному типу даних. Система повинна не лише блокувати збереження некоректної інформації, а й надавати користувачеві зрозумілі пояснення щодо причини помилки. Перевірка показала, що механізми валідації допомагають уникнути внесення помилкових даних і зменшують навантаження на відповідальних працівників.

Окрема увага приділялася перевірці навігації між сторінками. Для адміністративних систем важливо, щоб користувач міг швидко переходити між списками, формами редагування, сторінками перегляду та аналітичними панелями. У процесі тестування було перевірено, чи зберігається логіка переходів після виконання дій, чи не виникають помилки при поверненні до попередніх сторінок і чи не втрачаються результати фільтрації без потреби. Це дозволило оцінити не лише технічну працездатність, а й зручність практичного використання системи.

Також було перевірено коректність розмежування доступу між ролями користувачів. Користувачі без відповідних прав не повинні мати можливості переглядати або редагувати кадрові записи, модерувати документи, змінювати системні налаштування чи експортувати службову інформацію. Під час тестування аналізувалося, чи приховуються недоступні елементи інтерфейсу та чи блокується прямий перехід за адресами сторінок, які не відповідають правам конкретного користувача. Така перевірка є важливою для захисту персональних і службових даних.

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		75

Проведене ручне тестування підтвердило працездатність основних функцій розробленої системи. Перевірені сценарії охопили ключові напрями роботи модулів «Кадрове забезпечення» та «Статистика та звіти», зокрема обробку документів, перевірку ліцензійних умов, аналіз руху контингенту, фільтрацію даних і експорт звітів. За результатами тестування можна зробити висновок, що реалізований програмний продукт відповідає визначеним функціональним вимогам і може бути використаний для подальшого розвитку інформаційної системи.

Висновки до третього розділу

У третьому розділі було успішно виконано етап практичного розгортання та комплексної перевірки працездатності створеної інформаційної системи. На основі розробленої діаграми розгортання було організовано взаємодію між основними компонентами та допоміжними сервісами архітектури, що дозволило забезпечити високу відмовостійкість додатка та оптимальний розподіл обчислювального навантаження на серверне середовище.

Під час реалізації інтерфейсної логіки кадрового модуля було створено зручні інструменти для ведення реєстру працівників, обліку їхньої трудової діяльності, наукових публікацій та підвищення кваліфікації. Особливу увагу приділено автоматизації моніторингу ліцензійних вимог Міністерства освіти і науки України. Розроблений алгоритм розрахунку статусів із відповідним кольоровим кодуванням та зручним розподілом по вкладках дозволив суттєво спростити контроль відповідності викладацького складу нормативним критеріям.

Важливим кроком стало впровадження аналітичного блоку статистики та звітів, який дозволяє оцінювати ключові показники установи за кількома напрямками: вступна кампанія, документи викладачів, загальний контингент студентів та його рух. Завдяки інтеграції динамічних графіків та діаграм користувачі отримали можливість наочно аналізувати зміни чисельності

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		76

здобувачів освіти. Також було успішно реалізовано функціонал експорту звітів, що забезпечило швидке вивантаження табличних даних у Excel файли для їх подальшої обробки.

Проведене тестування за ключовими сценаріями підтвердило стабільність вебдодатка, коректність оновлення інформації в інтерфейсі та точність виконання бізнес-логіки. Отримані результати свідчать про повну готовність системи до реальної експлуатації в межах закладу вищої освіти та успішне досягнення всіх поставлених завдань розробки.

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		77

ВИСНОВКИ

У межах виконання кваліфікаційної роботи було реалізовано комплексний процес аналізу, проектування та розробки програмних модулів, призначених для автоматизації кадрових і аналітичних процесів закладу вищої освіти. Основною метою роботи стало створення сучасного програмного рішення, здатного забезпечити централізоване управління даними, підвищити ефективність адміністративної діяльності та оптимізувати процеси збору й обробки статистичної інформації.

