

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 «Веб-технології»)

на тему:

**«Розробка інформаційної системи онлайн-
менеджменту освітніх процесів Digital University UA:
модулі "Кваліфікаційні роботи" та "Опитування
здобувачів освіти»**

Виконав студент групи ВТ-22-1
КОНОВАЛІ Микола Михайлович

Керівник роботи:
БАКАЛЮК Тетяна Анатоліївна

Рецензент:
ЄФРЕМОВ Юрій Миколайович

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: **КОНОВАЛ Микола Михайлович**

Керівник роботи: **ВАКАЛЮК Тетяна Анатоліївна**

Тема роботи: **«Розробка інформаційної системи онлайн-менеджменту освітніх процесів Digital University UA: модулі "Кваліфікаційні роботи" та "Опитування здобувачів освіти»,**

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

Вихідні дані для роботи: **розробка модулів «Кваліфікаційні роботи» та «Опитування здобувачів освіти» інформаційної системи Digital University UA, що забезпечать цифровізацію процесу підготовки та захисту кваліфікаційних робіт із юридично значущим електронним підписанням документів через сервіс «Дія», а також автоматизацію збору, обробки та аналізу зворотного зв'язку від здобувачів освіти.**

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 «Кваліфікаційні роботи» та «Опитування здобувачів освіти», а також пояснювальної записки. Пояснювальна записка до випускної роботи містить 104 сторінки, 30 ілюстрацій та 3 таблиці.

Головною метою є розробка функціональних модулів інформаційної системи, що забезпечують цифровізацію процесів управління кваліфікаційними роботами здобувачів освіти з юридично значущим електронним підписанням документів через державний сервіс «Дія», а також автоматизацію організації, проведення та аналізу опитувань здобувачів у закладах вищої освіти.

У ході роботи були визначені основні завдання для розробки модулів, проведений аналіз існуючих аналогів інформаційних систем, досліджено механізми інтеграції з API державного сервісу «Дія» та обрано технології, що дозволяють реалізувати комплекс вимог до програмного забезпечення.

КЛЮЧОВІ СЛОВА: BACKEND, ІНФОРМАЦІЙНА СИСТЕМА, КВАЛІФІКАЦІЙНІ РОБОТИ, ОПИТУВАННЯ, РНР, YII2, ДІЯ, ЕЛЕКТРОННИЙ ПІДПИС, МІКРОСЕРВІС.

					ІПЗ.КР.Б – 121 – 26 – ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	Розробка інформаційної системи онлайн-менеджменту освітніх процесів Digital University UA: модулі "Кваліфікаційні роботи" та "Опитування здобувачів освіти" Пояснювальна записка	Літ.	Арк.	Аркуші
Розроб.		Коновал М.М.						
Керівник		Вакалюк Т.А.					3	104
Н. контр.						Житомирська політехніка, група ВТ-22-1		
Зав. каф.		Вакалюк Т.А.						

ABSTRACT

The qualification work consists of a software package for the online management information system of educational processes Digital University UA, in particular the modules "Qualification Works" and "Student Surveys", as well as an explanatory note. The explanatory note to the graduation work contains 104 pages, 30 illustrations and 3 tables.

The main goal is to develop functional modules of the information system that ensure the digitalization of the qualification works management process for students with legally significant electronic document signing through the state service "Diia", as well as the automation of organizing, conducting and analyzing student surveys in higher education institutions.

During the work, the main tasks for developing the modules were identified, an analysis of existing analogues of information systems was conducted, the mechanisms of integration with the "Diia" state service API were investigated, and technologies were selected that allow implementing a comprehensive set of software requirements.

KEYWORDS: BACKEND, INFORMATION SYSTEM, QUALIFICATION WORKS, STUDENT SURVEYS, PHP, YII2, DIIA, ELECTRONIC SIGNATURE, MICROSERVICE.

		Коновал М.М.			ІІЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Вакалюк Т.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

ЗМІСТ

РЕФЕРАТ.....	3
ABSTRACT	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	6
ВСТУП.....	7
РОЗДІЛ 1. АНАЛІЗ ВИМОГ КОРИСТУВАЧА ТА КОНЦЕПТУАЛЬНЕ МОДЕЛЮВАННЯ	9
1.1. Постановка задачі.....	9
1.2. Аналіз аналогів програмного продукту	10
1.3. Вибір архітектури системи.....	14
1.4. Інструментальні засоби розробки та вимоги до апаратного забезпечення..	18
Висновки до першого розділу.....	23
РОЗДІЛ 2. ПРОЕКТУВАННЯ СИСТЕМИ ОНЛАЙН-МЕНЕДЖМЕНТУ ОСВІТНІХ ПРОЦЕСІВ.....	25
2.1. Визначення варіантів використання та структури системи.....	25
2.2. Обґрунтування вибору системи управління базами даних.....	29
2.3. Інтеграція зовнішніх сервісів.....	32
2.4. Реалізація функціоналу системи управління навчальним процесом.....	37
Висновки до другого розділу	48
РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З СИСТЕМОЮ ОНЛАЙН-МЕНЕДЖМЕНТУ ОСВІТНІХ ПРОЦЕСІВ.....	50
3.1. Порядок встановлення та налаштування параметрів додатку	50
3.2. Структура інтерфейсу додатку	53
3.3. Тестування роботи розробленого додатку	68
Висновки до третього розділу.....	72
ВИСНОВКИ	74
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	76
ДОДАТКИ.....	78

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API – Application Programming Interface

БД – база даних

JSON – JavaScript Object Notation

КР – кваліфікаційна робота

ПЗ – програмне забезпечення

REST – Representational State Transfer

СУБД – система управління базою даних

CRUD – Create-Read-Update-Delete

ЗВО – заклад вищої освіти

		Коновал М.М.			ПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

ВСТУП

Сучасний етап розвитку інформаційного суспільства висуває принципово нові вимоги до організації та управління освітніми процесами у закладах вищої освіти. Традиційні підходи до управління кваліфікаційними роботами та збору зворотного зв'язку від здобувачів освіти, що базуються на паперовій документації та розрізнених інструментах, виявляються неефективними в умовах зростаючих обсягів даних, підвищених вимог до прозорості процесів та необхідності забезпечення віддаленого доступу до інформації. Особливої гостроти набуває проблема юридично значущого підписання документів, пов'язаних із захистом кваліфікаційних робіт, а також відсутність систематизованих механізмів збору та аналізу думок здобувачів щодо якості освітнього процесу. У цьому контексті актуальним є впровадження автоматизованих модулів управління, інтегрованих із державними цифровими сервісами, зокрема із застосунком «Дія» для електронного підписання документів, а також створення зручних інструментів для проведення опитувань здобувачів освіти.

Метою дослідження є створення цифрових модулів інформаційної системи Digital University UA, що автоматизують процеси управління кваліфікаційними роботами здобувачів освіти із забезпеченням юридично значущого електронного підпису через інтеграцію з сервісом «Дія», а також процеси організації, проведення та аналізу результатів опитувань здобувачів освіти.

Досягнення поставленої мети передбачає виконання таких завдань: аналіз сучасних підходів до цифровізації управління кваліфікаційними роботами та механізмів збору зворотного зв'язку від здобувачів освіти; дослідження можливостей інтеграції інформаційних систем із державним сервісом електронного підпису «Дія»; побудова моделі взаємодії користувачів у межах розроблюваних модулів; проєктування архітектури бази даних та програмних компонентів; розробка функціональних модулів «Кваліфікаційні роботи» та

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		7

«Опитування здобувачів освіти» як для університетського порталу, так і для особистого кабінету здобувача.

Об'єктом дослідження є процес розробки програмного забезпечення для автоматизації управління кваліфікаційними роботами з електронним підписанням документів через «Дію» та організації опитувань здобувачів освіти в межах інформаційної системи онлайн-менеджменту освітніх процесів закладу вищої освіти.

Предметом дослідження є технології, алгоритми та програмні засоби, що застосовуються для створення модулів цифрової системи управління освітніми процесами, зокрема механізми інтеграції з API сервісу «Дія» для реалізації електронного підпису документів, інструменти побудови та обробки опитувань, а також їхня адаптація до потреб сучасних закладів вищої освіти.

Практичне значення одержаних результатів полягає у тому, що розроблені модулі впроваджено у реальну інформаційну систему Digital University UA та використовуються в освітньому процесі університету. Модуль «Кваліфікаційні роботи» забезпечує повний цифровий документообіг на всіх етапах підготовки та захисту кваліфікаційних робіт, усуваючи потребу в паперовому підписанні документів завдяки інтеграції з державним сервісом «Дія». Модуль «Опитування здобувачів освіти» надає адміністрації університету інструмент для систематичного отримання та аналізу зворотного зв'язку від здобувачів, що сприяє підвищенню якості освітнього процесу та обґрунтованості управлінських рішень.

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Вакалюк Т.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		8

РОЗДІЛ 1. АНАЛІЗ ВИМОГ КОРИСТУВАЧА ТА КОНЦЕПТУАЛЬНЕ МОДЕЛЮВАННЯ

1.1. Постановка задачі

Цифровізація вищої освіти є одним із пріоритетних напрямів розвитку сучасних університетів, що зумовлено необхідністю підвищення якості управління освітніми процесами та зниження адміністративного навантаження на персонал. Незважаючи на це, процеси управління кваліфікаційними роботами у більшості закладів вищої освіти досі залишаються слабко автоматизованими: документи формуються вручну, підписання відбувається у паперовий спосіб, а контроль термінів і статусів покладається виключно на людський фактор. Не менш гострою є проблема збору зворотного зв'язку від здобувачів освіти - опитування проводяться нерегулярно, їхні результати не систематизуються належним чином і рідко стають основою для прийняття управлінських рішень.

Розпорошеність даних між різними підрозділами університету ускладнює координацію учасників процесу підготовки та захисту кваліфікаційних робіт. Науковий керівник, здобувач, кафедра та деканат часто діють без спільного інформаційного простору, що породжує затримки, втрату документів і відсутність прозорості на кожному етапі. Водночас відсутність механізму юридично значущого електронного підпису змушує університети зберігати громіздкий паперовий документообіг навіть там, де цифрові процеси вже впроваджені частково.

Вирішення зазначених проблем потребує комплексного підходу - створення єдиного цифрового середовища, в якому всі учасники освітнього процесу мають доступ до актуальної інформації, а ключові адміністративні операції виконуються автоматизовано та прозоро.

У межах інформаційної системи «Цифровий університет» розроблено два функціональні модулі - «Кваліфікаційні роботи» та «Опитування здобувачів освіти». Визначальною технологічною особливістю першого є інтеграція з державним сервісом «Дія», що забезпечує можливість юридично значущого

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

електронного підписання документів безпосередньо в системі, без необхідності фізичної присутності сторін.

Модуль «Кваліфікаційні роботи» охоплює формування та ведення реєстру кваліфікаційних робіт із зазначенням теми, наукового керівника, спеціальності та поточного статусу, а також перегляд повної картки роботи з усіма супровідними матеріалами та зручний інтерфейс комунікації між здобувачем, науковим керівником і рецензентом. Окремим функціональним блоком виступає підписання документів через сервіс «Дія» - ініціювання процесу підписання через інтеграцію з API, відстеження статусу підписання кожним із учасників у режимі реального часу та збереження підписаних документів і журналу підписань у системі.

Модуль «Опитування здобувачів освіти» забезпечує конструювання опитувань із гнучким налаштуванням питань та цільової аудиторії, проходження опитувань здобувачами через особистий кабінет, автоматизоване формування аналітичних звітів за результатами у розрізі спеціальностей, курсів та підрозділів, а також гарантування анонімності відповідей відповідно до налаштувань конкретного опитування.

1.2. Аналіз аналогів програмного продукту

Аналіз ринку програмних рішень для управління освітніми процесами свідчить про наявність широкого спектру платформ, кожна з яких по-різному підходить до вирішення завдань університетської адміністрації. Для порівняння було відібрано чотири системи, що різняться за архітектурою, функціональним наповненням та підходом до автоматизації.

Moodle - платформа з відкритим кодом для організації дистанційного навчання, що здобула широке застосування у закладах освіти різного масштабу завдяки безкоштовному поширенню та розвиненій спільноті розробників. Система надає інструменти для створення навчальних курсів, проведення тестувань, оцінювання знань і підтримки комунікації між учасниками освітнього процесу. Платформа відповідає міжнародним стандартам e-learning - SCORM і

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

хAPI - та підтримує інтеграцію із Zoom, Google Workspace і Microsoft Teams (рис. 1.1).

Офіційний сайт: <https://moodle.org/>

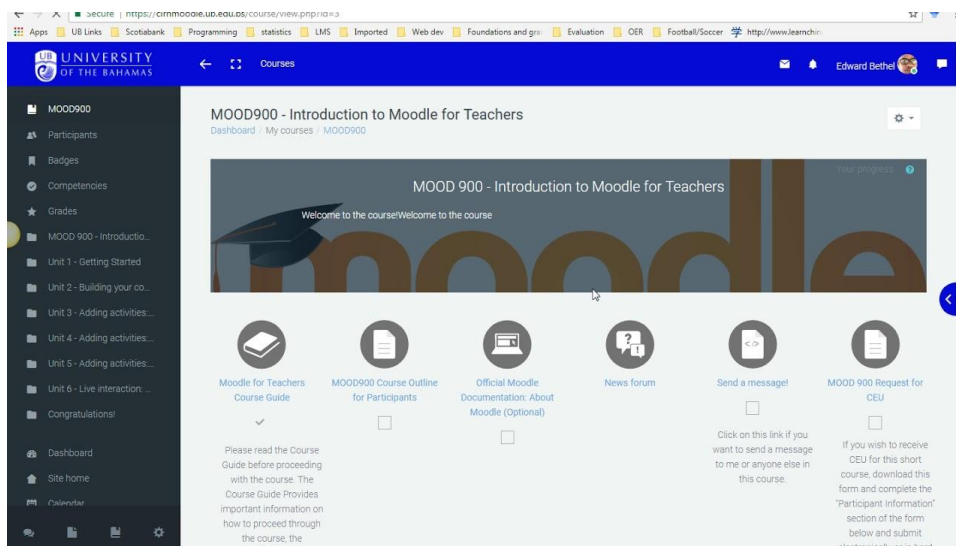


Рисунок 1.1 – Платформа Moodle

Відкритий вихідний код дозволяє розробникам створювати власні модулі та розширення, а висока гнучкість налаштування робить платформу придатною для різноманітних освітніх середовищ. Moodle є мультиплатформенною та доступна як на комп'ютерах, так і на мобільних пристроях. Разом із тим адміністрування системи потребує значного рівня технічної компетентності, інтерфейс може бути неінтуїтивним для нових користувачів, а без додаткової оптимізації серверів стабільна робота при великому навантаженні не гарантована. Управління кваліфікаційними роботами та інструменти для систематизованого опитування здобувачів у стандартному функціоналі відсутні.

OpenEduCat - модульна система управління освітнім процесом із відкритим кодом, побудована на базі ERP-платформи Odoo. Рішення орієнтоване на університети та інші навчальні заклади й охоплює широкий спектр адміністративних функцій: ведення електронного журналу, формування розкладів, управління бібліотекою, фінансами та університетською інфраструктурою. Завдяки інтеграції з екосистемою Odoo система легко розширюється модулями бухгалтерського обліку та CRM (рис. 1.2).

Офіційний сайт: <https://www.openeducat.org/>

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

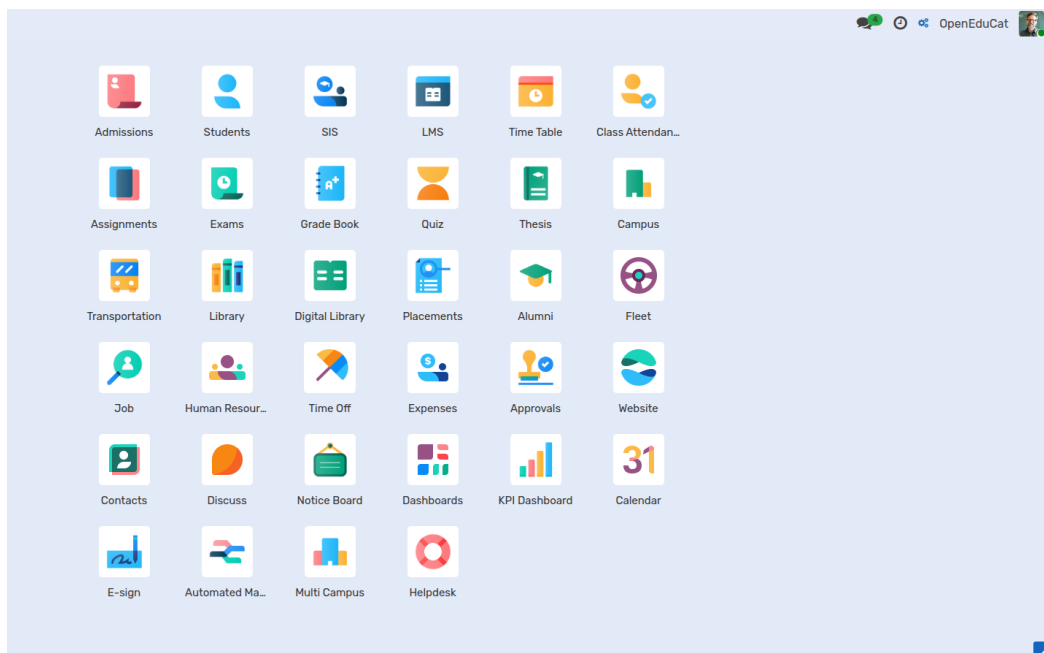


Рисунок 1.2 – Система OpenEduCat

До переваг системи належать відкритість вихідного коду, можливість розгортання як у хмарному середовищі, так і на локальних серверах, а також потужний аналітичний інструментарій для відстеження динаміки освітнього процесу. Однак початкове налаштування системи вимагає спеціалізованих знань у сфері Odoo, частина функціональних модулів доступна лише у платній версії, а документація та підтримка розробників є обмеженими. Інтеграція з державними сервісами електронного підпису, зокрема «Дією», у системі не передбачена.

Електронний кампус КПІ - українська інформаційна система для комплексної автоматизації діяльності закладу вищої освіти, розроблена з урахуванням специфіки вітчизняної освітньої системи. Платформа забезпечує управління навчальними планами, облік успішності здобувачів, організацію документообігу та взаємодію між структурними підрозділами університету. Система реалізована як вебзастосунок і доступна з будь-якого пристрою через браузер (рис. 1.3).

Офіційний сайт: <https://ecampus.kpi.ua/>

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

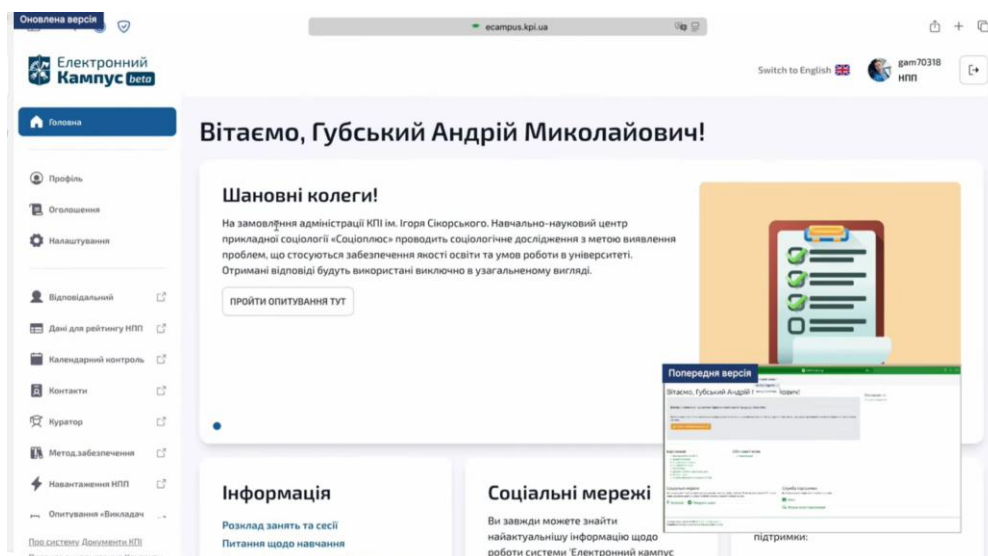


Рисунок 1.3 – Система Електронний кампус КПІ

Платформа вирізняється орієнтованістю на потреби українських університетів, інтеграцією з внутрішніми університетськими системами та зрозумілим інтерфейсом для адміністративного персоналу. Водночас система розроблена під конкретний заклад і не є універсальним тиражованим рішенням, функціонал управління кваліфікаційними роботами реалізований лише частково, а інтеграція з сервісом «Дія» для електронного підписання документів та інструменти для проведення опитувань здобувачів у ній відсутні.

Canvas LMS - хмарна платформа для управління освітнім процесом, розроблена компанією Instructure та орієнтована на університети й коледжі. Система забезпечує організацію онлайн-навчання, створення курсів і тестів, оцінювання здобувачів, ведення форумів та автоматичне збереження змін у реальному часі. Canvas LMS має відкритий API та підтримує інтеграцію з Google Workspace, Microsoft 365 і Zoom, а також мобільний застосунок для здобувачів і викладачів (рис. 1.4).