На етапі дослідження було проаналізовано особливості функціонування інформаційних систем у сфері освіти та визначено основні проблеми, що виникають під час ведення кадрового обліку, формування звітності та контролю за освітніми процесами. Проведений аналіз дозволив сформулювати вимоги до програмного забезпечення та визначити основні принципи побудови системи. Для реалізації програмного продукту було обрано мову програмування PHP, фреймворк Yii2 та систему управління базами даних MySQL, що забезпечують гнучкість розробки, надійність зберігання інформації та високу продуктивність під час роботи з великими обсягами даних.

У процесі проектування було сформовано структуру системи, визначено основні сценарії взаємодії користувачів та розроблено логічну модель бази даних. Значну увагу приділено організації кадрового обліку, зберігання інформації про працівників, документацію, публікації та професійні досягнення. Також було передбачено механізми формування статистичних вибірок, аналітичних звітів та моніторингу ключових показників діяльності закладу освіти.

Під час практичної реалізації програмного продукту було створено функціональні модулі для роботи з кадровими даними, автоматизації перевірки документів, ведення електронного обліку та формування статистичної інформації. Реалізовано механізми обробки та модерації документів, автоматизованого формування звітності й експорту даних.

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		78

Окрему увагу приділено забезпеченню цілісності інформації, оптимізації роботи системи та підтримці зручної взаємодії користувача з вебінтерфейсом.

Фінальний етап роботи включав тестування основного функціоналу системи та перевірку коректності виконання ключових бізнес-процесів. Було перевірено роботу механізмів обробки даних, формування статистики, управління кадровою інформацією та генерації звітів. Результати тестування підтвердили стабільність роботи програмного забезпечення, коректність реалізованої логіки та відповідність поставленим функціональним вимогам.

Таким чином, було розроблено програмне рішення для автоматизації кадрових та аналітичних процесів закладу вищої освіти. Реалізований функціонал забезпечує централізоване зберігання інформації, формування статистичної звітності, контроль кадрових даних та автоматизацію обробки документів.

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		79

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Digital Transformation of Higher Education / A. W. Bates – Tony Bates Associates Ltd, 2019. – 320 с.
2. Платформа Moodle [Електронний ресурс] – Режим доступу до ресурсу: <https://moodle.org/>
3. Інформаційна система «АСУ ВНЗ» [Електронний ресурс] – Режим доступу до ресурсу: <https://vuz.osvita.net>
4. Програмний комплекс «ПС-Університет» [Електронний ресурс] – Режим доступу до ресурсу: <https://www.politek-soft.kiev.ua>
5. Yii 2.0 Guide: Views and Server Rendering [Електронний ресурс] – Режим доступу до ресурсу: <https://www.yiiframework.com/doc/guide/2.0/en/structure-views>
6. MVC Architecture / MDN Web Docs [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/en-US/docs/Glossary/MVC>
7. DataGrip: The Cross-Platform IDE for Databases [Електронний ресурс] – Режим доступу до ресурсу: <https://www.jetbrains.com/datagrip/>
8. Active Record Pattern / Martin Fowler [Електронний ресурс] – Режим доступу до ресурсу: <https://www.martinfowler.com/eaCatalog/activeRecord.html>
9. Node.js Official Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://nodejs.org/en/docs/>
10. MongoDB Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mongodb.com/docs/>
11. JSON (JavaScript Object Notation) [Електронний ресурс] – Режим доступу до ресурсу: <https://www.json.org/json-en.html>
12. Chart.js – Open source HTML5 Charts [Електронний ресурс] – Режим доступу до ресурсу: <https://www.chartjs.org/docs/latest/>
13. SheetJS Community Edition [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.sheetjs.com/>

		Кукса Р.О.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Морозов А.В.				80
Змн.	Арк.	№ докум.	Підпис	Дата		

- 14.PJAX (PushState + Ajax) [Электронный ресурс] – Режим доступа до ресурсу: <https://github.com/defunkt/jquery-pjax>
- 15.Docker Official Documentation [Электронный ресурс] – Режим доступа до ресурсу: <https://docs.docker.com/>
- 16.MySQL 8.0 Reference Manual [Электронный ресурс] – Режим доступа до ресурсу: <https://dev.mysql.com/doc/refman/8.0/en/>
- 17.PHP Manual [Электронный ресурс] – Режим доступа до ресурсу: <https://www.php.net/manual/en/>
- 18.PhpStorm Official Documentation [Электронный ресурс] – Режим доступа до ресурсу: <https://www.jetbrains.com/help/phpstorm/>
- 19.Patterns of Enterprise Application Architecture / Martin Fowler – Addison-Wesley, 2002. – 560 с.
- 20.Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems / Martin Kleppmann – O'Reilly Media, 2017. – 616 с.
- 21.Fundamentals of Software Architecture: An Engineering Approach / Mark Richards, Neal Ford – O'Reilly Media, 2020. – 432 с.
- 22.Clean Architecture: A Craftsman's Guide to Software Structure and Design / Robert C. Martin – Prentice Hall, 2017. – 432 с.
- 23.Design Patterns: Elements of Reusable Object-Oriented Software / Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides – Addison-Wesley, 1994. – 395 с.
- 24.SQL Performance Explained / Markus Winand – 2012. – 204 с.
- 25.Git Documentation [Электронный ресурс] – Режим доступа до ресурсу: <https://git-scm.com/doc>
- 26.REST API Design Rulebook / Mark Masse – O'Reilly Media, 2011. – 116 с.
- 27.Unified Modeling Language (UML) Specification [Электронный ресурс] – Режим доступа до ресурсу: <https://www.omg.org/spec/UML/>
- 28.Requirements Engineering: Foundation for Software Quality / Klaus Pohl – Springer, 2010. – 833 с.

		Кукса Р.О.			ПІЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				81
Змн.	Арк.	№ докум.	Підпис	Дата		

29. Web compatibility / MDN Web Docs [Электронный ресурс] – Режим доступа до ресурсу: <https://developer.mozilla.org/en-US/docs/Web/Guide/Compatibility>
30. HTML Living Standard [Электронный ресурс] – Режим доступа до ресурсу: <https://html.spec.whatwg.org/>
31. CSS: Cascading Style Sheets / MDN Web Docs [Электронный ресурс] – Режим доступа до ресурсу: <https://developer.mozilla.org/en-US/docs/Web/CSS>
32. The Modern JavaScript Tutorial [Электронный ресурс] – Режим доступа до ресурсу: <https://javascript.info/>

		Кукса Р.О.			ПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				82
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

		Кукса Р.О.			ПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		83

```

    public static function saveFile($model, UploadedFile $file,
    string $uploadDir): array
    {
        $uploadDir = dirname(Yii::getAlias('@app')) . $uploadDir;
        $uniqueId = uniqid();
        $modelPath = $model->person_id ?? 'file';
        $fileName = "{$modelPath}_{$uniqueId}.{$file->extension}";
        $filePath = $uploadDir . $fileName;
        if ($file->saveAs($filePath)) {
            return [
                'success' => true,
                'filepath' => $filePath
            ];
        }
        return [
            'success' => false
        ];
    }
    /**
     * @param $model
     * @param string $attribute
     * @return bool
     */
    public static function deleteFile($model, string $attribute =
    'filepath'): bool
    {
        $filePath = $model->$attribute;
        if (!$filePath || !file_exists($filePath)) {
            return false;
        }
        if (unlink($filePath)) {

```

		Кукса Р.О.			ІІЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Морозов А.В.				84
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        $model->$attribute = '';

        return $model->save();

    }

    return false;

}

/**
 * @param $model
 * @param $attribute
 * @return array|false
 */
public static function downloadFile($model, $attribute =
'filepath'): array|false
{
    $filePath = $model->$attribute;

    if (!$filePath || !file_exists($filePath)) {
        return false;
    }

    return [
        'path' => $filePath,
        'name' => basename($filePath),
        'mimeType' => mime_content_type($filePath),
        'isImage' =>
str_starts_with(mime_content_type($filePath), 'image/')
    ];
}

private function processAchievementItem(array $item, int
$personId, array $batchData): array
{
    $methodId = $item['internal_code'];