Офіційний сайт: <https://www.instructure.com/canvas>

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

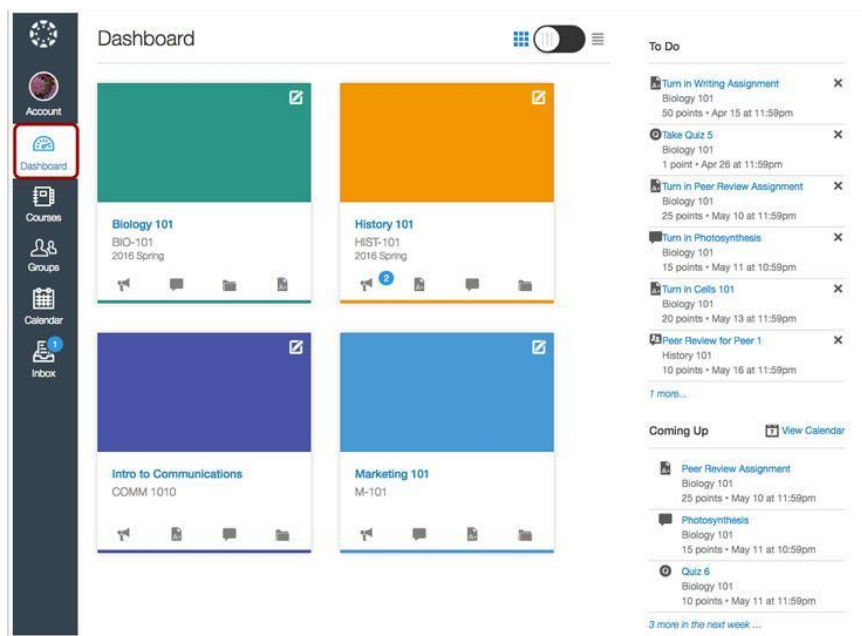


Рисунок 1.4 – Система Canvas LMS

Платформа вирізняється інтуїтивним інтерфейсом, гнучким налаштуванням ролей користувачів і рівнів доступу та стабільною технічною підтримкою з боку розробника. Проте висока вартість ліцензії може стати суттєвим обмеженням для закладів із невеликим бюджетом, окремі розширені функції доступні лише в платних версіях, а підтримка локального розгортання на серверах університету є обмеженою. Функціонал для управління кваліфікаційними роботами та проведення опитувань здобувачів у системі практично відсутній.

1.3. Вибір архітектури системи

Проектування сучасних інформаційних систем нерозривно пов'язане з вибором архітектурного підходу, який визначає не лише структуру коду, а й довгострокову здатність системи до масштабування, модернізації та підтримки. Архітектурне рішення, прийняте на початковому етапі розробки, безпосередньо впливає на складність супроводу системи, швидкість впровадження нових функціональних можливостей та загальну надійність програмного продукту в умовах реального навантаження. Особливо гостро це питання постає при розробці багатомодульних платформ, де різні функціональні блоки мають власну логіку, інтенсивність навантаження та вимоги до незалежного розвитку.

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Вакалюк Т.А.				14
Змн.	Арк.	№ докум.	Підпис	Дата		

Неправильно обраний архітектурний підхід може призвести до накопичення технічного боргу, ускладнення інтеграції із зовнішніми сервісами та неможливості ефективного масштабування окремих компонентів системи без впливу на її загальну стабільність.

У сучасній практиці розробки програмного забезпечення виокремлюють два основних архітектурних підходи, кожен із яких має власні переваги та обмеження залежно від специфіки проєкту, його масштабу та довгострокових вимог до розвитку.

Традиційним підходом до побудови програмних систем є монолітна архітектура, за якої всі компоненти - серверна логіка, база даних та інтерфейс користувача - об'єднані в єдиний неподільний застосунок. Сам термін «моноліт» у програмній інженерії означає цілісність і нерозривність структури системи, де всі її частини функціонують як єдине ціле та розгортаються спільно. Такий підхід є природним вибором на початкових етапах розробки, оскільки суттєво спрощує розгортання, налагодження та тестування системи. Усі компоненти функціонують в одному середовищі, що виключає мережеві затримки між модулями та забезпечує єдиний контроль над версіями й оновленнями. Для невеликих команд та проєктів із відносно стабільними вимогами монолітна архітектура дозволяє швидко досягти робочого результату без значних витрат на інфраструктуру.

Серед переваг монолітної архітектури варто виділити простоту розробки на початковому етапі, коли вся кодова база знаходиться в одному репозиторії та не потребує складної оркестрації між сервісами. Розгортання моноліту є значно простішим порівняно з розподіленими системами - достатньо оновити єдиний застосунок без необхідності координувати зміни між кількома незалежними сервісами. Тестування монолітного застосунку також є більш прямолінійним, оскільки всі компоненти доступні в одному середовищі виконання, що спрощує налагодження та виявлення помилок.

Проте зі зростанням функціональності монолітна система поступово перетворюється на складно керований об'єкт. Внесення змін до окремого модуля

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				15
Змн.	Арк.	№ докум.	Підпис	Дата		

потребує повного перетестування всього застосунку, оскільки будь-яка модифікація може непередбачувано вплинути на суміжні компоненти системи. Масштабування окремих частин моноліту стає технічно неможливим без масштабування всієї системи - навіть якщо навантаження зростає лише в одному функціональному блоці, ресурси виділяються для всього застосунку цілком. Ризик конфліктів між модулями зростає пропорційно до обсягу кодової бази, що суттєво уповільнює темп розробки нових функцій та ускладнює підтримку. Крім того, монолітна архітектура обмежує технологічний вибір - усі компоненти системи змушені використовувати єдиний стек технологій, навіть якщо окремі задачі могли б бути вирішені ефективніше засобами іншої платформи чи мови програмування.

Альтернативою моноліту є мікросервісна архітектура, яка передбачає декомпозицію системи на сукупність незалежних сервісів, кожен з яких реалізує окрему бізнес-функцію та взаємодіє з іншими через чітко визначені API. Кожен мікросервіс є самодостатнім компонентом із власною логікою, сховищем даних та циклом розгортання. Такий підхід дає змогу командам розробників працювати над різними сервісами паралельно, не створюючи конфліктів у спільній кодовій базі. Кожен мікросервіс може бути розгорнутий, оновлений або замінений незалежно від решти системи, що суттєво спрощує її еволюцію та знижує ризики при впровадженні змін.

Ключовою перевагою мікросервісного підходу є незалежне масштабування окремих компонентів відповідно до реального навантаження. Якщо певний сервіс отримує значно більше запитів порівняно з іншими, його можна масштабувати горизонтально без впливу на решту системи. Це забезпечує ефективніше використання обчислювальних ресурсів та дозволяє гнучко реагувати на зміни у профілі навантаження.

Для платформи з модулями різної інтенсивності використання - як-от «Кваліфікаційні роботи» з інтеграцією зовнішнього сервісу «Дія» та «Опитування здобувачів освіти» з піковим навантаженням у певні академічні періоди - незалежне масштабування є принципово важливою властивістю. Під

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				16
Змн.	Арк.	№ докум.	Підпис	Дата		

час захисту кваліфікаційних робіт або масових опитувань система повинна витримувати значно вищі навантаження, ніж у звичайний робочий день. Мікросервісна архітектура дозволяє виділяти додаткові ресурси саме для навантажених компонентів, не впливаючи на стабільність решти платформи.

Додатковою перевагою мікросервісного підходу є технологічна гетерогенність - можливість використовувати різні технологічні стеки для різних сервісів залежно від специфіки задачі. Це дозволяє обирати найефективніші інструменти для кожного конкретного завдання, не будучи обмеженим єдиним технологічним рішенням для всієї системи. Наприклад, основний застосунок може бути реалізований на PHP з фреймворком Yii2, тоді як сервіс для роботи з асинхронними операціями та зовнішніми API ефективніше реалізувати на Node.js з його подієво-орієнтованою моделлю виконання.

Мікросервісна архітектура також забезпечує вищу відмовостійкість системи загалом. Збій в одному сервісі не призводить до недоступності всієї платформи - решта компонентів продовжує функціонувати. Це є критично важливим для університетської інформаційної системи, де недоступність окремих функцій у відповідальні моменти - під час захисту кваліфікаційних робіт або проведення масових опитувань - може мати суттєві негативні наслідки для освітнього процесу.

Разом із тим мікросервісний підхід висуває підвищені вимоги до планування міжсервісної взаємодії, налаштування мережевої інфраструктури та моніторингу. Розподіленість компонентів неминуче вносить мережеві затримки у комунікацію між сервісами, а управління розподіленими транзакціями та забезпечення безпеки даних при передачі між сервісами потребують окремої уваги на етапі проєктування. Зростає також складність операційного управління - розгортання, моніторинг та налагодження розподіленої системи вимагають більш зрілих DevOps-практик порівняно з монолітним застосунком. Крім того, налагодження помилок у розподіленій системі є складнішим завданням, оскільки трасування запиту, що проходить через кілька сервісів, потребує спеціалізованих інструментів спостережуваності.

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				17
Змн.	Арк.	№ докум.	Підпис	Дата		

При виборі між монолітною та мікросервісною архітектурами необхідно враховувати не лише поточні потреби проєкту, а й його довгострокові перспективи розвитку. Для невеликих команд і проєктів із чітко визначеними та стабільними вимогами монолітний підхід може виявитися оптимальним через нижчу складність розробки та операційного управління. Натомість для складних багатомодульних систем із різнорідними вимогами до окремих компонентів, необхідністю інтеграції із зовнішніми сервісами та потребою в незалежному масштабуванні мікросервісна архітектура надає суттєві переваги, що виправдовують вищу початкову складність реалізації.

З огляду на багатомодульну структуру системи Digital University UA, необхідність інтеграції із зовнішнім державним сервісом «Дія» як окремого мікросервісу, а також вимоги до незалежного масштабування та розвитку кожного функціонального блоку, для реалізації було обрано мікросервісну архітектуру. Основний застосунок на базі Yii2 реалізує ключову бізнес-логіку управління кваліфікаційними роботами та опитуваннями здобувачів, тоді як окремий мікросервіс на Node.js відповідає за всю комунікацію з API сервісу «Дія» - отримання токенів автентифікації, генерацію deeplink для підписання, обробку вебхуків із підписаними документами та підвищення рівня підпису до формату CAdES-X-Long. Такий розподіл забезпечує ізоляцію ризиків - зміни в специфікації API «Дії» торкнуться лише одного ізольованого компонента, не впливаючи на стабільність основного застосунку. Цей підхід забезпечує необхідну гнучкість, технологічну незалежність модулів та стійкість системи до зростання навантаження, що є критично важливим для платформи університетського масштабу.

1.4. Інструментальні засоби розробки та вимоги до апаратного забезпечення

Після визначення архітектурного підходу постало завдання вибору інструментальних засобів для реалізації системи. Вибір технологічного стеку є одним із ключових рішень на етапі проєктування, оскільки від нього залежить не

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				18
Змн.	Арк.	№ докум.	Підпис	Дата		

лише швидкість початкової розробки, а й довгострокова підтримуваність системи, доступність кваліфікованих фахівців та можливість інтеграції з необхідними зовнішніми сервісами. При формуванні технологічного стеку враховувалися такі критерії: зрілість технології та наявність активної спільноти розробників, якість документації та екосистеми інструментів, відповідність технології специфіці розв'язуваних задач, а також сумісність із наявною інфраструктурою університету.

Як основну мову серверної розробки обрано РНР - зрілу та широко застосовувану мову веброботи з розвиненою екосистемою фреймворків, великою спільнотою та значним обсягом документації. РНР є однією з найпоширеніших мов для побудови серверної частини вебзастосунків - за різними оцінками, понад 75% вебсерверів у світі використовують РНР для обробки серверної логіки. Це забезпечує широку доступність фахівців, розгалужену екосистему готових рішень та бібліотек, а також тривалу підтримку з боку спільноти розробників.

Сучасні версії РНР демонструють значні покращення продуктивності порівняно з попередніми - починаючи з версії 7.0 продуктивність мови зросла вдвічі, а версії 8.x принесли JIT-компіляцію, що наближає РНР за швидкістю виконання до компільованих мов для певних типів обчислювальних задач. Строга типізація, підтримка іменованих аргументів, перелічень, файберів та інших сучасних конструкцій робить РНР повноцінною мовою для розробки складних корпоративних застосунків. Тісна інтеграція РНР з реляційними базами даних через PDO та ORM-бібліотеки, розвинена підтримка HTTP-протоколу та зручні інструменти для роботи з файлами роблять його обґрунтованим вибором для побудови складних вебзастосунків університетського рівня.

Основним фреймворком для бекенду визначено Yii2 - один із найпродуктивніших РНР-фреймворків із добре структурованою архітектурою та широкими можливостями для розширення функціоналу. Yii2 реалізує патерн MVC, надає потужний ORM ActiveRecord для роботи з базами даних, вбудовані механізми кешування, автентифікації та авторизації, а також зручну систему

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				19
Змн.	Арк.	№ докум.	Підпис	Дата		

міграцій бази даних. Фреймворк підтримує побудову як монолітних застосунків, так і API-серверів, що забезпечує гнучкість при еволюції архітектури системи.

Серед ключових переваг Yii2 для даного проєкту слід виділити розвинену систему ролей та дозволів RBAC, яка є основою рольової моделі доступу до кваліфікаційних робіт. Вбудований клас Query та ActiveRecord дозволяють будувати складні SQL-запити з об'єднанням таблиць, фільтрацією та сортуванням без написання «сирого» SQL, що підвищує безпеку та підтримуваність коду. Механізм поведінок та компонентів Yii2 спрощує повторне використання логіки між різними частинами системи. Розвинена система кешування з підтримкою різних бекендів - файлового, Memcached, Redis - дозволяє ефективно знижувати навантаження на базу даних при опрацюванні частих запитів до статистики опитувань та реєстру кваліфікаційних робіт.

Як систему управління базами даних використовується MySQL - одна з найпоширеніших реляційних СУБД у світі, що характеризується високою продуктивністю, надійністю та широкою підтримкою з боку хмарних провайдерів та інфраструктурних інструментів. MySQL забезпечує повну підтримку транзакцій через механізм InnoDB, що є критично важливим для збереження цілісності даних при зміні статусів кваліфікаційних робіт та збереженні результатів опитувань. Підтримка зовнішніх ключів дозволяє забезпечити референціальну цілісність між пов'язаними таблицями на рівні бази даних, а не лише на рівні застосунку.

Тісна інтеграція MySQL з PHP через розширення PDO та MySQLi, а також нативна підтримка в ActiveRecord Yii2 суттєво спрощує розробку та усуває потребу в додаткових шарах адаптації. Інструмент адміністрування phpMyAdmin надає зручний вебінтерфейс для управління структурою бази даних, перегляду та редагування даних, виконання SQL-запитів та експорту даних у різних форматах. Розвинена система індексування MySQL дозволяє оптимізувати продуктивність складних запитів до великих таблиць - зокрема при пошуку та фільтрації серед тисяч записів кваліфікаційних робіт.

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				20
Змн.	Арк.	№ докум.	Підпис	Дата		

Окремо розроблено мікросервіс для взаємодії з державним сервісом електронного підпису «Дія» на платформі Node.js. Вибір Node.js для цього компонента зумовлений кількома принциповими міркуваннями. По-перше, подієво-орієнтована модель виконання Node.js із неблокуючим введенням-виведенням ідеально підходить для задачі, що полягає переважно в очікуванні відповідей від зовнішнього API та обробці вхідних вебхуків. По-друге, Node.js забезпечує нативну підтримку WebSocket-з'єднань, що використовуються для сповіщення порталу про завершення підписання документа в режимі реального часу. По-третє, єдина мова JavaScript для серверної та клієнтської частини мікросервісу спрощує розробку та підтримку коду.

Node.js надає розвинену екосистему пакетів через npm, зокрема бібліотеку axios для HTTP-запитів до API «Дії», бібліотеку crypto для обчислення хешів файлів, а також драйвери для роботи з MySQL та MongoDB. Асинхронна природа Node.js дозволяє обслуговувати численні одночасні підключення з мінімальними накладними витратами, що є важливим при паралельному підписанні документів кількома учасниками процесу захисту кваліфікаційних робіт.

Коректна та стабільна робота розробленої системи залежить не лише від якості програмного забезпечення, а й від відповідності серверного обладнання встановленим технічним параметрам. Нижче наведено мінімальні вимоги до сервера для розгортання мікросервісу Node.js з урахуванням специфіки його роботи - інтенсивної мережевої взаємодії з API «Дії», обробки бінарних файлів підписів та підтримки WebSocket-з'єднань у режимі реального часу.

Щодо процесора - сервер повинен мати не менше чотирьох ядер із тактовою частотою не менше 3.00 GHz, підтримувати багатопоточність та мати архітектуру x86_64. Така конфігурація забезпечує достатню обчислювальну потужність для паралельної обробки запитів на підписання від кількох учасників, виконання криптографічних операцій через бібліотеку ПТ та підтримки активних WebSocket-з'єднань.

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				21
Змн.	Арк.	№ докум.	Підпис	Дата		

Щодо оперативної пам'яті - мінімальний обсяг RAM становить 32 GB, при цьому бажаною є підтримка технології ECC для підвищення надійності в умовах інтенсивного логування та передачі даних. Node.js утримує активні з'єднання та черги завдань у пам'яті, тому достатній обсяг RAM є критичним для стабільної роботи під навантаженням.

Щодо дискової підсистеми - необхідний SSD-накопичувач обсягом не менше 100 GB із зарезервованим простором для зберігання підписаних документів у форматі P7S, логів операцій підписання, тимчасових файлів у процесі обробки вебхуків та черг повідомлень RabbitMQ. Використання SSD замість HDD є обов'язковим з огляду на інтенсивні операції читання та запису файлів підписів.

Щодо операційної системи - рекомендованою є Linux на базі Debian або Alpine Linux. Сервер повинен підтримувати systemd або supervisord для автозапуску процесів мікросервісу після перезавантаження, а також мати можливість запуску у Docker-контейнерах або на віртуальній машині KVM для спрощення розгортання та ізоляції середовища виконання.

Щодо мережевої інфраструктури - гарантована пропускна здатність каналу має становити не менше 100 Mbps із забезпеченням стабільної внутрішньої комунікації між мікросервісом та основним сервером із затримкою менше 1 мс. Для коректної роботи мікросервісу необхідні відкриті порти для REST API та WebSocket-з'єднань - 80, 443 та порти діапазону 3000 і вище.

Додатково система повинна підтримувати резервне копіювання конфігурацій та журналів операцій підписання щонайменше раз на добу для забезпечення можливості відновлення після збоїв. Підтримка автоматичного розгортання через CI/CD-пайплайни - GitLab CI, GitHub Actions або подібні інструменти - є обов'язковою вимогою для забезпечення ефективного процесу оновлення мікросервісу без ручного втручання.

Наведені вимоги сформовані з урахуванням реального профілю навантаження системи - пікові значення виникають у періоди масового підписання документів кваліфікаційних робіт перед захистом та під час

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				22
Змн.	Арк.	№ докум.	Підпис	Дата		

проведення масових опитувань здобувачів наприкінці семестру. Дотримання зазначених технічних параметрів гарантує стабільну роботу системи в умовах реального університетського навантаження та забезпечує необхідний запас потужності для масштабування при зростанні кількості користувачів.

Висновки до першого розділу

У першому розділі було проведено детальний аналіз поставленого завдання, що дозволило чітко визначити мету та ключові цілі кваліфікаційної роботи. Особливу увагу приділено формуванню коректної постановки проблеми, адже саме це є відправною точкою для побудови ефективної та функціональної програмної системи. Визначення основних функціональних вимог до розроблюваного продукту стало важливим етапом у процесі планування, адже воно сприяло встановленню пріоритетів та розробці стратегічного підходу до реалізації.

Проведене дослідження аналогічних програмних рішень дало змогу глибше зрозуміти поточний стан ринку, оцінити наявні рішення та виділити їхні сильні й слабкі сторони. Це стало основою для обґрунтування шляхів удосконалення, які дозволять запропонованому продукту бути конкурентоспроможним та відповідати сучасним вимогам користувачів. У результаті аналізу було сформульовано рекомендації щодо покращення функціональних можливостей, підвищення ефективності роботи системи та забезпечення її адаптивності до майбутніх змін.

На основі отриманих висновків обрано оптимальну архітектуру, що забезпечує високу продуктивність, надійність та масштабованість системи. Обґрунтовано вимоги до апаратного забезпечення, необхідного для стабільної роботи системи. Було визначено оптимальні характеристики серверного обладнання, що забезпечать безперебійну роботу в умовах реальних навантажень та гарантують належний рівень продуктивності.

Таким чином, виконаний аналіз та прийняті рішення формують надійну концептуальну базу для наступних етапів розробки. Визначені напрямки

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				23
Змн.	Арк.	№ докум.	Підпис	Дата		

вдосконалення, обрані інструменти та розроблена загальна архітектура системи забезпечують ефективний старт для подальшої реалізації програмного продукту, сприяючи його відповідності сучасним технологічним вимогам та очікуванням користувачів.

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		24

РОЗДІЛ 2. ПРОЕКТУВАННЯ СИСТЕМИ ОНЛАЙН-МЕНЕДЖМЕНТУ ОСВІТНІХ ПРОЦЕСІВ

2.1. Визначення варіантів використання та структури системи

Для забезпечення надійного функціонування модулів «Кваліфікаційні роботи» та «Опитування здобувачів освіти» в межах інформаційної системи Digital University UA сформульовано комплекс вимог до розробки, що охоплюють як функціональні, так і нефункціональні характеристики, спрямовані на підтримку високої ефективності, безпеки, масштабованості та сумісності з іншими підсистемами.

До основних бізнес-цілей розробки можна віднести наступне. Модулі створюються з метою автоматизації та цифровізації двох ключових процесів університетської діяльності. Модуль «Кваліфікаційні роботи» забезпечує централізоване управління інформацією про кваліфікаційні роботи здобувачів, координацію між усіма учасниками процесу підготовки та захисту, а також юридично значуще електронне підписання документів через інтеграцію з державним сервісом «Дія». Модуль «Опитування здобувачів освіти» орієнтований на організацію, проведення та аналіз опитувань здобувачів з метою отримання систематизованого зворотного зв'язку щодо якості освітнього процесу. Система реалізується у вигляді вебзастосунку з мікросервісною архітектурою - серверна частина побудована на основі PHP-фреймворку Yii2, вся інформація зберігається в централізованій базі даних MySQL, а для взаємодії з API державного сервісу «Дія» використовується окремий мікросервіс на Node.js, який обробляє запити на ініціювання та підтвердження електронного підпису документів кваліфікаційних робіт.

Вимоги адміністрації стосуються обох модулів системи. Для модуля «Кваліфікаційні роботи» адміністрація потребує можливості перегляду реєстру кваліфікаційних робіт із фільтрацією та сортуванням за спеціальністю, науковим керівником, кафедрою та поточним статусом роботи. Критично важливим є доступ до повного комплексу документів кожного здобувача, відстеження статусу підписання документів кожним із учасників процесу через сервіс «Дія»,

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		25

а також можливість експорту інформації у форматі Excel для формування офіційної звітності. Для модуля «Опитування» адміністрація має мати змогу створювати опитування з гнучким налаштуванням питань і цільової аудиторії, переглядати зведені результати у розрізі спеціальностей та підрозділів, а також формувати аналітичні звіти за підсумками проведених опитувань.

Власники системи, у свою чергу, очікують наявності гнучких налаштувань ключових параметрів та можливості керування доступом до модулів відповідно до ролей користувачів. Важливими вимогами є фіксація всіх дій персоналу та здобувачів у системному журналі, а також регулярне створення резервних копій бази даних для забезпечення збереження інформації та можливості відновлення системи у разі збою.

Об'єктом комп'ютеризації є процеси управління кваліфікаційними роботами та проведення опитувань здобувачів освіти в закладі вищої освіти. Основними користувачами модулів є науково-педагогічний персонал, адміністрація кафедр і деканатів, технічні оператори системи, а також самі здобувачі освіти, що взаємодіють із системою через особистий кабінет. Доступ до системи здійснюється через браузер, що усуває потребу у встановленні додаткового програмного забезпечення на пристрої користувача.

До функціональних вимог системи належить забезпечення авторизації користувачів із подальшим призначенням відповідних ролей. Модуль «Кваліфікаційні роботи» має підтримувати формування та ведення реєстру робіт, ініціювання процесу підписання документів через API сервісу «Дія», відстеження статусу підписання в режимі реального часу, генерацію супровідних документів у форматі Word та збереження підписаних документів у системі. Модуль «Опитування» повинен забезпечувати створення опитувань із різними типами питань, анонімне збирання відповідей від цільової аудиторії та автоматизоване формування аналітичних звітів.

Нефункціональні вимоги охоплюють кілька ключових аспектів. З точки зору сприйняття, система повинна бути достатньо простою для повноцінного освоєння оператором протягом одного навчального заняття тривалістю до 60

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				26
Змн.	Арк.	№ докум.	Підпис	Дата		

хвилин, відгук інтерфейсу на основні дії не повинен перевищувати 2 секунди, тоді як для складних аналітичних запитів допустима затримка до 10 секунд, а всі форми мають містити підказки до полів для зменшення кількості помилок під час введення даних.

З точки зору надійності, система повинна забезпечувати доступність не менше ніж 96% часу протягом доби, відновлення роботи після збою має відбуватися за час, що не перевищує 15 хвилин, а взаємодія з API сервісу «Дія» має супроводжуватися механізмами повторних спроб та журналюванням усіх запитів і відповідей для гарантування цілісності процесу підписання.

З точки зору продуктивності, система має підтримувати одночасну роботу понад 200 користувачів без зниження продуктивності, ефективно обробляти великі обсяги даних під час фільтрації, сортування та експорту, а інтерфейс повинен бути адаптивним і коректно відображатися на різних пристроях, зокрема на смартфонах і планшетах.

З точки зору експлуатаційних характеристик, система повинна мати модульну структуру, що дозволяє легко розширювати її функціональність без суттєвого рефакторингу існуючого коду, забезпечувати ведення журналу подій та механізми резервного копіювання, а мікросервіс інтеграції з «Дією» має підтримувати незалежне оновлення без будь-якого впливу на основну систему.

Після дослідження усіх необхідних характеристик майбутнього програмного забезпечення та виокремлення ключових функцій, що потребують актори системи, було створено діаграму використання (рис. 2.1):

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				27
Змн.	Арк.	№ докум.	Підпис	Дата		

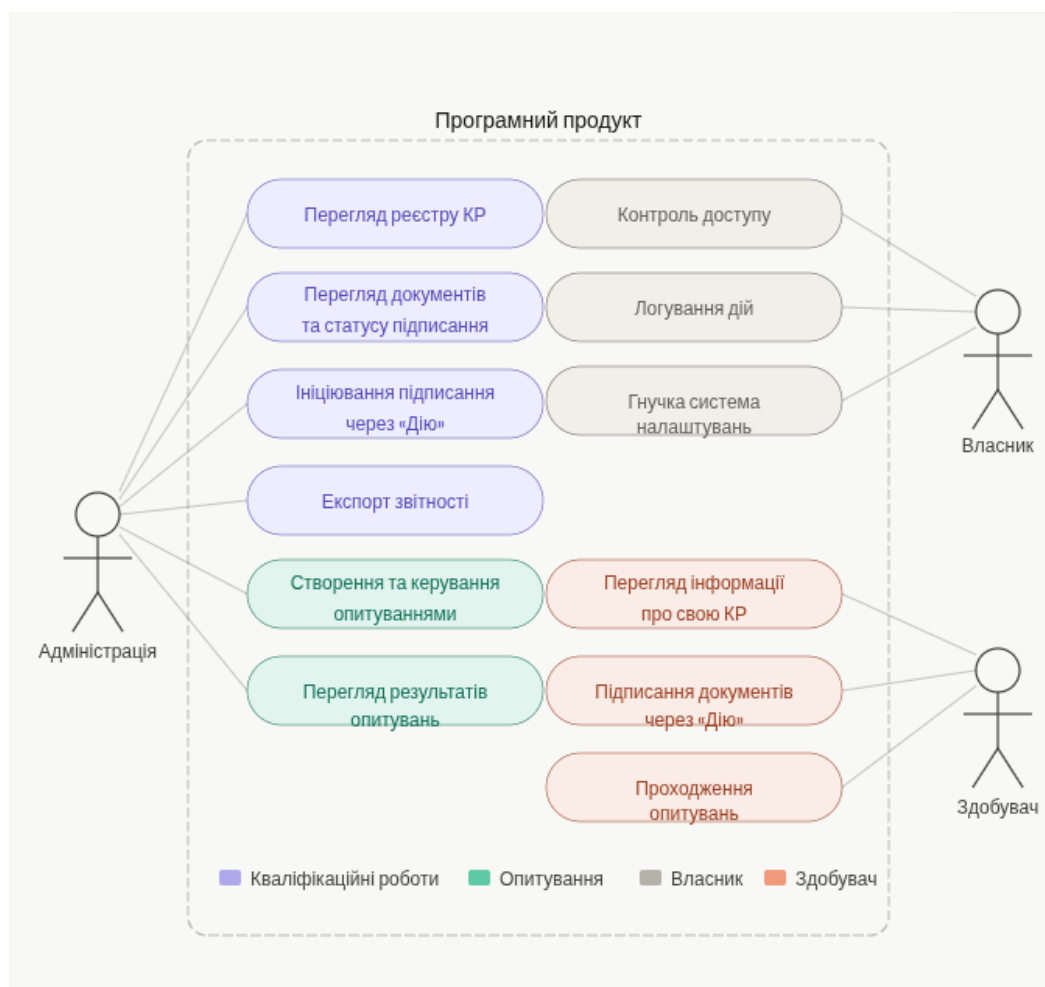


Рисунок 2.1 - Діаграма використання системи

Головним користувачем системи є адміністрація - група науково-педагогічних працівників і співробітників університету, наділених різними ролями відповідно до своїх повноважень. Вебпортал надає адміністраторам можливість самостійно створювати нові ролі та визначати їхні привілеї, що дозволяє гнучко масштабувати зону відповідальності кожної з них. Модуль «Кваліфікаційні роботи» охоплює функціонал управління реєстром робіт, координації підписання документів через «Дію» та формування звітності, тоді як модуль «Опитування здобувачів освіти» забезпечує повний цикл роботи з опитуваннями - від їх створення до аналізу результатів. Актор «Здобувач освіти» взаємодіє із системою через особистий кабінет, де має доступ до інформації про свою кваліфікаційну роботу, може підписувати документи через «Дію» та проходити призначені опитування. Актор «Власник» забезпечує технічну підтримку платформи, має можливість обмежувати доступ на період технічних

робіт, переглядати журнал дій персоналу та налаштовувати систему без прямого втручання у код.

2.2. Обґрунтування вибору системи управління базами даних

Вибір системи управління базами даних є одним із фундаментальних архітектурних рішень при проєктуванні інформаційної системи, оскільки від нього залежить не лише продуктивність та надійність зберігання даних, а й складність реалізації бізнес-логіки, можливості масштабування та довгострокова підтримуваність системи. При виборі СУБД для інформаційної системи Digital University UA враховувалися специфіка предметної області, характер даних, що зберігаються, вимоги до цілісності та безпеки інформації, а також технологічна сумісність з обраним стеком розробки.

У процесі реалізації інформаційної системи обґрунтовано доцільність використання системи керування базами даних MySQL, а для адміністрування - інструмента phpMyAdmin, який забезпечує інтуїтивно зрозуміле середовище для взаємодії з базою даних через вебінтерфейс. Зазначений вибір зумовлений низкою технологічних та практичних переваг MySQL у контексті веборієнтованих застосунків університетського рівня.

Перш за все, варто розглянути альтернативні рішення, що були проаналізовані в процесі вибору СУБД. Серед реляційних систем управління базами даних на ринку присутні PostgreSQL, Microsoft SQL Server, Oracle Database та SQLite, кожна з яких має власні переваги та обмеження. PostgreSQL є потужною об'єктно-реляційною СУБД із широкими можливостями розширення, підтримкою складних типів даних та розвиненими аналітичними функціями. Однак для задач даного проєкту розширений функціонал PostgreSQL є надлишковим, а крива освоєння та вимоги до налаштування є складнішими порівняно з MySQL. Microsoft SQL Server та Oracle Database є комерційними рішеннями з відповідними ліцензійними витратами, що суперечить принципу використання відкритого програмного забезпечення в університетських проєктах. SQLite є вбудованою базою даних без підтримки паралельного доступу

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		29

від кількох клієнтів, що робить її непридатною для системи з понад 200 одночасними користувачами.

MySQL характеризується високою продуктивністю, надійністю та гнучкістю масштабування, що дозволяє ефективно обробляти як невеликі, так і великі обсяги структурованих даних. Рушій InnoDB, що використовується за замовчуванням у сучасних версіях MySQL, забезпечує повну підтримку транзакцій відповідно до властивостей ACID - атомарності, узгодженості, ізольованості та довговічності. Ця властивість є критично важливою для модуля «Кваліфікаційні роботи», де зміна статусу роботи та збереження відповідного запису в історії змін мають відбуватися як єдина неподільна операція. У разі будь-якої помилки в процесі виконання транзакції система автоматично відкочує всі зміни до попереднього стану, гарантуючи цілісність даних навіть при збоях у роботі сервера.

Підтримка зовнішніх ключів у MySQL дозволяє забезпечити референціальну цілісність між пов'язаними таблицями на рівні бази даних. Наприклад, запис про підписання документа не може посилатися на неіснуючу кваліфікаційну роботу, а відповідь здобувача в опитуванні не може бути пов'язана з неіснуючим питанням. Ця властивість суттєво знижує ризик появи «осирілих» записів та забезпечує логічну узгодженість даних навіть при паралельних операціях з боку кількох користувачів.

Ці властивості є критично важливими при роботі з інформацією про кваліфікаційні роботи, документами підписання та результатами опитувань, де передбачається постійне оновлення й зберігання значної кількості записів. База даних системи Digital University UA містить десятки взаємопов'язаних таблиць - від реєстру кваліфікаційних робіт та їхніх статусів до таблиць опитувань, питань, відповідей та записів про участь здобувачів. Коректна обробка таких складних реляційних структур вимагає надійної СУБД із розвиненими механізмами забезпечення цілісності даних.

Розвинена система індексування MySQL дозволяє суттєво підвищити продуктивність запитів до великих таблиць. Первинні ключі, унікальні індекси

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		30

та складені індекси за кількома полями дозволяють прискорити пошук і фільтрацію записів у реєстрі кваліфікаційних робіт, навіть якщо він містить десятки тисяч записів. Механізм EXPLAIN у MySQL дозволяє аналізувати план виконання запитів та виявляти потенційні вузькі місця продуктивності на етапі розробки, до розгортання системи в продуктивне середовище. Підтримка партиціонування таблиць відкриває можливість горизонтального масштабування бази даних при зростанні обсягу даних у майбутньому.

Вибір MySQL також зумовлений її тісною інтеграцією з мовою PHP, обраною для розробки серверної частини системи. Розширення MySQLi та PDO забезпечують безпечну взаємодію з базою даних через підготовлені вирази, що є основним механізмом захисту від SQL-ін'єкцій. Фреймворк Yii2 надає потужний ORM ActiveRecord, побудований поверх PDO, що дозволяє працювати з даними MySQL через об'єктно-орієнтований інтерфейс без написання «сирого» SQL для більшості типових операцій. Така технологічна сумісність суттєво спрощує реалізацію логіки обміну даними між прикладною частиною та сховищем, мінімізуючи витрати на реалізацію базових CRUD-операцій та знижуючи ймовірність помилок при роботі з даними.

Клас Query у Yii2 дозволяє будувати складні SQL-запити з об'єднанням таблиць, умовами фільтрації, сортуванням та агрегацією в зрозумілий об'єктно-орієнтований спосіб. Наприклад, запит для отримання статистики проходження опитувань по групах із підрахунком кількості учасників та відсотка залученості будується через ланцюжок викликів методів select, from, innerJoin, where, groupBy та indexBy, що є значно безпечнішим та зручнішим підходом порівняно з конкатенацією рядків SQL. Механізм міграцій Yii2 дозволяє версіонувати зміни структури бази даних разом із кодом застосунку, що суттєво спрощує розгортання оновлень у різних середовищах та відкочування змін у разі проблем.

Застосування phpMyAdmin дозволило оперативно виконувати конфігурацію структури бази даних на етапі розробки, включно зі створенням таблиць, налаштуванням типів та атрибутів полів, встановленням зовнішніх ключів між сутностями та управлінням правами доступу користувачів бази

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		31

даних. Візуальний інтерфейс phpMyAdmin суттєво прискорив початковий етап проєктування схеми бази даних, дозволяючи одразу бачити результати структурних змін та перевіряти коректність зв'язків між таблицями. Інструмент також надає зручні засоби для імпорту та експорту даних у різних форматах, що використовувалося при розробці для наповнення бази тестовими даними та резервного копіювання стану бази на різних етапах розробки.

Завдяки гнучким механізмам безпеки MySQL забезпечує надійний захист конфіденційних даних здобувачів від несанкціонованого доступу. Система привілеїв MySQL дозволяє створювати окремих користувачів бази даних із різними рівнями доступу - наприклад, застосунок підключається до бази з правами, необхідними лише для виконання CRUD-операцій, без можливості зміни структури таблиць або видалення бази даних. Це відповідає принципу найменших привілеїв та суттєво знижує потенційний збиток у разі компрометації облікових даних застосунку. Підтримка SSL/TLS шифрування з'єднання між застосунком та сервером бази даних забезпечує захист даних при передачі по мережі, що є особливо важливим в умовах університетської інфраструктури з розподіленими серверними вузлами.

MySQL також надає вбудовані механізми для резервного копіювання даних через утиліту `mysqldump`, яка дозволяє створювати повні та інкрементальні резервні копії бази даних у форматі SQL-скриптів. Автоматизація резервного копіювання за розкладом забезпечує можливість відновлення даних у разі апаратних збоїв або людських помилок, що є критично важливим для системи, що зберігає юридично значущу інформацію про кваліфікаційні роботи та підписані документи.

2.3. Інтеграція зовнішніх сервісів

Ключовою технологічною особливістю модуля «Кваліфікаційні роботи» є інтеграція із зовнішнім державним сервісом електронного підпису «Дія». Підписання документів, пов'язаних із підготовкою та захистом кваліфікаційних

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		32

робіт, є юридично значущою операцією, що потребує надійного, відстежуваного та безпечного механізму взаємодії між системою університету та API «Дії».

Для реалізації цього процесу розроблено окремий мікросервіс на Node.js, який відповідає за всю комунікацію з API сервісу «Дія». Оскільки діаграма послідовності підписання документів є розгорнутою та містить значну кількість взаємодій між компонентами системи, для зручності сприйняття її подано у трьох частинах. Перша частина діаграми ілюструє ініціювання процесу підписання та взаємодію між порталом і мікросервісом (рис. 2.2):

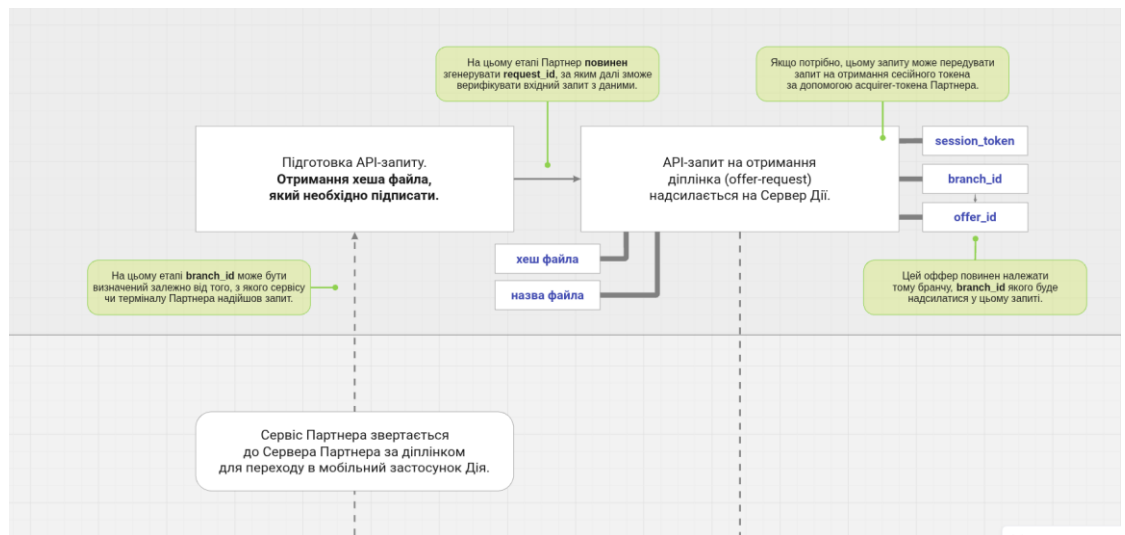


Рисунок 2.2 - Діаграма послідовності підписання документів через сервіс «Дія» (частина 1)

Перша частина охоплює момент, коли користувач ініціює підписання на порталі - система формує запит до мікросервісу Node.js, який обчислює хеш-суму документа за алгоритмом SHA-256 та надсилає її до API «Дії». Передача хешу замість самого файлу є принциповим рішенням - це мінімізує обсяг трафіку та забезпечує конфіденційність вмісту документа, оскільки сам файл кваліфікаційної роботи не виходить за межі серверної інфраструктури університету. Державний сервіс «Дія» опрацьовує надісланий хеш, реєструє нову транзакцію підписання та повертає у відповідь тимчасовий deeplink і QR-код. Ця інформація транслюється мікросервісом назад до Yii2-застосунку і відображається у кабінеті учасника процесу. Друга частина діаграми відображає

взаємодію користувача з мобільним застосунком «Дія» та отримання підтвердження підписання (рис. 2.3):

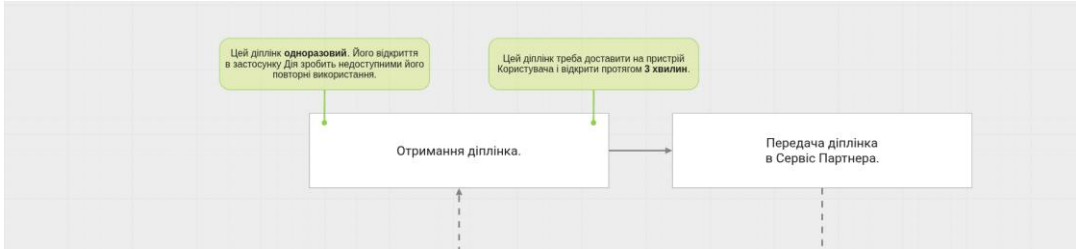


Рисунок 2.3 - Діаграма послідовності підписання документів через сервіс «Дія» (частина 2)

На цьому етапі користувач, відкривши сторінку підписання, бачить QR-код для настільних комп'ютерів або кнопку-посилання для мобільних пристроїв. При скануванні коду або переході за deeplink на пристрої автоматично відкривається застосунок «Дія», де користувач підтверджує свою особу за допомогою FaceID або коду авторизації та дає згоду на накладання електронного підпису. Після успішного підписання сервіс «Дія» надсилає асинхронне підтвердження на заздалегідь налаштований вебхук мікросервісу Node.js. Цей шлюз постійно очікує відповіді у фоновому режимі та захищений від зловмисних запитів шляхом прийняття звернень виключно з фіксованих IP-адрес Міністерства цифрової трансформації України. Третя частина діаграми демонструє обробку отриманого підпису та збереження результатів у базі даних (рис. 2.4):

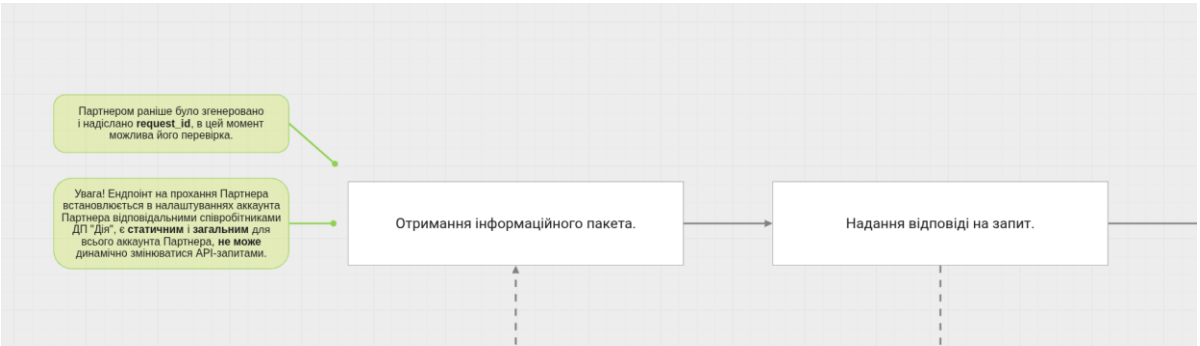


Рисунок 2.4 - Діаграма послідовності підписання документів через сервіс «Дія» (частина 3)

Отриманий інформаційний пакет містить криптоконтейнер із накладеним підписом та зашифрованими метаданими сертифіката користувача. Для забезпечення легітимності та коректної обробки отриманого пакета мікросервіс

Node.js задіює спеціалізовану сертифіковану криптографічну бібліотеку від АТ «Інститут інформаційних технологій» (ІІТ). Використання саме цієї бібліотеки є критично важливим, оскільки вона забезпечує повну сумісність із державними стандартами криптографічного захисту інформації України.