    $codeLevel1 = $item['code_level1'] ?? null;

```

		Кукса Р.О.			ІІЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Морозов А.В.				85
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        $indicatorTypeId = $codeLevel1 &&
isset($this->staffIndicators[$codeLevel1 - 1]) ?
$this->staffIndicators[$codeLevel1 - 1]['indicator_type_id'] :
null;

        $data = [];

        switch ($methodId) {

            case 'p.1':

                $data =
$this->calcP1($batchData['publications']);

                break;

            case 'p.2':

                $data =
$this->calcP2($batchData['publications']);

                break;

            case 'p.3':

                $data =
$this->calcP3($batchData['publications']);

                break;

            case 'p.4':

                $data =
$this->calcP4($batchData['publications']);

                break;

            case 'p.5':

                $data = $this->calcP5($batchData['degrees']);

                break;

            case 'p.6':

                $data =
$this->calcP6($batchData['supervisedDissertations']);

                break;

            case 'p.7':

                $data =
$this->calcP7($batchData['supervisedDissertations'],
$batchData['councilMembers']);

                break;

            case 'p.8':

```

		Кукса Р.О.			ІІЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		86

```

        $data =
$this->calcP8($batchData['scientificProjects'],
$batchData['editorialBoards']);

        break;

        case 'p.9':

            $data =
$this->calcP9($batchData['expertCommissions']);

            break;

        case 'p.10':

            $data =
$this->calcP10($batchData['internationals'],
$batchData['internationalReferees']);

            break;

        case 'p.11':

            $data =
$this->calcP11($batchData['enterpriseConsulting']);

            break;

        case 'p.12':

            $data =
$this->calcP12($batchData['publications']);

            break;

        case 'p.13':

            $data =
$this->calcP13($batchData['multilingualTeaching']);

            break;

        case 'p.14':

            $data =
$this->calcP14($batchData['studentResults'],
$batchData['organizationalActivities']);

            break;

        case 'p.15':

            $data =
$this->calcP15($batchData['studentResults'],
$batchData['organizationalActivities']);

            break;

        case 'p.19':

```

		Кукса Р.О.			ІІЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Морозов А.В.				87
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        $data =
$this->calcP19($batchData['staffOrganizations']);

        break;

    case 'p.20':

        $data =
$this->calcP20($batchData['practicalWorks']);

        break;

    case 'p.18':
    case 'p.17':
    case 'p.16':
    default:

        break;

}

$item['data'] = $data;
$item['indicator_type_id'] = $indicatorTypeId;

if (isset($item['children'])) {
    foreach ($item['children'] as &$child) {
        if
(isset($item['data'][$child['internal_code']])) {
            $child['data'] =
$item['data'][$child['internal_code']];

            unset($item['data'][$child['internal_code']]);
        } else {
            $child['data'] = [];
        }

        if (isset($child['children'])) {
            $child =
$this->processAchievementItem($child, $personId, $batchData);
        }
    }
}

```

		Кукса Р.О.			ІІЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Морозов А.В.				88
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        return $item;
    }

    protected function getSelectedTabName(): string
    {
        return DocumentsEnum::AWARD->value;
    }

    protected function fetchSummaryData(): array
    {
        return
$this->repository->getGeneralTabStatisticsQuery('award')->all();
    }

    protected function fetchAvailableYears(): array
    {
        return $this->repository->getAvailableYearsQuery('award',
'date')->column();
    }

    protected function fetchBreakdownsData(?string $year): array
    {
        $queryLevel =
$this->repository->getCategoryTabStatisticsQuery(
            'award',
            'dictionary_award_level',
            'award_level_id',
            'name'
        );

        $queryType =
$this->repository->getCategoryTabStatisticsQuery(
            'award',

```

		Кукса Р.О.			ІІЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Морозов А.В.				89
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        'dictionary_award_type',
        'award_type_id',
        'name'
    );

    $this->applyYearFilter($queryLevel, 'date', $year);
    $this->applyYearFilter($queryType, 'date', $year);

    return [
        $queryLevel->all(),
        $queryType->all(),
    ];
}

protected function getBreakdownsConfig(): array
{
    return [
        [
            'title' => 'Розподіл за рівнем нагороди',
            'columnName' => 'Рівень нагороди'
        ],
        [
            'title' => 'Розподіл за типом нагороди',
            'columnName' => 'Тип / Категорія'
        ]
    ];
}