Процес обробки отриманого пакета охоплює кілька послідовних етапів. Спочатку перевіряється цілісність самого вебхука та цифровий підпис серверів «Дія», що унеможливорює підміну результату підписання. Після успішної валідації джерела бібліотека ІІТ проводить декомпозицію отриманого криптоконтейнера, вилучає базовий підпис та перевіряє його статус за допомогою протоколу OCSP або списків відкликаних сертифікатів у режимі реального часу. Завершальним і найважливішим кроком є удосконалення підпису до формату CAdES-X-Long - для цього здійснюється запит до акредитованого центру сертифікації ключів для отримання сертифікованої позначки часу та повних даних про статус сертифіката. Цей формат гарантує, що навіть через багато років, коли особистий сертифікат здобувача чи викладача закінчиться, архівний документ залишиться юридично чинним, а підпис можна буде успішно верифікувати.

Для забезпечення універсальності та безперебійності освітнього процесу в системі реалізовано двокомпонентну модель підписання. Окрім сервісу «Дія.Підпис», користувачам доступний альтернативний класичний сценарій - використання кваліфікованого електронного підпису через файлові носії або апаратні токени. Цей сценарій розгорнутий безпосередньо на стороні клієнтського інтерфейсу вебпорталу за допомогою JavaScript-модуля бібліотеки ІІТ. Процес відбувається повністю в браузері користувача - інтерфейс пропонує завантажити файл особистого ключа у форматі dat, pfx або zs2 або підключити апаратний захищений носій до USB-порту. Після введення пароля захисту локальний криптографічний модуль ІІТ здійснює розбір структури ключа, визначає кваліфікованого надавача електронних довірчих послуг, що видав сертифікат, та надсилає запит на перевірку його чинності. Підпис під обчисленим хешем документа формується безпосередньо в браузері та у вигляді готового

		Коновал М.М.			ІІЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		35

криптоконтейнера відправляється на сервер університету, минаючи мікросервіс Node.js. Така гібридна модель суттєво підвищує відмовостійкість системи - якщо один із каналів стає недоступним через технічні роботи, користувачі можуть завершити підписання класичним методом, не зупиняючи графік роботи атестаційної комісії.

Після того як мікросервіс Node.js або клієнтська бібліотека сформували фінальний файл із підписом CAdES-X-Long, дані повертаються до ядра системи на базі Yii2 для остаточного збереження та синхронізації статусів у реляційній базі даних MySQL. На цьому етапі для кожного документа виконується перевірка наявності відповідного запису в реєстрі - якщо запис про підписання цієї версії файла відсутній, він створюється як нова сутність; якщо існує, наприклад у разі повторного підписання або виправлення помилок, його метадані оновлюються.

Усі операції модифікації даних, завантаження фізичного файлу у захищене сховище та оновлення статусів відбуваються виключно в межах ізольованої ACID-транзакції бази даних під керуванням ORM Active Record. Якщо на будь-якому проміжному етапі - чи то під час фізичного запису файлу на диск сервера, чи при спробі оновити запис у MySQL - станеться системний збій або мережева помилка, вся транзакція буде негайно скасована командою відкату змін. У такому разі система повертається до свого початкового стабільного стану, що передував спробі підписання, повністю виключаючи появу неузгоджених статусів або документів, що втратили зв'язок із відповідними записами бази даних.

Таким чином, описаний механізм забезпечує надійний, послідовний та юридично коректний процес електронного підписання документів кваліфікаційних робіт як через державний сервіс «Дія», так і через класичні кваліфіковані електронні підписи, що є визначальною перевагою розробленої системи порівняно з наявними аналогами. Гнучке поєднання асинхронної обробки завдань через RabbitMQ, ізольованого мікросервісу на Node.js та сертифікованого криптографічного ядра ІТ дозволило створити стійку до

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		36

навантажень інфраструктуру, що повністю відповідає нормативним вимогам законодавства України у сфері електронних довірчих послуг..

2.4. Реалізація функціоналу системи управління навчальним процесом

Розглянемо код побудованого програмного забезпечення. Почнемо огляд із модуля «Опитування здобувачів освіти», а далі перейдемо до модуля «Кваліфікаційні роботи» з інтеграцією сервісу «Дія».

Основним контролером для управління опитуваннями на порталі є PollController, що реалізує повний цикл CRUD-операцій. При створенні нового опитування встановлюються значення за замовчуванням - тип опитування та тип відображення питань:

```
$model = new Poll();  
$model->type = PollAlias::TYPE_GENERAL;  
$model->display_type = PollAlias::DISPLAY_TYPE_SINGLE;
```

Тип опитування визначає, чи є воно загальним або прив'язаним до навчальних дисциплін, а тип відображення - чи показуються питання посторінково або всі одразу. Від цих двох параметрів залежить уся подальша логіка рендерингу форми опитування як на порталі, так і в кабінеті здобувача, що робить їх критично важливими на етапі створення. Загальне опитування відображає єдину форму з усіма питаннями, тоді як опитування за освітніми компонентами потребує окремого проходження для кожної навчальної дисципліни здобувача. Після успішного збереження моделі відбувається перенаправлення до списку опитувань, у разі помилки - повторне відображення форми зі збереженими даними та повідомленнями про помилки валідації.

Система підтримує копіювання опитувань із усіма зв'язками. Спочатку зберігається нова модель, після чого копіюються зв'язки з групами:

```
$transaction = Yii::$app->db->beginTransaction();  
try {  
    if ($newModel->save()) {  
        $groups = $db->createCommand(  
            "SELECT group_id FROM poll_group WHERE poll_id = :id",  
            [':id' => $id]  
        )->queryColumn();
```

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Вакалюк Т.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		37

```

foreach ($groups as $groupId) {
    $db->createCommand()
        ->insert('poll_group', [
            'poll_id' => $newModel->id,
            'group_id' => $groupId
        ])->execute();
    }
    $transaction->commit();
}
} catch (Throwable $e) {
    $transaction->rollBack();
}

```

Аналогічним чином копіюються зв'язки з окремими здобувачами та освітніми програмами. У разі будь-якої помилки виконується rollBack(), що гарантує відсутність неповних копій у системі. Варто зазначити, що при великій кількості груп або здобувачів копіювання виконується в циклі по одному запиту, що може створювати навантаження на базу даних - це свідоме архітектурне рішення, яке забезпечує простоту коду ціною дещо вищого навантаження при копіюванні великих опитувань.

Для визначення цільової аудиторії опитування сервіс PollTargetService реалізує багаторівневу логіку пріоритетності у методі getTargetGroupIds. Насамперед перевіряється наявність результату в кеші:

```

$cacheKey = $poll->id . md5(serialize($filters));
if (isset(self::$_groupIdsCache[$cacheKey])) {
    return self::$_groupIdsCache[$cacheKey];
}

```

Якщо результат відсутній у кеші, виконується запит з урахуванням пріоритетності призначення опитування:

```

$query = AcademicGroup::find()->alias('ag')->select('ag.id');
if (!empty($poll->pollStudents)) {
    $query->innerJoin('student s', 's.academic_group_id = ag.id')
        ->andWhere(['s.education_id' => array_column($poll->pollStudents, 'student_id')]);
} elseif (!empty($poll->pollGroups)) {
    $query->andWhere(['ag.id' => array_column($poll->pollGroups, 'group_id')]);
} elseif (!empty($poll->pollStudyPrograms)) {

```

		Коновал М.М.			ІІЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Вакалюк Т.А.				38
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        $query->andWhere(['ag.study_program_id' => array_column($poll->pollStudyPrograms,
'study_program_id')]);
    }
    return self::$_groupIdsCache[$cacheKey] = $query->column();

```

Якщо опитування призначене конкретним здобувачам - фільтрується за ними, якщо групам - за групами, якщо освітнім програмам - за програмами. Лише за відсутності всіх вищезазначених обмежень застосовуються загальні параметри факультету та форми навчання з налаштувань самого опитування. Неправильний порядок цих перевірок призводив би до показу опитування непризначеним здобувачам, тому послідовність умов є принциповою. Результат зберігається в масиві `$_groupIdsCache` за ключем, що є комбінацією ідентифікатора опитування та MD5-хешу параметрів фільтрації, що запобігає повторним важким запитами при багаторазовому зверненні до методу в межах одного запиту.

Для отримання статистики проходження по групах метод `getGroupStats` виконує два окремі запити до бази даних. Перший підраховує загальну кількість здобувачів по кожній групі, другий - кількість унікальних здобувачів, що пройшли опитування:

```

$passedCounts = (new Query())
    ->select(['s.academic_group_id', 'COUNT(DISTINCT p.student_id) AS passed'])
    ->from(['p' => 'poll_student_answer_participation'])
    ->innerJoin(['s' => Student::tableName()], 's.education_id = p.student_id')
    ->where(['p.poll_id' => $poll->id, 's.academic_group_id' => $groupIds])
    ->groupBy('s.academic_group_id')
    ->indexBy('academic_group_id')
    ->all();

```

Після об'єднання результатів формується масив із показниками `total`, `passed` та `not_passed` для кожної групи. Використання `COUNT(DISTINCT p.student_id)` замість простого `COUNT` є принциповим - воно гарантує коректний підрахунок навіть якщо здобувач відповів на кілька питань, оскільки кожна відповідь створює окремий запис у таблиці участі. Важливим нюансом є використання `max(0, $total - $passed)` для поля `not_passed` - це захист від ситуації, коли кількість

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		39

відповідей перевищує кількість зареєстрованих здобувачів через можливі розбіжності в даних між таблицями.

Сервіс PollExportService формує Excel-файл із результатами опитування, що містить кілька аркушів із різними зрізами даних. Середнє значення числових відповідей обчислюється безпосередньо під час ітерації по рядках результату запиту - для кожної комбінації групи, дисципліни та питання накопичуються сума та кількість відповідей:

```
if (is_numeric($row['answer_value'])) {
    $key = $row['group_id'] . '_' . ($row['discipline_name'] ?? '') . '_' . $row['question_id'];
    $averages[$key]['sum'] += $row['answer_value'] * $row['answer_count'];
    $averages[$key]['count'] += $row['answer_count'];
}
```

Після завершення ітерації по всіх рядках середнє обчислюється як $\text{round}(\text{\$averages}[\text{\$key}][\text{'sum'}] / \text{\$averages}[\text{\$key}][\text{'count'}], 2)$. Такий підхід дозволяє уникнути додаткового запиту до бази даних для обчислення агрегованих значень - вся необхідна інформація вже присутня в результаті основного запиту. Щоб уникнути N+1 проблеми при отриманні метаданих груп для кожного рядка звіту, результати кешуються у масиві `$this->groupInfoCache` - повторні звернення до тієї самої групи повертають вже отриманий результат без додаткового запиту до бази даних.

На стороні здобувача контролер frontend\PollController відображає список доступних опитувань. Ключовою особливістю запиту є використання обчислюваних полів через конструкцію CASE WHEN для визначення статусу кожного опитування в одному зверненні до бази даних:

```
$polls = Poll::find()
->alias('p')
->select([
    'p.*',
    'COUNT(pfq.id) as questionCount',
    'COUNT(psap.id) as answeredCount',
    "CASE WHEN COUNT(psap.id) = COUNT(pfq.id)
        AND COUNT(pfq.id) != 0
        THEN 1 ELSE 0 END as isCompleted",
```

		Коновал М.М.			ІІЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Вакалюк Т.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		40

```
"CASE WHEN COUNT(psap.id) > 0
  THEN 1 ELSE 0 END as isStarted",
"CASE WHEN NOW() BETWEEN date_begin AND date_end
  THEN 1 ELSE 0 END as isActive",
```

)

Поле isCompleted визначається як рівність кількості відповідей і кількості питань форми, isStarted - як наявність хоча б однієї відповіді, isActive - через перевірку NOW() BETWEEN date_begin AND date_end. Обчислення всіх трьох статусів в одному запиті суттєво знижує кількість звернень до бази даних порівняно з послідовними запитам для кожного опитування. Для опитувань типу «за дисциплінами» статус завершеності обчислюється окремо після основного запиту - кількість відповідей порівнюється з добутком кількості питань на кількість дисциплін конкретного здобувача, оскільки кожна дисципліна потребує окремого проходження форми.

При збереженні відповідей здобувача виконується кілька рівнів валідації. Спочатку перевіряється чинність опитування для поточного здобувача, потім - коректність ідентифікаторів дисциплін для опитувань за освітніми компонентами. Для числових відповідей виконується явне приведення типу для запобігання збереженню некоректних значень:

```
if ($answerType === 'answer_id') {
    $studentAnswer->$answerType = (int) $item['answer'];
} else {
    $studentAnswer->$answerType = $item['answer'];
}
```

Усі записи зберігаються в межах єдиної транзакції. Для кожної відповіді послідовно створюються два записи - PollStudentAnswer із самою відповіддю та PollStudentAnswerParticipation із поточним часом проходження:

```
$participation = new PollStudentAnswerParticipation();
$participation->poll_id = $id;
$participation->student_id = $educationId;
$participation->datetime = date('Y-m-d H:i:s');
$participation->save();
```

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Вакалюк Т.А.				41
Змн.	Арк.	№ докум.	Підпис	Дата		

Наявність окремої таблиці участі дозволяє точно відстежувати не лише факт відповіді на конкретне питання, а й час проходження, що є важливим для аналітики та звітності. У разі будь-якої помилки транзакція відкочується, що гарантує відсутність часткових записів відповідей у базі даних.

Контролер `QualificationWorkViewController` є центральним компонентом модуля «Кваліфікаційні роботи». Навігація між статусами реалізована через конструкцію `match`, що викликає відповідний метод сервісу залежно від поточного стану роботи:

```
match ($work->status_id) {
    1 => $this->qualificationWorkService->method1($qualificationWorkId, $data),
    50 => $this->qualificationWorkService->method50($qualificationWorkId, $data),
    100 => $this->qualificationWorkService->method100($qualificationWorkId, $status, $data),
    200 => $this->qualificationWorkService->method200($qualificationWorkId, $status, $data),
    800 => $this->qualificationWorkService->method800($qualificationWorkId, $status, $data),
    3000, 4000, 5000 => $this->qualificationWorkService->methodFinalStatuses(
        $qualificationWorkId, $work->status_id),
};
```

Кожен статус відповідає конкретному етапу: статус 1 - первинне завантаження роботи здобувачем, 100 - перевірка науковим керівником, 150 та 175 - додаткові перевірки для магістрів з охорони праці та екологічного консультанта, 200 - нормоконтроль, 300 - антиплагіат, 400 - погодження завідувачем кафедри, 500-600 - рецензування та допуск до захисту, 700-800 - захист, 3000-5000 - фінальні статуси. Використання конструкції `match` замість ланцюжка `if-elseif` забезпечує суворе порівняння типів та генерує виняток при отриманні статусу, що не передбачений жодним із варіантів, що запобігає мовчазній обробці некоректних станів. Такий підхід також дозволяє легко розширювати кількість етапів у майбутньому без зміни структури контролера - достатньо додати новий рядок до конструкції `match` та реалізувати відповідний метод у сервісі.

Сервіс `QualificationWorkService` реалізує рольову модель доступу. Метод `getAvailableQualificationWorks` спочатку перевіряє, чи є поточний користувач адміністратором - у такому разі повертаються всі роботи без фільтрації:

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		42

```

if ($this->isAdmin()) {
    return $this->getAllQualificationWorks();
}

```

Для інших ролей результати накопичуються через `array_merge` з подальшою дедуплікацією:

```

if ($this->isDean() || $this->isDeputyDean() || $this->isDeanOfficeEmployee()) {
    $result = array_merge($result, $this->getDeanQualificationWorks());
}
if ($this->isTeacher()) {
    $result = array_merge($result, $this->getTeacherQualificationWorks());
}
$uniqueWorks = [];
foreach ($result as $work) {
    $uniqueWorks[$work['id']] = $work;
}
return array_values($uniqueWorks);

```

Дедуплікація за ідентифікатором роботи є критично важливою, оскільки користувач може одночасно мати кілька ролей - наприклад, бути і завідувачем кафедри, і членом екзаменаційної комісії. Без дедуплікації одна й та сама робота могла б потрапити до результату кілька разів, що призвело б до некоректного відображення списку на порталі.

Зміна статусу роботи завжди виконується через приватний метод `changeWorkStatus`, який у межах єдиної транзакції оновлює статус моделі та зберігає запис в журналі змін:

```

$work->status_id = $newStatusId;
$work->save();
$history = new GraduationWorkStatusHistory();
$history->graduation_work_id = $workId;
$history->datetime = new Expression('NOW()');
$history->status_id = $newStatusId;
$history->save();
$transaction->commit();

```

Використання `new Expression('NOW()')` замість `date('Y-m-d H:i:s')` є принциповим - час фіксується на стороні сервера бази даних, що виключає розбіжності при роботі в різних часових поясах або при некоректно

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		43

налаштованому системному часі на сервері застосунку. Це забезпечує повний аудит усіх переходів між статусами та можливість відновлення хронології подій у разі виникнення спірних ситуацій.

Ключовою особливістю модуля є інтеграція з державним сервісом електронного підпису «Дія». Клас SignController мікросервісу Node.js відповідає за повний цикл підписання. Отримання сесійного токена здійснюється через автентифікацію за токеном набувача:

```
const response = await axios.get(
  `${this.diiiaHost}/api/v1/auth/acquirer/${this.acquirerToken}`,
  {
    headers: {
      Authorization: `Basic ${this.authAcquirerToken}`,
      Accept: "application/json",
    },
  }
);
this.sessionToken = response.data.token;
this.sessionTokenTimestamp = Date.now();
```

Перед кожним зверненням до API метод ensureSessionToken перевіряє актуальність токена та оновлює його у разі закінчення терміну дії:

```
async ensureSessionToken() {
  if (!this.sessionToken ||
    Date.now() - this.sessionTokenTimestamp > this.tokenExpiryTime) {
    await this.getSessionToken();
  }
}
```

Токен кешується разом із часом отримання та автоматично оновлюється після закінчення двох годин. Такий підхід мінімізує кількість звернень до API автентифікації, зберігаючи при цьому актуальність сесії. Для ініціювання підписання документа спочатку для кожного файлу обчислюється SHA-256 хеш у форматі Base64:

```
hashFile(fileBuffer) {
  const hash = crypto.createHash('sha256');
  hash.update(fileBuffer);
```

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				44
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    return hash.digest('base64');
}

```

Хешування файлу замість передачі його вмісту безпосередньо до API «Дії» є вимогою протоколу підписання - сервіс «Дія» отримує хеш документа, підписує його у застосунку та повертає підписану версію. Це забезпечує конфіденційність вмісту документа, оскільки сам файл не передається до зовнішнього сервісу. Після обчислення хешів формується запит на генерацію deeplink:

```

const payload = {
  offerId,
  requestId,
  signAlgo: "ECDSA",
};

const response = await axios.post(
  `${this.diiHost}/api/v2/acquirers/branch/${branchId}/offer-request/dynamic`,
  payload,
  { headers: { Authorization: `Bearer ${this.sessionToken}` } }
);

```

У відповідь API повертає deeplink, з URL якого витягується OTP-код через `url.pathname.split('/').pop()` - він використовується для подальшої перевірки статусу підписання. Після того як користувач підписує документ у застосунку «Дія», підписані файли надходять на вебхук мікросервісу. Тут реалізовано розрізнення двох режимів роботи залежно від наявності запису в таблиці `request_ids`. У legacy-режимі підписаний файл зберігається у директорії конкретної кваліфікаційної роботи:

```

const savedFilePath = path.join(
  BASE_UPLOAD_DIRECTORY,
  String(requestData.qualification_work_id),
  savedFileName
);

await fs.writeFile(savedFilePath, signatureBuffer);

await updateQualificationWorkFilePath(
  requestData.qualification_work_id,
  fieldName,
  savedFilePath
);

```

		Коновал М.М.			ІІЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Вакалюк Т.А.				45
Змн.	Арк.	№ докум.	Підпис	Дата		

);

У universal-режимі файли зберігаються у тимчасовій директорії та фіксуються у таблиці підписаних файлів:

```
const savedFilePath = path.join(  
  BASE_UPLOAD_DIRECTORY_UNIVERSAL,  
  String(requestId),  
  savedFileName  
);  
await fs.writeFile(savedFilePath, signatureBuffer);  
await processUniversalSignedFiles(requestId, savedFilePaths);
```

Такий підхід забезпечує гнучкість системи для майбутнього розширення без зламу існуючої логіки. Отримані підписи перед збереженням підвищуються до формату CAdES-X-Long через бібліотеку ІТ:

```
const cadesSigned = await EU.makeCAdESSign(  
  rawSignature,  
  null,  
  EU.EU_SIGN_TYPE_CADES_X_LONG  
);
```

Це юридично значущий крок - формат CAdES-X-Long містить вбудовані відомості про сертифікат та мітку часу, що гарантує можливість верифікації підпису навіть після закінчення строку дії сертифіката підписанта. Результат зберігається на диску з розширенням .p7s, а шлях до файлу записується у відповідне поле таблиці `qualification_work`.

Для перевірки статусу підписання мікросервіс одночасно звертається до API «Дії» та читає підписані файли з бази даних:

```
const filePaths = await getSignedFilesFromDB(requestId);  
for (const filePath of filePaths) {  
  const fileBuffer = await fs.readFile(filePath);  
  signedFilesList.push({  
    fileName: path.basename(filePath),  
    data: `data:application/pkcs7-signature;base64,${fileBuffer.toString('base64')}`  
  });  
}  
return { ...response.data, signed_files: signedFilesList };
```

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				46
Змн.	Арк.	№ докум.	Підпис	Дата		

Це дозволяє фронтенду в одному запиті отримати і статус підписання, і самі підписані файли у форматі Data URI для подальшого відображення або завантаження користувачем без додаткових звернень до сервера.

Для здобувачів, що підписують роботу безпосередньо на порталі без мобільного застосунку, реалізовано механізм підписання через криптографічну бібліотеку ІТ. Функція onSign отримує вміст файлу у форматі Uint8Array та викликає метод підписання з алгоритмом DSTU4145WithGOST34311 та типом підпису CAdES_X_Long:

```
euSign.SignData(data, external, asBase64String, signAlgo, null, signType)
.then(function (sign) {
    const signFormData = new FormData();
    signFormData.append('file', new Blob([sign]));
    signFormData.append('qualificationWorkId', fileData.qualificationWorkId);
    signFormData.append('fileName', fileData.fileName);
    $.ajax({
        url: '/student/qualification-work/upload-sign-file',
        type: 'POST',
        data: signFormData,
        processData: false,
        contentType: false,
        success: function () { successCallback(); }
    });
});
```

Підписання кожного файлу та надсилання підпису виконуються послідовно в циклі по масиву loadedFilesData - це запобігає race condition при одночасному підписанні кількох документів, коли паралельні асинхронні запити могли б перезаписати один одного.

На стороні здобувача перевірка належності кваліфікаційної роботи реалізована через метод userCheck, що зіставляє ідентифікатор освіти з усіма освітніми записами особи:

```
$studentEducationIds = EdeboStudent::find()
->select('education_id')
->where(['person_id' => $userPersonId])
->column();
```

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		47

```
return in_array($qualificationWork->education_id, $studentEducationIds);
```

Отримання всіх освітніх записів особи, а не лише поточного, є принциповим - один здобувач може мати кілька паралельних освітніх записів при навчанні за кількома спеціальностями одночасно. Цей метод викликається на початку кожної дії, що стосується конкретної роботи, забезпечуючи надійний захист від несанкціонованого доступу до чужих документів. Відкликання роботи здобувачем рекурсивно видаляє всю директорію файлів, скидає всі файлові поля до null та повертає статус до початкового значення з відповідним записом в історії. Транзакція тут свідомо не використовується - видалення файлів із файлової системи є незворотною операцією і не може бути відкочено разом із базою даних, тому спочатку виконуються операції з базою даних і лише після їх успіху - видалення файлів.

Висновки до другого розділу

У процесі проектування та розробки модулів «Кваліфікаційні роботи» та «Опитування здобувачів освіти» було визначено функціональні та нефункціональні вимоги, сформульовано загальні цілі розробки, а також спроектовано архітектуру взаємодії основного застосунку з мікросервісом інтеграції державного сервісу електронного підпису «Дія». Отримані рішення дозволяють цифровізувати повний цикл управління кваліфікаційними роботами - від первинного завантаження до підписання документів та фіксації результатів захисту - а також забезпечують систематизований збір та аналіз зворотного зв'язку від здобувачів освіти.

У ході аналізу предметної області було створено формалізовану інформаційну модель, що охоплює всі релевантні сутності цифрового освітнього середовища. На підставі цієї моделі побудовано діаграму класів, яка відображає логічні зв'язки між об'єктами системи. Архітектурні рішення розроблялись з урахуванням вимог технічного завдання, що забезпечило відповідність проекту критеріям масштабованості, стійкості до збоїв, безпечної обробки даних та зручності подальшого супроводу.

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				48
Змн.	Арк.	№ докум.	Підпис	Дата		

Результати реалізації підтверджують відповідність програмного продукту поставленим технічним і функціональним вимогам. Зокрема, реалізовано повний цикл управління статусами кваліфікаційних робіт із рольовою моделлю доступу, механізм юридично значущого електронного підписання документів через інтеграцію з API сервісу «Дія» з підвищенням рівня підпису до формату CAdES-X-Long, інструменти створення та проведення опитувань із гнучким налаштуванням цільової аудиторії, а також автоматизоване формування аналітичних звітів за результатами опитувань у форматі Excel. У підсумку досягнуто основну мету проєкту - створено ефективний засіб цифрової підтримки освітнього процесу, що відповідає актуальним вимогам до сучасних університетських інформаційних систем.

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		49

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

3.1. Порядок встановлення та налаштування параметрів додатку

До початку розгортання системи на виробничих серверах було побудовано діаграму розгортання, яка відображає фізичну та логічну організацію компонентів програмного комплексу (рис. 3.1):

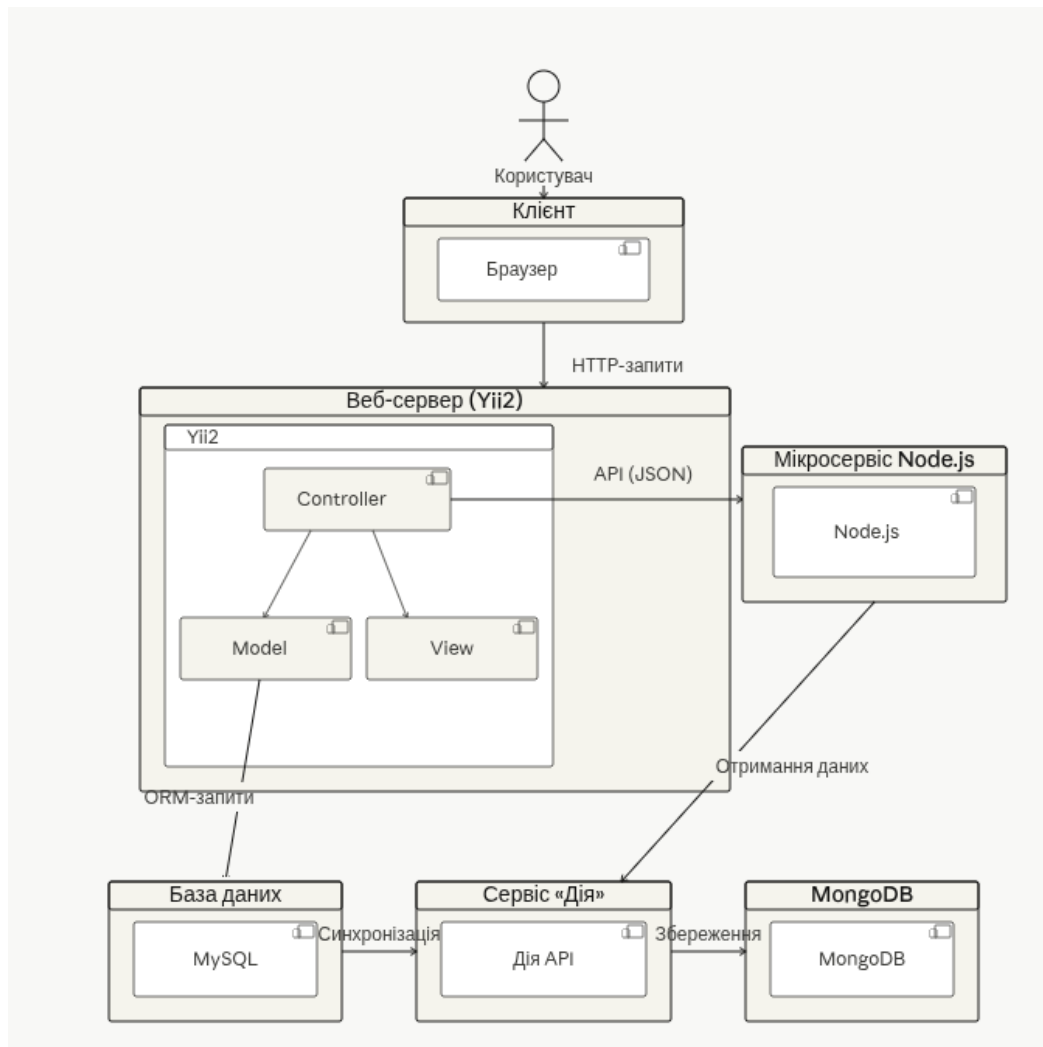


Рисунок 3.1 - Діаграма розгортання додатку

Побудована діаграма ілюструє розподіл відповідальності між вузлами інформаційної системи Digital University UA. В основі архітектури лежить мікросервісний підхід: Yii2-застосунок із серверним рендерингом виступає ядром системи, тоді як взаємодія з державним сервісом електронного підпису «Дія» винесена в окремий мікросервіс на платформі Node.js. Такий розподіл

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		50

дозволяє розвивати кожен компонент незалежно та мінімізує ризики при оновленні інтеграційного шару.

Кінцевий користувач отримує доступ до системи через стандартний веббраузер без необхідності встановлення додаткового програмного забезпечення. Вхідні запити обробляються Yii2-застосунком відповідно до патерну MVC, де кожен рівень має чітко визначену зону відповідальності. Контролери виступають головною точкою входу для HTTP-запитів - вони координують прийом даних від користувача, забезпечують первинну валідацію, викликають відповідні сервіси бізнес-логіки та передають результати на рівень відображення. Серед ключових контролерів системи варто виділити QualificationWorkViewController, який відповідає за координацію процесів перегляду та зміни статусів кваліфікаційних робіт, PollController, що регулює доступ користувачів до активних опитувань, та PollResultsController, який забезпечує агрегацію відповідей і формування аналітичних звітів для адміністрації. Моделі є центральним шаром бізнес-логіки застосунку - вони безпосередньо взаємодіють із реляційною базою даних через механізм Active Record ORM, відповідають за декларативну валідацію вхідних даних перед їх збереженням та опис зв'язків між сутностями предметної області. Рівень відображення формує кінцеву HTML-відповідь на основі даних, підготовлених контролером, із використанням вбудованих механізмів захисту від XSS-атак шляхом автоматичного екранування символів.

Реляційна база даних MySQL зберігає всю структуровану інформацію системи - детальні дані про здобувачів освіти, їхні профілі та академічні групи, кваліфікаційні роботи разом із поточними статусами, підписані електронні документи та шляхи до відповідних файлів у сховищі. Окремо зберігається вся логіка модуля опитувань, включаючи переліки питань, варіанти відповідей, зафіксовані результати та записи про участь здобувачів для запобігання повторному проходженню. Для забезпечення високої швидкодії при обробці великих масивів інформації впроваджено індексування ключових полів, що

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		51

використовуються у фільтрах та операціях об'єднання таблиць, а всі критичні операції виконуються в межах строго контрольованих ACID-транзакцій.

Мікросервіс Node.js забезпечує повний цикл взаємодії з API державного сервісу «Дія». Вибір цього технологічного стеку зумовлений асинхронною природою платформи, що дозволяє ефективно обробляти велику кількість паралельних мережових запитів без блокування основного потоку виконання. Мікросервіс автоматично авторизується в системі державного сервісу та проактивно поновлює сесійний токен доступу після закінчення терміну його дії, що гарантує безперебійну роботу для кінцевих користувачів. Для ініціювання підписання мікросервіс формує та криптографічно захищає запити на генерацію унікальних deeplink-посилань і QR-кодів для підписання конкретних документів у мобільному застосунку користувача. Після завершення підписання мікросервіс приймає асинхронні вебхуки з підписаними файлами від серверів «Дія», перевіряє їхню автентичність та обробляє через інтегровану сертифіковану бібліотеку ІТ - на цьому етапі базовий цифровий підпис підвищується до формату довгострокового зберігання CAdES-X-Long, що забезпечує юридичну силу документів навіть після завершення терміну дії сертифікатів користувачів.

Обмін даними між основним застосунком і мікросервісом відбувається через HTTP-запити у форматі JSON - технологічно нейтральний підхід, що не прив'язує компоненти до конкретного стеку і дозволяє в майбутньому замінити або масштабувати будь-який із сервісів без зміни суміжних компонентів. Особливо це актуально для мікросервісу «Дія» - у разі змін у специфікації державного API оновлення торкнеться лише одного ізольованого компонента, не порушуючи стабільність ядра системи. Для запобігання втраті даних у разі тимчасових мережових збоїв взаємодія підтримується механізмами автоматичного повторення запитів та контролю ідемпотентності.

Система сумісна з усіма актуальними браузерами - Google Chrome (v45+), Mozilla Firefox (v38+), Microsoft Edge та Safari. Завдяки використанню серверного рендерингу на базі Yii2 та оптимізованих клієнтських скриптів однаковий рівень функціональності та коректне відображення інтерфейсу

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				52
Змн.	Арк.	№ докум.	Підпис	Дата		

забезпечуються незалежно від типу пристрою, роздільної здатності екрана чи операційної системи користувача.

3.2. Структура інтерфейсу додатку

Розроблена інформаційна система Digital University UA вирізняється зручним користувацьким інтерфейсом, що забезпечує однаковий рівень функціональності як на десктопних, так і на мобільних пристроях завдяки повній адаптивності верстки.

Доступ до системи розмежований відповідно до ролей учасників освітнього процесу. Перед початком роботи з модулями «Кваліфікаційні роботи» та «Опитування здобувачів освіти» користувач проходить автентифікацію шляхом введення облікових даних у форму входу (рис. 3.2). Це гарантує, що кожен учасник - здобувач освіти, науковий керівник, завідувач кафедри, декан або адміністратор - отримує доступ виключно до того функціоналу, який відповідає його ролі в системі.

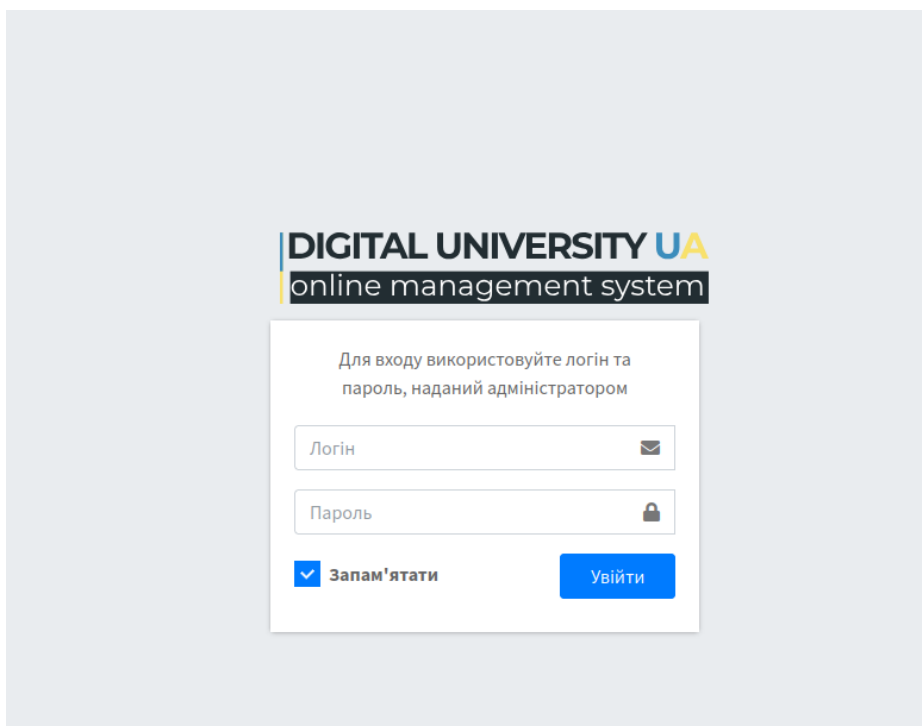


Рисунок 3.2 - Сторінка авторизації

З міркувань інформаційної безпеки самостійна реєстрація користувачів через вебінтерфейс у системі не передбачена. Створення облікових записів здійснюється централізовано відповідальним підрозділом університету, який

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		53

формує та розповсюджує облікові дані для доступу. Такий підхід унеможливорює потрапляння до системи осіб, що не мають відношення до університету, та забезпечує повний контроль над складом користувачів на всіх рівнях - від здобувачів освіти до адміністраторів.

Після успішної автентифікації користувач потрапляє на головну сторінку платформи. Інтерфейс спроектований таким чином, щоб мінімізувати кількість кроків до потрібного функціоналу. Модулі «Кваліфікаційні роботи» та «Опитування здобувачів освіти» разом з іншими компонентами системи логічно згруповані та відображені у навігаційній панелі в лівій частині екрана (рис. 3.3), що дозволяє швидко переміщатися між розділами незалежно від поточного місцезнаходження в системі. (рис. 3.3).

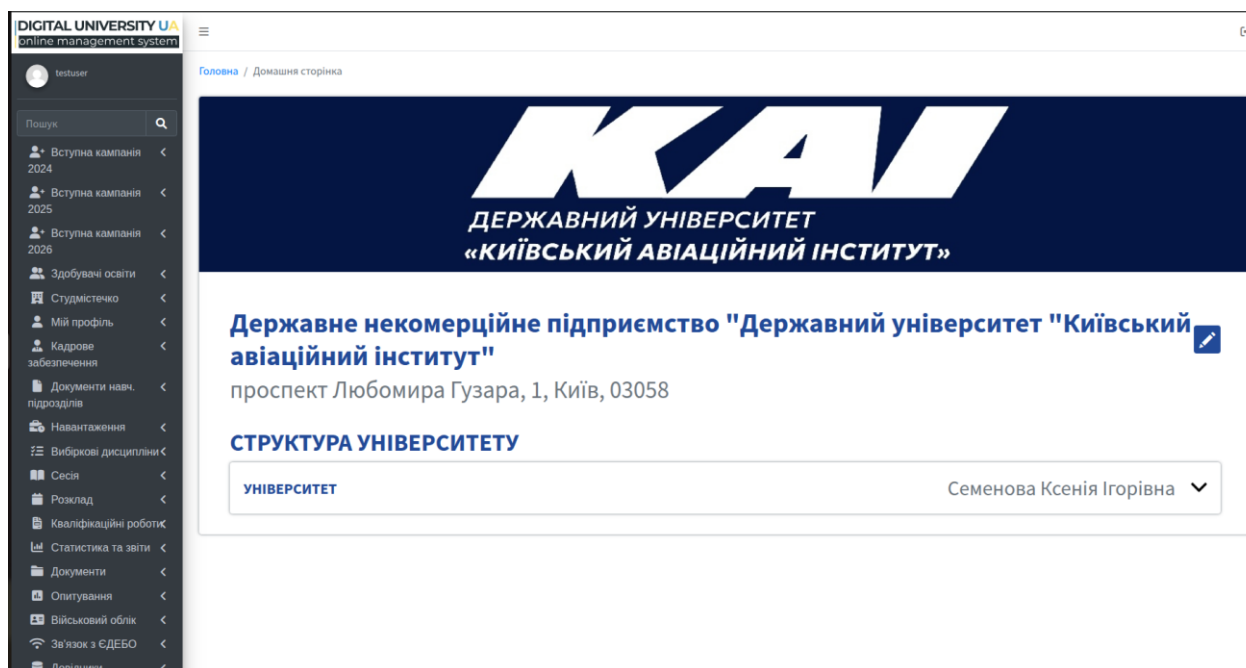


Рисунок 3.3 - Головна сторінка

Серед усіх частин спроектованого додатку розглянемо модулі «Кваліфікаційні роботи» та «Опитування здобувачів освіти», які є ключовими компонентами системи Digital University UA в межах даної кваліфікаційної роботи. Модуль «Кваліфікаційні роботи» забезпечує повний цикл управління процесом підготовки та захисту кваліфікаційних робіт здобувачів, включно з юридично значущим електронним підписанням документів через сервіс «Дія».