public function getStatisticData(array $filters): array
{
    $query = $this->repository->getBaseQuery();

```

		Кукса Р.О.			ІІЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Морозов А.В.				90
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        $monthFilter = $filters['snapshot_date'] ?? null;

        unset($filters['snapshot_date']);

        if (!empty($monthFilter)) {
            $query->andWhere(['MONTH(snapshot_date)' =>
(int)$monthFilter]);
        }

        $query->andFilterWhere($filters);

        $rawData = $query->all();

        return $this->groupChartData($rawData);
    }

    function loadStatisticDetails() {
        const url = new URL(window.location.href);
        const year = $('input[name="year"]:checked').val();

        url.pathname
            = '/statistics/documents/get-data';
        url.searchParams.set('year', year);

        const $statisticDetails = $('#statistic-details');

        createLoader($statisticDetails[0], { replace:
false }).show();

        $statisticDetails.load(url.toString(), function () {
            initExportButtons();
        });
    }
}

```

		Кукса Р.О.			ІІЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Морозов А.В.				91
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        $(document).on('change', "input[name='year']",
loadStatisticDetails);

        loadStatisticDetails();
    });

    public function getQualificationStatistic($filters): array
    {
        $successDefenseQuery =
$this->repository->getQualificationSuccessWorkStatistic();

        $uploadedWorkQuery =
$this->repository->getQualificationUploadedStatistic();

        $workStatusesQuery =
$this->repository->getStatisticByStatusQuery();

        $this->applyFilters($successDefenseQuery, $filters);
        $this->applyFilters($uploadedWorkQuery, $filters);
        $this->applyFilters($workStatusesQuery, $filters);

        return [
            'successDefense' => $successDefenseQuery->all(),
            'uploadedWork' => $uploadedWorkQuery->all(),
            'workStatuses' => $workStatusesQuery->all(),
        ];
    }

/**
 * @param string $table
 * @return Query
 * @throws Exception
 */
    public function getGeneralTabStatisticsQuery(string $table):
Query
    {

```

		Кукса Р.О.			ІІЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Морозов А.В.				92
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        return (new Query())
            ->select([
                'statusName' => 'document_status.name',
                'statusId' => 'document_status.id',
                'generalCount' => new Expression('COUNT(*)')
            ])
            ->from($table)
            ->andWhere(["$table.person_id" =>
$this->getPersonIds()])
            ->innerJoin('staff_document_status', $table .
'.document_status_id' . ' = ' . 'document_status.id')
            ->groupBy('staff_document_status_id');
    }

/**
 * @param string $table
 * @param string $dictionaryTable
 * @param string $foreignKey
 * @param string $field
 * @return Query
 * @throws Exception
 */
    public function getCategoryTabStatisticsQuery(string $table,
string $dictionaryTable, string $foreignKey, string $field):
Query
    {
        return (new Query())
            ->select([
                'statusName' => 'document_status.name',
                'statusId' => 'document_status.id',
                'criteriaName' => "$dictionaryTable.$field",
                'generalCount' => new Expression('COUNT(*)')
            ])
    }

```

```

        ->from($table)

        ->andWhere(["$table.person_id" =>
$this->getPersonIds()])

        ->innerJoin('staff_document_status', $table .
'.document_status_id' . ' = ' . 'document_status.id')

        ->innerJoin($dictionaryTable, $table . '.' .
$foreignKey . ' = ' . $dictionaryTable . '.id')

        ->groupBy([

            'document_status_id',

            $foreignKey

        ]);

    }

/**
 * @param $table
 * @param $dateField
 * @return Query
 */
public function getAvailableYearsQuery($table, $dateField):
Query
{
    return (new Query())

        ->select([

            "YEAR($dateField) as year"

        ])

        ->from($table)

        ->distinct()

        ->orderBy(["YEAR($dateField)" => SORT_DESC]);

}

public function getStatisticData($filters): array
{

    $year = ArrayHelper::remove($filters, 'year');