Модуль «Опитування здобувачів освіти» надає інструменти для

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		54

організації, проведення та аналізу опитувань із метою отримання систематизованого зворотного зв'язку щодо якості освітнього процесу. Доступ до модуля «Кваліфікаційні роботи» здійснюється через відповідний розділ навігаційної панелі порталу (рис. 3.4):

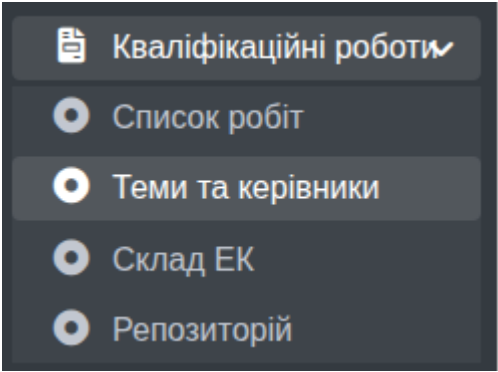


Рисунок 3.4 – Пункт меню «Кваліфікаційні роботи»

Огляд модуля «Кваліфікаційні роботи» здійснюється від імені користувача з правами адміністратора, оскільки повний доступ до реєстру робіт, управління статусами та налаштування інтеграції з сервісом «Дія» доступні відповідно до рольової моделі системи. При переході до розділу «Кваліфікаційні роботи» користувачу відображається сторінка реєстру із переліком усіх кваліфікаційних робіт здобувачів (рис. 3.5):

Головна / Кваліфікаційні роботи

Експортувати всі відфільтровані

Кваліфікаційні роботи

Показані 1-50 із 3 831 запису.

Рік випуску

№	ПІБ здобувача	Експортовано в репозитарій	Публічний доступ	Група	Статус роботи	Тема роботи
1	Абакумов Артем Олегович	Ні	Публічно	Б-186-21-1-БП	Завантаження	Модель навчального ресурсу «Методи конвертації зі стисненням і редагуванням цифрових ф
2	Абашіна Катерина Олександрівна	Ні	Публічно	Б-161-21-1-ХП	Завершення навчання	Покращення антикорозійних властивостей дизельних палив додаванням при
3	Аббасова Марина Сеймурівна	Ні	Публічно	Б-192-21-1-ЦБ	Завершення навчання	Склад міжнародних відправлень аеропорту "Бориспіль" Київської област
4	Абдрахманов Ренат Маратович	Ні	Публічно	Б-121-21-2-ПІ	Завершення навчання	Засіб розробки інтернет-магазину з адаптивним інтерфейсом на основі React т
5	Абласва Єлизавета Сергіївна	Так	Публічно	Ба-275-21-2-ОП	Завершення навчання	Організація та технологія транспортно-експедиційного обслуговування
6	Абраменко Анастасія Андріївна	Так	Публічно	М-163-24-1-БМ	Завершення навчання	Остеоінтеграційне протезування нижніх кінцівок
7	Абушкур Імад Еддін	Ні	Публічно	Б-191-21-1-АР	Завершення навчання	Навчальний заклад у м. Касабланка, Марокко
8	Авласова Катерина Вадимівна	Ні	Публічно	Ба-293-21-1-МП	Завершення навчання	The Implementation and Effectiveness of EU Sanctions Against Russia
9	Авраменко Ангеліна Сергіївна	Ні	Публічно	Б-126-21-1-ІТ	Завершення навчання	Мобільний додаток для екстреного оповіщення та виклику служб допомоги
10	Авраменко Андрій Юрійович	Ні	Публічно	Бст-076-22-1-БТ	Завершення навчання	Формування моделі розвитку підприємницької діяльності за сучасних ум
11	Авраменко Вікторія Сергіївна	Так	Публічно	Б-275-21-1-БС	Завершення навчання	Удосконалення технологічних операцій забезпечення сервісу як чинника підвищення конкурентоспромож
12	Авраменко Марина Вікторівна	Ні	Публічно	Б-073-21-1-МА	Завершення навчання	Розроблення операційної стратегії для забезпечення конкурентоспроможності ТОВ «
13	Авраменко Олександра Юріївна	Ні	Публічно	Б-101-21-1-ЕК	Завершення навчання	Оцінка екологічного стану річки Десна
14	Аврамчук Лада Ігорівна	Ні	Публічно	Мз-072-24-1-ФК	Завершення навчання	Управління ресурсами банківської установи та шляхи його удосконалення
15	Агалов Ілля Максимович	Ні	Публічно	Б-075-21-2-МР	Завершення навчання	Дослідження шляхів підвищення конкурентних переваг підприємства (на прикладі ТОВ «
16	Агеєв Денис Олексійович	Так	Публічно	Б-073-21-1-АЛ	Завершення навчання	Планування міжнародних авіаперевезень в умовах глобальних криз
17	Адамов Олександр Олексійович	Ні	Публічно	Б-121-21-3-ПІ	Завершення навчання	Програма компонента виявлення кібератак на комп'ютерні системи за принципами гли

Рисунок 3.5 – Сторінка реєстру кваліфікаційних робіт

На сторінці реєстру кваліфікаційних робіт відображається перелік усіх робіт здобувачів університету з основною інформацією: ПІБ здобувача, академічна група, тема роботи, поточний статус, статус експортування до репозитарію та публічний доступ. Для кожного запису доступні кнопки перегляду, редагування та видалення. Передбачено фільтрацію за роком випуску, групою та статусом роботи.

Нижче зображена сторінка призначення тем, керівників та складу екзаменаційної комісії кваліфікаційних робіт. (рис. 3.6):

Рисунок 3.6 – Сторінка призначення тем та керівників КВ

При переході до картки конкретної кваліфікаційної роботи відкривається детальна сторінка з кількома вкладками: «Статус роботи», «Загальна інформація», «Повідомлення», «Результати навчання» та «Історія змін». Така структура дозволяє логічно розмежувати різні аспекти роботи з кваліфікаційною роботою та забезпечує зручну навігацію для всіх учасників процесу.

Вкладка «Статус роботи» відображає поточний етап проходження кваліфікаційної роботи у вигляді наочної покрокової схеми. Кожен етап представлений окремим блоком: «Завантаження» → «Перевірка керівником» → «Нормоконтроль» → «Антиплагіат» → «Погодження завідувача кафедри» →

«Підготовка відгуку і рецензії» → «Отримання допуску до захисту» → «Допущено до захисту» → «Роботу захищено» → «Завершення навчання». Усі пройдені етапи підсвічуються зеленим кольором, що дозволяє миттєво оцінити стан роботи, а поточний активний етап виділяється окремо, надаючи учаснику чітке розуміння того, яка дія від нього очікується на даному кроці (рис. 3.7):

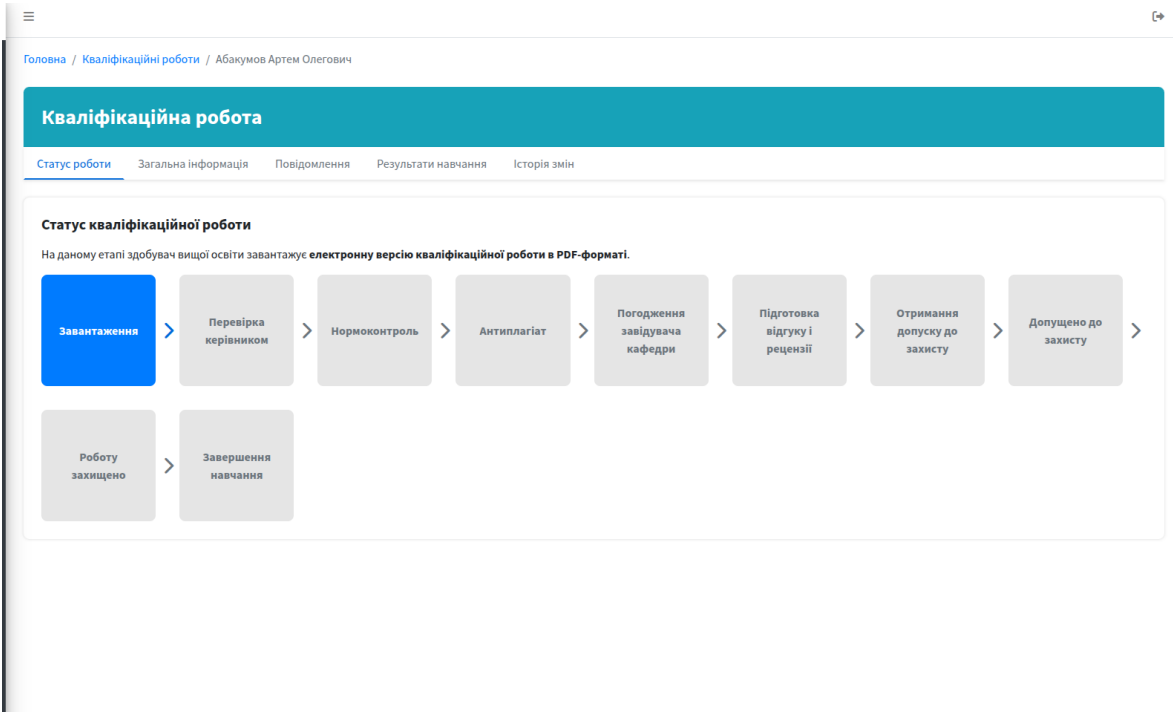


Рисунок 3.7 - Перегляд статусу кваліфікаційної роботи

Вкладка «Історія змін» є невіддільним доповненням до схеми статусів - вона фіксує всі переходи між етапами із зазначенням точної дати та часу кожної події, а також ідентифікатора користувача, який ініціював зміну. Це забезпечує повний аудит процесу підготовки роботи та дозволяє відновити хронологію подій у разі виникнення спірних ситуацій між учасниками процесу (рис. 3.8):

Кваліфікаційна робота		
Статус роботи	Загальна інформація	Повідомлення
Результати навчання	Історія змін	
Історія змін		
№	Назва статусу	Дата
1	Завершення навчання	17.06.2025 19:54
2	Роботу захищено	17.06.2025 19:54
3	Допущено до захисту	17.06.2025 11:03
4	Отримання допуску до захисту	17.06.2025 10:06
5	Підготовка відгуку і рецензії	16.06.2025 23:00
6	Погодження завідувача кафедри	16.06.2025 22:59
7	Антиплагіат	16.06.2025 17:19
8	Нормоконтроль	16.06.2025 15:45
9	Перевірка керівником	16.06.2025 15:34

Рисунок 3.8 - Історія змін статусів кваліфікаційної роботи

Вкладка «Загальна інформація» містить повний перелік відомостей про роботу - назву, ПІБ наукового керівника, консультанта та рецензента, а також усі прикріплені документи з можливістю їх завантаження. Для кожного типу документа передбачені окремі кнопки доступу - PDF-файл та підпис у форматі P7S. Зокрема, для основної кваліфікаційної роботи доступні: PDF, P7S, підпис керівника, підпис завідувача кафедри та підпис нормоконтролера. Окремо відображаються довідки про текстові збіги, відгук та рецензія. Наявність підписаних файлів у форматі P7S поруч із кожним документом дозволяє будь-якому учаснику процесу в будь-який момент перевірити автентичність та цілісність підписаного документа (рис. 3.9):

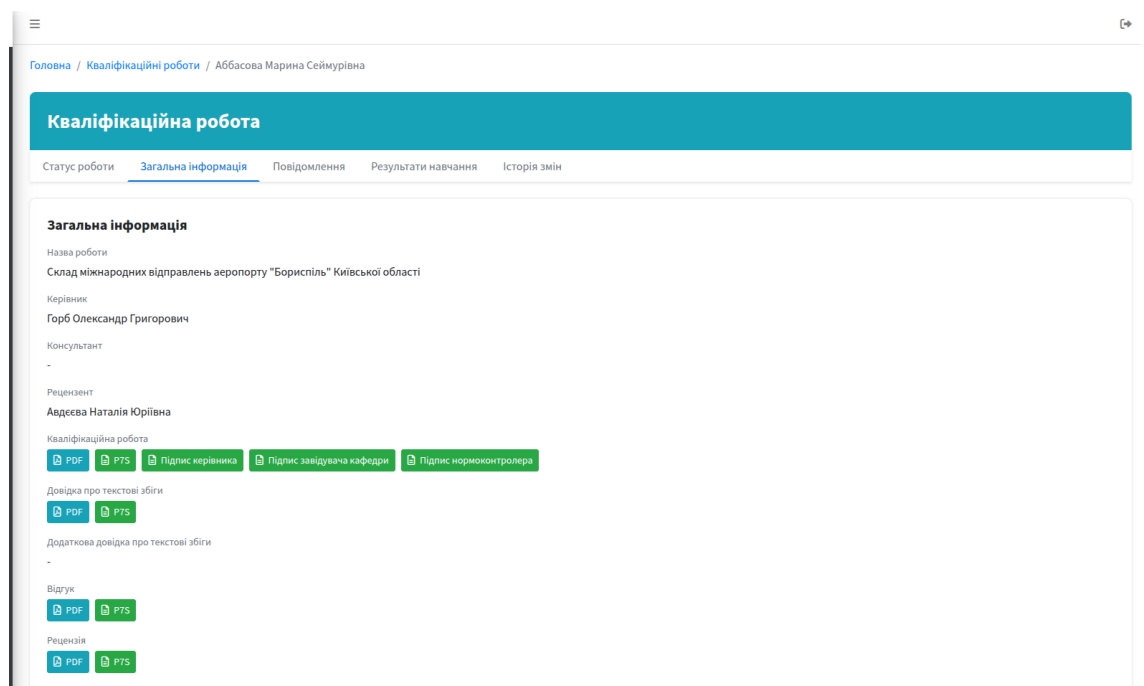


Рисунок 3.9 - Перегляд інформації і файлів кваліфікаційної роботи

Вкладка «Повідомлення» забезпечує комунікацію між здобувачем та усіма задіяними у процесі особами - науковим керівником, нормоконтролером, завідувачем кафедри та іншими учасниками. Наявність вбудованого чату усуває потребу у зовнішніх засобах комунікації та зберігає всю переписку в контексті конкретної роботи. Вкладка «Результати навчання» підтягує академічну успішність здобувача з бази даних системи, надаючи членам екзаменаційної

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		58

комісії повну картину успішності здобувача без необхідності звернення до інших підсистем.

Окремої уваги заслуговує блок виконання операцій над роботою, що відображається на сторінці залежно від поточного статусу та ролі користувача. Блок містить дві дії - «Погодження роботи» та «Повернення на доопрацювання». При виборі погодження користувачу пропонується підписати документ одним із двох способів: за допомогою КЕП або Дія.Підпис (рис. 3.10).

Виконання операцій над роботою

✓ Погодження роботи ✗ Повернення на доопрацювання

Підпишіть роботу за допомогою:

КЕП Дія.Підпис

Рисунок 3.10 – Виконання операцій над КВ

При виборі підписання через КЕП відкривається модальне вікно з двокроковим процесом: на першому кроці користувач завантажує PDF-файл роботи, на другому - зчитує особистий ключ електронного підпису через файловий носій, токен або хмарний сервіс кваліфікованого надавача електронних довірчих послуг (рис. 3.11).

В обох випадках QR-код та посилання мають обмежений термін дії, що відображається у вигляді зворотного відліку часу. Після закінчення терміну дії користувач може згенерувати новий запит на підписання:

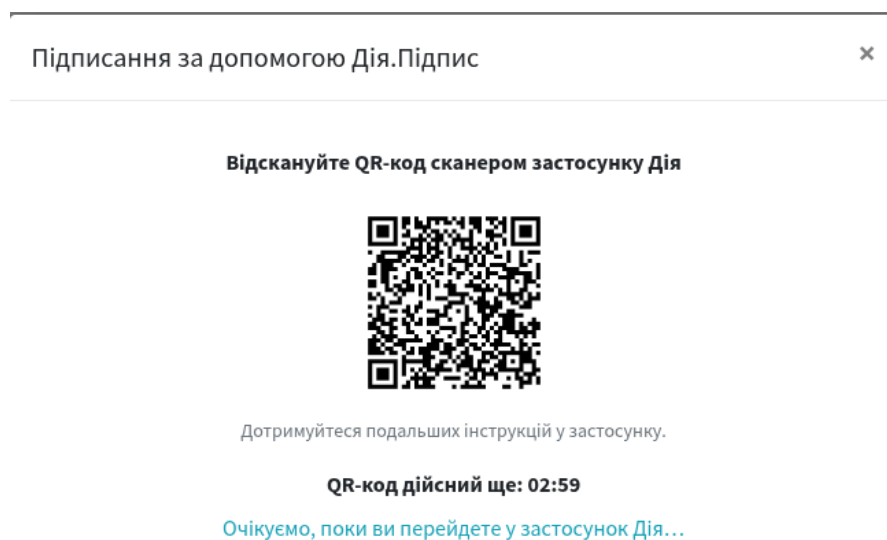


Рисунок 3.12 – Блок підписання КВ за допомогою ДІА підпис

Окрім порталу адміністрації, модуль «Кваліфікаційні роботи» реалізований також в особистому кабінеті здобувача освіти. Після авторизації здобувач потрапляє до свого кабінету з навігаційною панеллю у лівій частині екрана, де відображається його ім'я, академічна група та перелік доступних розділів - «Особиста інформація», «Сесія», «Розклад», «Розклад сесії», «Обхідний лист», «Кваліфікаційна робота», «Вибіркові дисципліни», «Опитування» та «Фінансові документи» (рис. 3.13).

При переході до розділу «Кваліфікаційна робота» здобувач бачить поточний статус своєї роботи у вигляді тієї самої покрокової схеми, що і на порталі адміністрації. Поточний активний етап підсвічується помаранчевим кольором, решта - сірим, що чітко вказує на те, на якому саме кроці перебуває робота. На початковому етапі «Завантаження» здобувачу відображається підказка про необхідну дію - завантажити останню версію кваліфікаційної роботи у PDF-форматі. Безпосередньо під схемою статусів розміщено блок завантаження файлу з полем вибору PDF-документа та кнопкою «Далі» для переходу до наступного кроку - підписання роботи. Таким чином, інтерфейс

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		61

кабінету здобувача є максимально зрозумілим і послідовно веде користувача через усі необхідні дії на кожному етапі процесу підготовки до захисту:

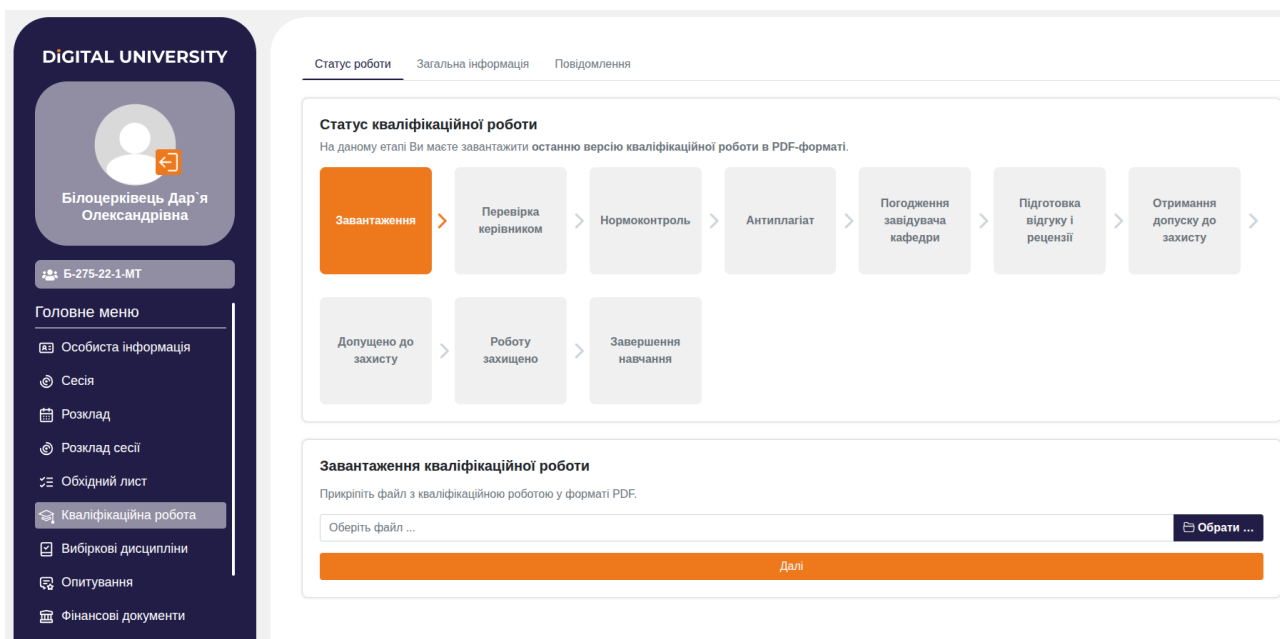


Рисунок 3.13 – Інтерфейс кабінету студента (модуль КР)

Модуль «Опитування здобувачів освіти» розміщений в окремому розділі навігаційної панелі порталу та містить чотири підрозділи: «Опитування», «Форми», «Результати» та «Варіанти відповідей» (рис. 3.14).