```

```

        $generalPublicationStatisticQuery =
$this->repository->getGeneralPublicationStatistic();

        $publicationTypeQuery =
$this->repository->getPublicationStatisticByType();

        $countQuery =
$this->repository->getAggregatedCategoryCounts();

        if ($year !== null && $year !== '') {

            $generalPublicationStatisticQuery->andFilterWhere(DateHelperService::dateFilter($year, 'date'));

            $publicationTypeQuery->andFilterWhere(DateHelperService::dateFilter($year, 'date'));

            $countQuery->andFilterWhere(DateHelperService::dateFilter($year, 'date'));

        }

        $generalPublicationStatisticQuery->andFilterWhere($filters);

        $publicationTypeQuery->andFilterWhere($filters);

        $countQuery->andFilterWhere($filters);

        $publicationTypeData = $publicationTypeQuery->all();

        $generalStatisticData =
$this->generalPublicationStatisticQuery->all();

        $formattedPublicationTypeData =
$this->formatStatisticData($publicationTypeData);

        $formattedGeneralPublicationData =
$this->formatGeneralStatisticData($generalStatisticData);

        $counts = $countQuery->one();

        return [

            'data' => $formattedPublicationTypeData,

```

		Кукса Р.О.			ІІЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Морозов А.В.				95
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        'general' => $formattedGeneralPublicationData,
        'isScopusCount' => $counts['isScopusCount'] ?? 0,
        'isWosCount' => $counts['isWosCount'] ?? 0,
        'isProfessionalCount' =>
$counts['isProfessionalCount'] ?? 0,
        'isPublicCount' => $counts['isPublicCount'] ?? 0,
        'totalCount' => $counts['totalCount'] ?? 0,
    ];
}

/**
 * @throws Exception
 */
public function getStatisticData($filters): array
{
    $VersionId = $filters['schedule_version_id'] ??
ScheduleRoomStatisticRepository::getCurrentScheduleVersion();

    $query =
$this->repository->getBaseQuery($scheduleVersionId);

    if (!empty($filters['faculty_id'])) {
        $query->andWhere(['faculty_id' =>
$filters['faculty_id']]);
    }

    if (!empty($filters['degree_id'])) {
        $query->andWhere(['degree_id' =>
$filters['degree_id']]);
    }

    if (!empty($filters['course'])) {
        $query->andWhere(['sag.course' =>
$filters['course']]);
    }
}

```

```

$results = $query->all();

$totalGroups = count($results);
$totalSlots = 0;
$occupiedSlots = 0;
$freeSlots = 0;

foreach ($results as &$row) {
    $row['per_week'] = json_decode($row['per_week'],
true);
    $row['details'] = json_decode($row['details'], true);

    $occupiedSlots += (int)$row['sumPairs'];
}

unset($row);

if ($totalGroups > 0) {
    $weekCount = (int)$results[0]['weekCount'];

    $slotsPerGroup = 49 * $weekCount;
    $totalSlots = $totalGroups * $slotsPerGroup;
    $freeSlots = $totalSlots - $occupiedSlots;
}

$uniqueWeeks =
GroupStatisticRepository::getUniqueWeeks($scheduleVersionId);

$uniqueHours =
Hour::find()->select(['number'])->asArray()->column();

$uniqueDays =
Day::find()->select(['short_name'])->asArray()->distinct()->column();

```

		Кукса Р.О.			ІІЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Морозов А.В.				97
Змн.	Арк.	№ докум.	Підпис	Дата		

```

return [
    'weeks' => $uniqueWeeks,
    'hours' => $uniqueHours,
    'days' => $uniqueDays,
    'totals' => [
        'totalGroups' => $totalGroups,
        'totalSlots' => $totalSlots,
        'occupiedSlots' => $occupiedSlots,
        'freeSlots' => $freeSlots,
    ],
    'items' => $results,
];
}

public function actionIndex($tab = null)
{
    return $this->render('index');
}

public function actionGetTabData(string $tab): array
{
    Yii::$app->response->format = Response::FORMAT_JSON;

    try {
        $service = StatisticsFactory::create($tab);

        $params = Yii::$app->request->queryParams;

        return [
            'success' => true,
            'tab' => $tab,

```

		Кукса Р.О.			ІІЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Морозов А.В.				98
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        'filters' => $service->getFilterOptions(),
        'data' => $service->getStatisticsData($params),
    ];

    } catch (Exception $e) {
        return [
            'success' => false,
            'message' => $e->getMessage()
        ];
    }
}

```

		Кукса Р.О.			ПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				99
Змн.	Арк.	№ докум.	Підпис	Дата		