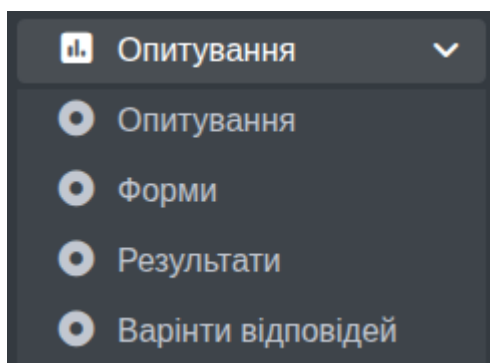


Рисунок 3.14 – Пункт навігаційної панелі «Опитування»

При переході до підрозділу «Опитування» відображається реєстр усіх створених опитувань із основними характеристиками кожного: назва, статус, тип опитування, освітній ступінь та факультет (рис. 3.15). Система підтримує два типи опитувань - «Загальне» та «За освітніми компонентами (За ОК)». Загальне опитування призначається для збору зворотного зв'язку щодо освітньої програми в цілому, тоді як опитування типу «За ОК» дозволяє оцінювати якість викладання кожної навчальної дисципліни окремо. Гнучка система налаштування цільової

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		62

аудиторії дозволяє призначати опитування для конкретного факультету, підрозділу, освітньої програми, академічної групи або навіть окремого здобувача. Усі опитування є анонімними - система фіксує лише сам факт проходження без прив'язки відповідей до особи здобувача. Для кожного опитування доступні дії перегляду, редагування, копіювання та видалення.

Опитування +						
Показані 1-20 із 192 записів.						
#	Дії	Статус	Назва опитування	Тип опитування	Освітній ступінь	Факультет
		Усі ▼ ✕		Усі ▼ ✕	▼ ✕	
1	  		Якість реалізації освітньої програми "Правознавство"	Загальне	Бакалавр	Факультет права та міжнародних відносин
2	  		Оцінювання якості викладання дисциплін ОПП "Прикладна механіка композиційних конструкцій та технічних систем"	За ОК	Бакалавр	Аерокосмічний факультет
3	  		Оцінювання якості викладання дисциплін ОПП "Комп'ютерні системи та мережі"	За ОК	Бакалавр	Факультет комп'ютерних наук та техніки
4	  		Оцінювання якості викладання дисциплін ОПП "Дизайн архітектурного середовища"	За ОК	Бакалавр	Факультет архітектури, будівництва та інфраструктури
5	  		Оцінювання якості викладання дисциплін ОПП "Політологія"	За ОК	Бакалавр	Гуманітарно-мистецький факультет
6	  		Оцінювання якості викладання дисциплін ОПП "Технології обробки великих даних"	За ОК	Магістр	Факультет комп'ютерних наук та техніки
7	  		Оцінювання якості викладання дисциплін ОПП "Технології обробки великих даних"	За ОК	Магістр	Факультет комп'ютерних наук та техніки
8	  		Оцінювання якості викладання дисциплін ОПП "Системи та технології кібербезпеки"	За ОК	Магістр	Факультет комп'ютерних наук та техніки
9	  		Оцінювання якості викладання дисциплін ОПП "Системи та технології кібербезпеки"	За ОК	Магістр	Факультет комп'ютерних наук та техніки
10	  		Оцінювання якості викладання дисциплін ОПП "Системи технічного захисту інформації, автоматизація її обробки"	За ОК	Магістр	Факультет комп'ютерних наук та техніки
11	  		Оцінювання якості викладання дисциплін ОПП "Системи технічного захисту інформації, автоматизація її обробки"	За ОК	Магістр	Факультет комп'ютерних наук та техніки
12	  		Оцінювання якості викладання дисциплін ОПП "Електронні технології Інтернету речей"	За ОК	Магістр	Факультет аеронавігації, електроніки та телекомунікацій
13	  		Оцінювання якості викладання дисциплін ОПП "Електронні технології Інтернету речей"	За ОК	Магістр	Факультет аеронавігації, електроніки та телекомунікацій
14	  		Оцінювання якості викладання дисциплін ОПП "Обслуговування повітряного руху"	За ОК	Магістр	Факультет аеронавігації, електроніки та телекомунікацій
15	  		Оцінювання якості викладання дисциплін ОПП "Обслуговування повітряного руху"	За ОК	Магістр	Факультет аеронавігації, електроніки та телекомунікацій
16	  		Оцінювання якості викладання дисциплін ОПП "Системи аеронавігаційного обслуговування"	За ОК	Магістр	Факультет аеронавігації, електроніки та телекомунікацій
17	  		Оцінювання якості викладання дисциплін ОПП "Системи аеронавігаційного обслуговування"	За ОК	Магістр	Факультет аеронавігації, електроніки та телекомунікацій
18	  		Оцінювання якості викладання дисциплін ОПП "Публічне управління та врегулювання"	За ОК	Магістр	Факультет менеджменту, фінансів та маркетингу
19	  		Оцінювання якості викладання дисциплін ОПП "Публічне управління та врегулювання"	За ОК	Магістр	Факультет менеджменту, фінансів та маркетингу
20	  		Оцінювання якості викладання дисциплін ОПП "Германські мови та літератури (переклад включно), перша - англійська"	За ОК	Магістр	Гуманітарно-мистецький факультет

Рисунок 3.15 – Сторінка «Опитування»

Попередній перегляд опитування дозволяє адміністратору побачити форму саме так, як її бачитиме здобувач (рис. 3.16). Питання згруповані за спільними варіантами відповідей - групи питань з однаковим набором відповідей відображаються в одній таблиці, що робить форму компактною та зручною для заповнення. Варіанти відповідей представлені у вигляді горизонтальної шкали від позитивної до негативної оцінки.

Оцінювання якості викладання дисциплін ОПП "Прикладна механіка композиційних конструкцій та технічних систем"

Матеріалознавство					
Питання	так	скоріше так	частково	ні	не маю відповіді
	5	4	3	2	1
Актуальність матеріалу	5	4	3	2	1
Зв'язок теорії з практикою і сучасними науковими дослідженнями	5	4	3	2	1
Критерії оцінювання результатів навчання та порядок накопичення балів є чіткими, прозорими і зрозумілими	5	4	3	2	1
Результати навчання оцінюються об'єктивно і неупереджено	5	4	3	2	1
Використання інноваційних методів навчання (проекти, кейс-стаді, тощо)	5	4	3	2	1
Використання інтерактивних методів (мозкові штурми, групова робота, робота в парах, тощо)	5	4	3	2	1
Зміст і обсяг самостійної роботи є оптимальним	5	4	3	2	1
Заняття проводяться за розкладом	5	4	3	2	1
Питання	так	іноді	частково	ні	не маю відповіді
	5	4	3	2	1
Викладач надає консультації	5	4	3	2	1
Питання	так	скоріше так	частково	ні	не маю відповіді
	5	4	3	2	1

Рисунок 3.16 – Попередній перегляд опитування

Підрозділ «Форми опитування» містить перелік шаблонів, на основі яких створюються опитування (рис. 3.17). Кожна форма має назву, статус активності, кількість питань та опис із інструкцією для здобувачів. Адміністратор може створювати нові форми, редагувати наявні та переглядати їхній вміст. Один шаблон форми може використовуватися для кількох різних опитувань, що дозволяє уникнути дублювання при організації опитувань для різних груп чи факультетів.

Головна / Опитування / Форми

Форми опитування

+ Створити форму

Пошук за назвою...

Пошук

Якість реалізації освітньо-професійної програми (магістр_2 курс)

Активна

Кількість питань: 25

Просимо Вас дати відповіді на запитання з метою покращення освітнього процесу в KAI.

Оцінювання якості викладання дисципліни

Активна

Кількість питань: 13

Оцініть, будь ласка, якість викладання дисципліни за критеріями по 5-ти бальній шкалі, обравши один варіант відповіді (5 – так; 4 – скоріше так; 3 – частково; 2 – ні; 1 – не маю відповіді)

Якість викладання дисциплін (дистанційна форма)

Активна

Кількість питань: 10

Ми прагнемо змін і вдосконалення навчання. Ваші відповіді допоможуть нам покращити дистанційні курси. Опитування анонімне. Результати будуть використані лише в узагальненому вигляді для підвищення якості освіти.

Якість реалізації освітньо-професійної програми

Активна

Кількість питань: 25

Просимо Вас дати відповіді на запитання з метою покращення освітнього процесу в KAI.

Рисунок 3.17 – Перелік шаблонів опитувань

Підрозділ «Результати» відображає зведений реєстр усіх опитувань із показниками охоплення - кількістю здобувачів, що взяли участь, та загальною

кількістю тих, хто мав пройти опитування (рис. 3.18). Для кожного запису відображається відсоток проходження, що дозволяє адміністратору швидко оцінити рівень залученості здобувачів.

Результати опитувань									
Показані 1-20 із 192 записів.									
#	Дії	Статус	Взяло участь	Мають пройти	Назва опитування	Тип опитування	Освітній ступінь	Форма зоді	
		Усі				Усі			
1			2 (1%)	194	Якість реалізації освітньої програми "Правознавство"	Загальне	Бакалавр	Діп	
2			0 (0%)	4	Оцінювання якості викладання дисциплін ОПП "Прикладна механіка композиційних конструкцій та технічних систем"	За ОК	Бакалавр		
3			2 (22%)	9	Оцінювання якості викладання дисциплін ОПП "Комп'ютерні системи та мережі"	За ОК	Бакалавр		
4			0 (0%)	27	Оцінювання якості викладання дисциплін ОПП "Дизайн архітектурного середовища"	За ОК	Бакалавр		
5			0 (0%)	4	Оцінювання якості викладання дисциплін ОПП "Політологія"	За ОК	Бакалавр		
6			0 (0%)	13	Оцінювання якості викладання дисциплін ОПП "Технології обробки великих даних"	За ОК	Магістр		
7			1 (8%)	13	Оцінювання якості викладання дисциплін ОПП "Технології обробки великих даних"	За ОК	Магістр		
8			0 (0%)	32	Оцінювання якості викладання дисциплін ОПП "Системи та технології кібербезпеки"	За ОК	Магістр		
9			0 (0%)	32	Оцінювання якості викладання дисциплін ОПП "Системи та технології кібербезпеки"	За ОК	Магістр		
10			0 (0%)	14	Оцінювання якості викладання дисциплін ОПП "Системи технічного захисту інформації, автоматизація її обробки"	За ОК	Магістр		
11			0 (0%)	14	Оцінювання якості викладання дисциплін ОПП "Системи технічного захисту інформації, автоматизація її обробки"	За ОК	Магістр		
12			1 (10%)	10	Оцінювання якості викладання дисциплін ОПП "Електронні технології інтернету речей"	За ОК	Магістр		
13			1 (10%)	10	Оцінювання якості викладання дисциплін ОПП "Електронні технології інтернету речей"	За ОК	Магістр		
14			0 (0%)	21	Оцінювання якості викладання дисциплін ОПП "Обслуговування повітряного руху"	За ОК	Магістр		
15			0 (0%)	21	Оцінювання якості викладання дисциплін ОПП "Обслуговування повітряного руху"	За ОК	Магістр		
16			0 (0%)	4	Оцінювання якості викладання дисциплін ОПП "Системи аеронавігаційного обслуговування"	За ОК	Магістр		
17			0 (0%)	4	Оцінювання якості викладання дисциплін ОПП "Системи аеронавігаційного обслуговування"	За ОК	Магістр		
18			8 (47%)	17	Оцінювання якості викладання дисциплін ОПП "Публічне управління та врегулювання"	За ОК	Магістр		
19			8 (47%)	17	Оцінювання якості викладання дисциплін ОПП "Публічне управління та врегулювання"	За ОК	Магістр		
20			13 (35%)	37	Оцінювання якості викладання дисциплін ОПП "Германські мови та літератури (переклад включно), перша - англійська"	За ОК	Магістр		

Рисунок 3.18 – Сторінка «Результати опитувань»

При переході до детального перегляду результатів конкретного опитування відображається повна аналітична сторінка (рис. 3.19). У верхній частині зазначаються основні параметри опитування - період проведення, цільові групи, освітня програма, освітній ступінь, факультет, семестр та форма опитування. Нижче розміщено блок фільтрів, що дозволяє звужувати вибірку за освітнім ступенем, формою навчання, факультетом, освітньою програмою, курсом та академічною групою. Зведена статистика представлена трьома інформаційними блоками: загальна кількість здобувачів, кількість тих хто взяв участь із відсотком охоплення та кількість тих хто не взяв участі. Нижче доступна деталізація за дисциплінами із розбивкою статистики проходження по кожному навчальному компоненту окремо. Для формування звітності передбачено кнопку «Експорт в Excel».

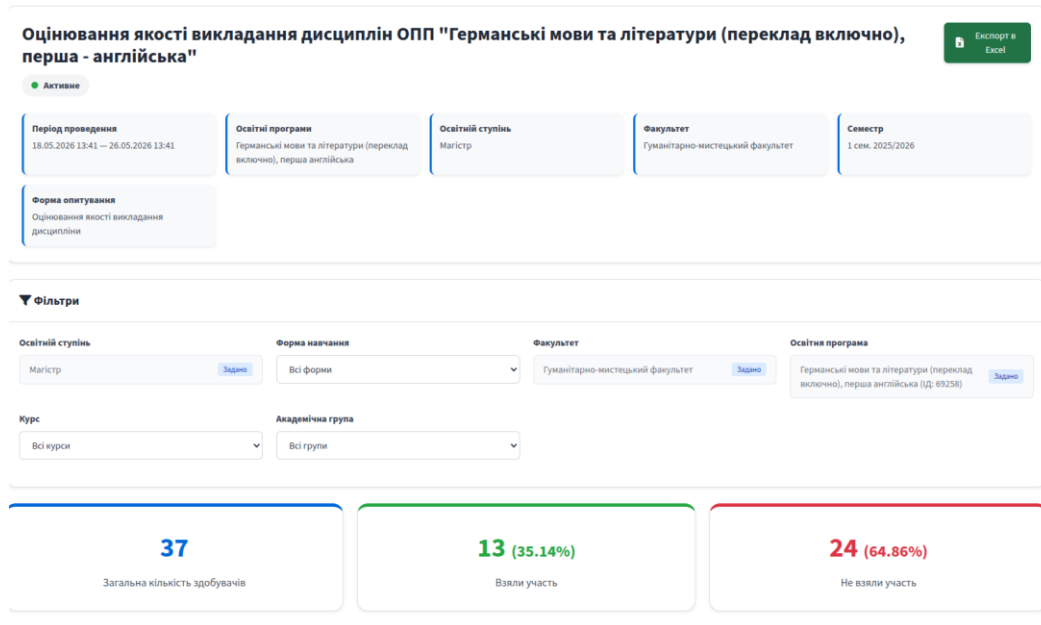


Рисунок 3.19 – Перегляд інформації про опитування

Експортований файл містить кілька аркушів (рис. 3.20): «Статистика відповідей» із детальним розподілом відповідей по питаннях та групах, «Відкриті відповіді» із текстовими коментарями здобувачів, «Пройшли опитування» та «Не пройшли опитування» із переліком здобувачів і їхніми даними - освітньою програмою, освітнім ступенем, формою навчання, факультетом, курсом, групою та ПІБ. Це дозволяє адміністрації отримати повну картину залученості здобувачів та використати дані для подальшого аналізу якості освітнього процесу:

Освітня програма	Освітній ступінь	Форма навчання	ID ОП	Факультет	Курс	Група	ПІБ	Person ID
Дизайн архітектурного середовища	Бакалавр	Денна	79515	Факультет архітектури, будівництва та дизайну	1	Б-G17-25-1-AP	Дроботько Ілля Олексійович	8666138
Дизайн архітектурного середовища	Бакалавр	Денна	79515	Факультет архітектури, будівництва та дизайну	1	Б-G17-25-1-AP	Гончаренко Владислав Максимович	9727163
Дизайн архітектурного середовища	Бакалавр	Денна	79515	Факультет архітектури, будівництва та дизайну	1	Б-G17-25-1-AP	Харченко Карина Олегівна	9664332
Дизайн архітектурного середовища	Бакалавр	Денна	79515	Факультет архітектури, будівництва та дизайну	1	Б-G17-25-1-AP	Ванка Владислава Тарасівна	9774224
Дизайн архітектурного середовища	Бакалавр	Денна	79515	Факультет архітектури, будівництва та дизайну	1	Б-G17-25-1-AP	Хмелик Марія Віталіївна	9749206
Дизайн архітектурного середовища	Бакалавр	Денна	79515	Факультет архітектури, будівництва та дизайну	1	Б-G17-25-1-AP	Атанова Аліна Юріївна	8632793
Дизайн архітектурного середовища	Бакалавр	Денна	79515	Факультет архітектури, будівництва та дизайну	1	Б-G17-25-1-AP	Рудаків Андрій Вадимович	9695151
Дизайн архітектурного середовища	Бакалавр	Денна	79515	Факультет архітектури, будівництва та дизайну	1	Б-G17-25-1-AP	Поліщук Данило Володимирович	9682148
Дизайн архітектурного середовища	Бакалавр	Денна	79515	Факультет архітектури, будівництва та дизайну	1	Б-G17-25-1-AP	Сумський Артем Олексійович	9673071
Дизайн архітектурного середовища	Бакалавр	Денна	79515	Факультет архітектури, будівництва та дизайну	1	Б-G17-25-1-AP	Габібзаде Раміль Арзу Оглі	9762451
Дизайн архітектурного середовища	Бакалавр	Денна	79515	Факультет архітектури, будівництва та дизайну	1	Б-G17-25-1-AP	Кучеренко Євген Володимирович	8434836
Дизайн архітектурного середовища	Бакалавр	Денна	79515	Факультет архітектури, будівництва та дизайну	1	Б-G17-25-1-AP	Менинпорок Віталія Віталіївна	8434568
Дизайн архітектурного середовища	Бакалавр	Денна	79515	Факультет архітектури, будівництва та дизайну	1	Б-G17-25-1-AP	Тимків Софія Юріївна	9738000
Дизайн архітектурного середовища	Бакалавр	Денна	79515	Факультет архітектури, будівництва та дизайну	1	Б-G17-25-1-AP	Марковська Єлизавета Віталіївна	9773854
Дизайн архітектурного середовища	Бакалавр	Денна	79515	Факультет архітектури, будівництва та дизайну	1	Б-G17-25-1-AP	Вдовиченко Ангеліна Євгенівна	9667941
Дизайн архітектурного середовища	Бакалавр	Денна	79515	Факультет архітектури, будівництва та дизайну	1	Б-G17-25-1-AP	Ткаченко Анастасія Віталіївна	9759268
Дизайн архітектурного середовища	Бакалавр	Денна	79515	Факультет архітектури, будівництва та дизайну	1	Б-G17-25-1-AP	Булутков Андрій Вікторович	9720709
Дизайн архітектурного середовища	Бакалавр	Денна	79515	Факультет архітектури, будівництва та дизайну	1	Б-G17-25-1-AP	Корчаківська Анастасія Павлівна	9781040
Дизайн архітектурного середовища	Бакалавр	Денна	79515	Факультет архітектури, будівництва та дизайну	1	Б-G17-25-1-AP	Гук Іван Віталійович	9666474
Дизайн архітектурного середовища	Бакалавр	Денна	79515	Факультет архітектури, будівництва та дизайну	1	Б-G17-25-1-AP	Терехова Анастасія Олександрівна	9585773
Дизайн архітектурного середовища	Бакалавр	Денна	79515	Факультет архітектури, будівництва та дизайну	1	Б-G17-25-1-AP	Сердечна Анастасія Андріївна	9696787
Дизайн архітектурного середовища	Бакалавр	Денна	79515	Факультет архітектури, будівництва та дизайну	1	Б-G17-25-1-AP	Капшученко Андрій Юрійович	9726350
Дизайн архітектурного середовища	Бакалавр	Денна	79515	Факультет архітектури, будівництва та дизайну	1	Б-G17-25-1-AP	Гриченко Валерія Олександрівна	9677638
Дизайн архітектурного середовища	Бакалавр	Денна	79515	Факультет архітектури, будівництва та дизайну	1	Б-G17-25-1-AP	Михайлова Марія Олександрівна	9755523
Дизайн архітектурного середовища	Бакалавр	Денна	79515	Факультет архітектури, будівництва та дизайну	1	Б-G17-25-1-AP	Збишко Ксенія Дмитрівна	9807992
Дизайн архітектурного середовища	Бакалавр	Денна	79515	Факультет архітектури, будівництва та дизайну	1	Б-G17-25-1-AP	Минзар Анастасія Сергіївна	7457081
Дизайн архітектурного середовища	Бакалавр	Денна	79515	Факультет архітектури, будівництва та дизайну	1	Б-G17-25-1-AP	Богданова Олександра Олександрівна	6769077

+ Статистика відповідей Відкриті відповіді Пройшли опитування Не пройшли опитування

Рисунок 3.20 – Експортовані результати опитування

		Коновал М.М.			ПІЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				66
Змн.	Арк.	№ докум.	Підпис	Дата		

Модуль «Опитування» реалізований також в особистому кабінеті здобувача освіти. При переході до розділу «Опитування» здобувач бачить перелік доступних для нього опитувань (рис. 3.21). Система відображає лише ті опитування, які призначені саме цьому здобувачу - або безпосередньо, або через належність до відповідної академічної групи, освітньої програми чи факультету. Для кожного опитування відображається його назва, тип, період проведення та поточний статус - «Відкрито» або «Завершено». Здобувач може розпочати проходження натисканням кнопки «Пройти»:

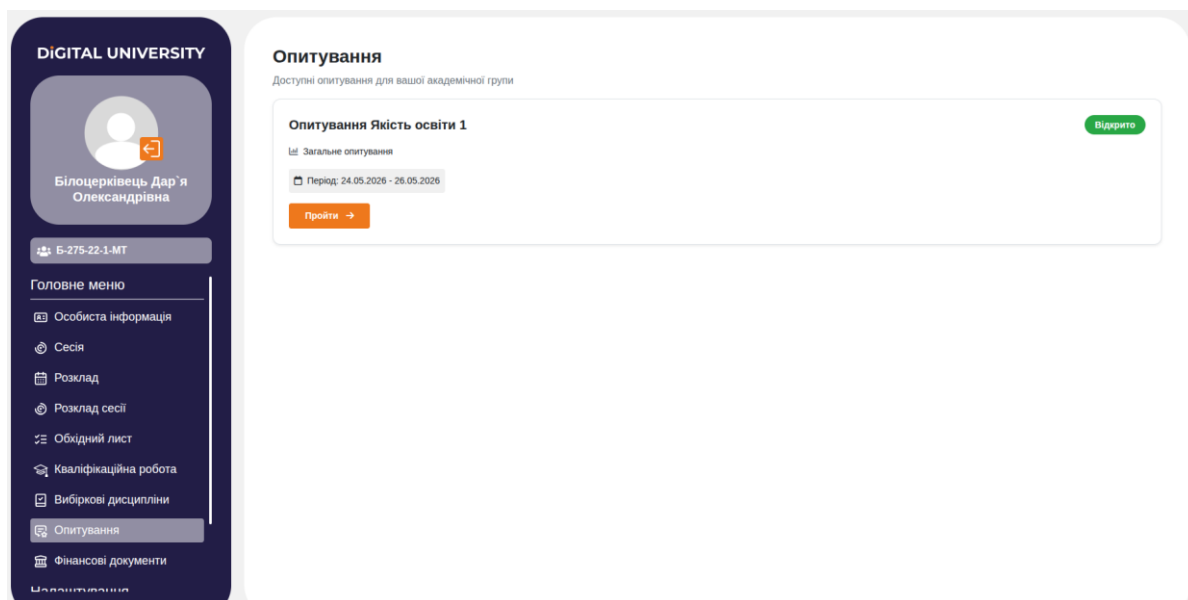


Рисунок 3.21 – Перелік доступних опитувань

Форма опитування в кабінеті здобувача відображається відповідно до типу опитування (рис. 3.22). Для загального опитування всі питання представлені в єдиній формі, згрупованій за спільними варіантами відповідей. Для опитування типу «За освітніми компонентами» здобувач проходить форму окремо для кожної навчальної дисципліни зі свого навчального плану поточного семестру. На початку форми відображається інформаційне повідомлення про анонімність опитування - відповіді не прив'язуються до особи здобувача, а використовуються виключно в узагальненому вигляді для підвищення якості освітнього процесу. Поточно обрана відповідь підсвічується помаранчевим кольором, що робить процес заповнення наочним та зручним. Питання, позначені зірочкою, є обов'язковими для відповіді:

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		67

DIGITAL UNIVERSITY

Білоцерківець Дар'я Олександрівна

Б-275-22-1-МТ

Головне меню

Особиста інформація

Сесія

Розклад

Розклад сесії

Обхідний лист

Кваліфікаційна робота

Вибіркові дисципліни

Опитування

Фінансові документи

Опитування Якість освіти 1

Опитування є анонімним. Ваші відповіді допоможуть покращити якість освітнього процесу.

Питання	Повністю задоволений	Скоріше так, ніж ні	Є над чим працювати	
Наскільки Ви задоволені своїм навчанням на цій програмі загалом? *	Повністю задоволений	Скоріше так, ніж ні	Є над чим працювати	
Питання	Так, навіть більше, ніж очікував(ла)	Загалом так	Чесно кажучи, очікував(ла) більшого.	
Чи справдилися ваші очікування від навчання? *	Так, навіть більше, ніж очікував(ла)	Загалом так	Чесно кажучи, очікував(ла) більшого.	
Питання	Так, було усе необхідне	Обладнання є, але частину варто оновити	Відчувається значний брак техніки та програмного забезпечення	Спеціального оснащення навчання не вимагало
Чи вистачало Вам сучасного обладнання та ресурсів у лабораторіях для навчання? *	Так, було усе необхідне	Обладнання є, але частину варто оновити	Відчувається значний брак техніки та програмного забезпечення	Спеціального оснащення навчання не вимагало
Питання	Так, усе необхідне було в наявності	Загалом так, але деякі ресурси складно знайти	Ні, відчувається брак актуальної інформації або доступу до джерел	
Чи вистачало Вам інформаційної підтримки під час навчання (доступ до баз даних, репозитарію, наповнення сайту кафедри та гугл класів)? *	Так, усе необхідне було в наявності	Загалом так, але деякі ресурси складно знайти	Ні, відчувається брак актуальної інформації або доступу до джерел	
Питання	Так, навантаження було збалансованим	Частково, але в цілому встигав(ла)	Ні, часу катастрофічно бракувало, забагато завдань	
Чи встигали Ви поєднувати навчання, виконання самостійної роботи, підготовку завдань та особистий час (work-life balance)? *	Так, навантаження було збалансованим	Частково, але в цілому встигав(ла)	Ні, часу катастрофічно бракувало, забагато завдань	

Рисунок 3.22 – Опитування кабінет студента

3.3. Тестування роботи розробленого додатку

У межах розробки модулів «Кваліфікаційні роботи» та «Опитування здобувачів освіти» було проведено ручне функціональне тестування, що передбачає перевірку основного функціоналу без використання автоматизованих скриптів. Такий підхід дозволив максимально наблизити перевірку до реальних сценаріїв використання, виявити неочевидні помилки у крайових умовах, а також оцінити зручність інтерфейсу та логіку навігації для різних категорій користувачів. Основною метою тестування було переконатися у стабільності роботи системи, коректності виконання кожного модуля та відповідності реалізованого функціоналу встановленим бізнес-вимогам і очікуванням адміністрації та власників системи.

Першим було протестовано перегляд реєстру кваліфікаційних робіт для адміністративного персоналу, методистів та представників деканатів. Таблиця кваліфікаційних робіт коректно відображає повний перелік записів, підтримує фільтрацію за роком випуску, групою та статусом роботи. Під час тестування особлива увага приділялася поведінці інтерфейсу при одночасному виведенні великої кількості записів - дані завантажуються динамічно із застосуванням механізмів пагінації через Yii2 DataProvider, і навіть при великому обсязі записів система не демонструє помітних затримок. Додатково перевірено роботу

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		68

текстового пошуку за прізвищем здобувача та темою роботи, де відпрацьовують алгоритми нечіткого збігу в SQL-запитах. Рольова модель доступу функціонує коректно - кожен користувач бачить лише ті роботи, до яких має доступ відповідно до своєї ролі: завідувач кафедри отримує доступ до робіт своєї кафедри, тоді як здобувач бачить виключно власну картку. Відображення, фільтрація, швидкість відгуку інтерфейсу та рольове розмежування доступу працюють належним чином.

Найбільш критичним інтеграційним вузлом системи є підписання документів через сервіс «Дія», якому було приділено особливу увагу під час тестування. Вхідною точкою сценарію виступав виклик відповідної дії в `QualificationWorkViewController`. Система коректно генерує QR-код та deeplink, передає унікальний хеш PDF-файлу до API «Дії» через мікросервіс Node.js та переходить у стан очікування підтвердження підписання. Випробування проводилися на реальних мобільних пристроях під керуванням операційних систем Android та iOS. Після успішного підписання документа користувачем у застосунку «Дія» за допомогою біометричних даних асинхронний вебхук мікросервісу Node.js коректно прийняв підписаний файл. У процесі обробки за допомогою інтегрованої бібліотеки ІТ було здійснено перевірку цілісності пакета та підвищено рівень підпису до формату CAdES-X-Long, що включає повні дані про чинність сертифікатів та позначки часу. Після цього файл збережено на сервері, а статус роботи в базі даних MySQL оновився автоматично без потреби перезавантаження сторінки. Паралельно протестовано альтернативний спосіб підписання через КЕП - зчитування файлових ключів у форматах `dat`, `prfx`, `zs2` та апаратних носіїв через веббібліотеку ІТ працює коректно, забезпечуючи тотожний рівень юридичної сили підпису. Обидва способи підписання функціонують згідно з технічним завданням, криптографічні перетворення відбуваються без помилок, підписані файли зберігаються коректно.

Послідовний перехід між статусами кваліфікаційної роботи - від первинного завантаження здобувачем до фінального статусу завершення

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		69

навчання - також пройшов перевірку. Впроваджений у систему автомат станів було перевірено на стійкість до некоректних дій користувачів: моделювалися навмисні спроби порушення логіки процесу через надсилання прямих HTTP-запитів до контролера в обхід інтерфейсу, а також спроби виконати дії, що не відповідають поточному статусу або ролі користувача. Усі ці спроби були коректно заблоковані системою на рівні шару бізнес-логіки з генерацією відповідних виключень безпеки та HTTP-статусу 403. Перевірено, що кожен статус активує відповідний блок дій лише для уповноваженого учасника процесу - наприклад, перевести роботу зі статусу рецензування до допуску на захист може лише науковий керівник або рецензент, але не сам здобувач. Історія змін статусів фіксується в окремій аудит-таблиці бази даних із точним часом кожного переходу та ідентифікатором користувача, який вніс зміни. Логіка переходів між статусами, захист від несанкціонованих дій та розмежування доступу за ролями працює відповідно до архітектурних вимог.

Процес створення опитувань обох типів - загального та за освітніми компонентами - було перевірено через адміністративний інтерфейс PollController. Налаштування цільової аудиторії протестовано для різних комбінацій: окремо вибраний здобувач освіти, конкретна академічна група, ціла освітня програма або весь факультет. Система продемонструвала точність вибірки на основі складних реляційних зв'язків у MySQL - опитування динамічно відображається в особистому кабінеті лише тих здобувачів, які підпадають під задані критерії. Окремо перевірено функцію копіювання опитування для нового навчального семестру - структура анкети, типи питань та зв'язки з групами і освітніми програмами копіюються коректно, не порушуючи цілісності вже існуючих історичних даних у суміжних таблицях. Гнучке налаштування цільової аудиторії, валідація конструктора питань та механізм копіювання опитувань працюють стабільно.

Проходження опитування здобувачем тестувалося безпосередньо в особистому кабінеті з використанням різних пристроїв - ПК, планшетів та смартфонів. Для загального опитування форма відображається єдиним логічним

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				70
Змн.	Арк.	№ докум.	Підпис	Дата		

блоком, розділеним на тематичні секції. Для опитування типу «За ОК» система автоматично визначає перелік дисциплін, які здобувач вивчав у поточному семестрі згідно з навчальним планом, та формує окремі блоки питань для кожної дисципліни. Особлива увага приділялася перевірці принципу анонімності - шляхом прямого аналізу записів у базі даних після відправки форми підтверджено, що система коректно фіксує сам факт проходження опитування здобувачем для контролю явки та блокування повторних спроб, проте самі відповіді зберігаються в ізольованій таблиці без жодного зв'язку з персональними даними або ідентифікатором користувача. Повторне проходження вже завершеного або протермінованого опитування заблоковане як на рівні інтерфейсу, так і на рівні серверних перевірок у контролері. Стовідсоткова анонімність відповідей, коректність відображення форми та надійність блокування повторного проходження підтверджено.

Відображення статистики проходження опитувань перевірялося через PollResultsController для різних аналітичних зрізів - за освітнім ступенем, формою навчання, факультетом, освітньою програмою та курсом. Усі фільтри працюють коректно, оновлюючи статистичні показники та відсоток залученості здобувачів у режимі реального часу завдяки оптимізованим агрегаційним SQL-запитам. Експорт результатів у формат Excel тестувався на великих масивах даних - отримані файли перевірялися на коректність кодування українських символів та повноту перенесення інформації на всіх чотирьох аркушах: «Статистика відповідей» із кількісними та відсотковими показниками по кожному питанню, «Відкриті відповіді» із текстовими коментарями здобувачів без прив'язки до імен, «Пройшли опитування» та «Не пройшли опитування» із переліками здобувачів для адміністративного моніторингу. При спробі несанкціонованого доступу до скачування файлів звітів користувачами без відповідних прав система коректно видає помилку доступу. Аналітика, розрахунок статистичних зрізів та експорт результатів працюють згідно з технічним завданням, розбіжностей чи помилок у даних не виявлено.

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		71

Загалом проведене ручне функціональне тестування підтвердило повну працездатність та експлуатаційну готовність обох модулів. Спільна робота основного Yii2-застосунку та інтеграційного мікросервісу на Node.js демонструє високу стабільність при виконанні криптографічних операцій та взаємодії з державним API «Дія». Під час випробувань не виявлено критичних помилок, витоків пам'яті або порушень цілісності даних у MySQL. Система повністю готова до етапу дослідної експлуатації на реальних потужностях університету.

Висновки до третього розділу

У третьому розділі було здійснено розгортання вебзастосунку на серверному середовищі, що відповідає технічним параметрам, встановленим на початковому етапі розробки. Після розгортання проведено комплексне тестування реалізованого функціоналу, що дало змогу перевірити стабільність та коректність роботи ключових компонентів системи.

У межах модуля «Кваліфікаційні роботи» реалізовано інструменти перегляду реєстру робіт із фільтрацією за роком випуску, групою та статусом, покрокову схему відстеження поточного стану кожної роботи, рольову модель доступу для всіх учасників процесу - здобувача, наукового керівника, нормоконтролера, завідувача кафедри та декана. Ключовим досягненням стала реалізація повноцінної інтеграції з державним сервісом електронного підпису «Дія», що забезпечує юридично значуще підписання документів кваліфікаційних робіт як через QR-код на мобільному пристрої, так і через deeplink з автоматичним перенаправленням у застосунок. Додатково реалізовано підписання через КЕП за допомогою бібліотеки ІТТ безпосередньо на порталі. Експорт даних у формат Excel підтверджено на коректність кодування та повноту вивантажених записів.

У межах модуля «Опитування здобувачів освіти» реалізовано повний цикл роботи з опитуваннями - від створення форм та налаштування цільової аудиторії до збору анонімних відповідей та аналізу результатів. Система підтримує два типи опитувань, гнучке призначення цільової аудиторії на

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		72

різних рівнях деталізації, деталізовану аналітику у розрізі груп та дисциплін, а також експорт повної статистики у формат Excel з кількома аркушами даних.

Результати тестування підтвердили повну працездатність розроблених модулів, їх відповідність заявленим функціональним вимогам та готовність до подальшої експлуатації. Таким чином, поставлені завдання були успішно реалізовані, що свідчить про досягнення головної мети розробки.

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		73

ВИСНОВКИ

У межах виконання кваліфікаційної роботи було реалізовано повний життєвий цикл розробки двох функціональних модулів інформаційної системи онлайн-менеджменту освітніх процесів Digital University UA - «Кваліфікаційні роботи» та «Опитування здобувачів освіти». Основна мета полягала у цифровізації процесів управління кваліфікаційними роботами із забезпеченням юридично значущого електронного підписання документів через державний сервіс «Дія», а також у створенні інструментів систематизованого збору та аналізу зворотного зв'язку від здобувачів освіти.

На етапі підготовки виконано порівняльний аналіз сучасних вебтехнологій, що використовуються для створення масштабованих інформаційних систем. За результатами техніко-економічного обґрунтування обрано мову програмування PHP у поєднанні з фреймворком Yii2, що забезпечило високу продуктивність, модульність коду та захищену роботу з базами даних. Вибір мікросервісної архітектури зумовлений необхідністю винесення інтеграції з API сервісу «Дія» в окремий незалежний компонент на платформі Node.js, що забезпечує можливість його оновлення без впливу на основний застосунок.

У рамках проєктного етапу сформульовано функціональні та нефункціональні вимоги до програмного продукту. Структуровано основні логічні процеси, розроблено UML-діаграми, що візуалізують сценарії використання модулів та взаємозв'язки їхніх ключових компонентів. Окрему увагу приділено проєктуванню механізму інтеграції з API «Дії» - реалізовано повний цикл підписання документів через QR-код та deeplink з підвищенням рівня підпису до формату CAdES-X-Long, а також альтернативне підписання через КЕП за допомогою бібліотеки ІТТ безпосередньо на порталі. Для модуля опитувань реалізовано гнучку систему призначення цільової аудиторії на різних рівнях деталізації та механізм анонімного збору відповідей.

Фінальний етап включав розгортання модулів на серверному середовищі та проведення серії функціональних тестів. Було перевірено коректність

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		74

механізмів автентифікації, рольової моделі доступу, навігації між статусами кваліфікаційних робіт, підписання документів через сервіс «Дія», проходження опитувань здобувачами в особистому кабінеті, відображення аналітики результатів опитувань, а також генерації звітності у форматі Excel.

Таким чином, реалізовані модулі «Кваліфікаційні роботи» та «Опитування здобувачів освіти» продемонстрували здатність підвищувати ефективність і прозорість адміністративних процесів у сфері освіти, а також готовність до повноцінного впровадження в комплексну систему керування закладом вищої освіти.

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		75

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Цифровізація освітнього середовища / Семеняко Ю.Б., Брюховецька І.В., Бохонько О.Є. – Класичний приватний університет, 2023. – 186 с.
2. Moodle Learning Management System [Електронний ресурс] – Режим доступу до ресурсу: <https://moodle.org/>
3. Canvas LMS – Instructure [Електронний ресурс] – Режим доступу до ресурсу: <https://www.instructure.com/canvas>
4. OpenEduCat – Open Source ERP for Education [Електронний ресурс] – Режим доступу до ресурсу: <https://www.openeducat.org/>
5. Yii 2.0 Framework Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://www.yiiframework.com/doc/guide/2.0/en>
6. Data Structures and Algorithm Analysis in PHP / Michael A. Redmond – Morgan Kaufmann, 2014. – 400 с.
7. PHP Cookbook: Modern Code Solutions for Professional Developers / Eric Mann – O'Reilly Media, 2023. – 420 с.
8. Building Microservices with Node.js / Daniel Kapexhiu – Packt Publishing, 2024. – 324 с.
9. Node.js Design Patterns: Design and implement production-grade Node.js applications / Mario Casciaro, Luciano Mammino – Packt Publishing, 2020. – 660 с.
10. Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript and Web Graphics / Jennifer Niederst Robbins – O'Reilly, 2021. – 640 с.
11. Introduction to Algorithms / Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein – MIT Press, 2022. – 1312 с.
12. Algorithms in a Nutshell / George T. Heineman, Gary Pollice, Stanley Selkow – O'Reilly, 2017. – 800 с.
13. Microservices: How To Make Your Application Scale / Dragoni et al. – Cornell University, 2017. – 346 с.
14. Building Microservices: Designing Fine-Grained Systems / Sam Newman – O'Reilly Media, 2021. – 616 с.

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Вакалюк Т.А.				76
Змн.	Арк.	№ докум.	Підпис	Дата		

15. MySQL Crash Course / Ben Forta – Addison-Wesley, 2024. – 400 с.
16. PHP, MySQL and Apache / Julie C. Meloni – Peachpit Press, 2022. – 512 с.
17. High Performance MySQL: Optimization, Backups, Replication & Load Balancing / Baron Schwartz, Peter Zaitsev, Vadim Tkachenko – O'Reilly, 2023. – 1024 с.
18. MongoDB in Action, Third Edition / Arek Borucki – Manning, 2023. – 450 с.
19. Data Modeling for MongoDB / Steve Hoberman – O'Reilly, 2021. – 320 с.
20. API Design Patterns / JJ Geewax – Manning, 2022. – 350 с.
21. Software Testing: A Craftsman's Approach / Paul C. Jorgensen – CRC Press, 2021. – 464 с.
22. Pro PHP Application Performance: Tuning PHP, MySQL and Apache / Armando Padilla – Apress, 2011. – 240 с.
23. JavaScript: The Good Parts / Douglas Crockford, 2008. – 176 с.
24. UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design / Jim Arlow, Ila Neustadt – Addison-Wesley, 2005. – 592 с.
25. Документація API сервісу «Дія» [Електронний ресурс] – Режим доступу до ресурсу: <https://api2.diia.gov.ua>
26. PhpSpreadsheet Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://phpspreadsheet.readthedocs.io/>
27. UML Distilled: A Brief Guide to the Standard Object Modeling Language / Martin Fowler – Addison-Wesley, 2018. – 208 с.
28. Software Requirements / Karl Wieggers, Joy Beatty – Microsoft Press, 2014. – 648 с.
29. Web Browser Architecture and Rendering Engine Internals [Електронний ресурс] – Режим доступу до ресурсу: <https://www.html5rocks.com/en/tutorials/internals/howbrowserswork/>
30. The JSON Data Interchange Standard [Електронний ресурс] – Режим доступу до ресурсу: <https://www.json.org/json-en.html>

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		77

ДОДАТКИ

		Коновал М.М.			ІІЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Вакалюк Т.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		78

```

public function actionGetStudyPrograms()
{
    Yii::$app->response->format = Response::FORMAT_JSON;
    $speciality_id = Yii::$app->request->get('speciality_id');
    if (!$speciality_id) {
        return [];
    }

    $studyPrograms = EdeboUniversityStudyPrograms::find()
        ->select([
            'edebo_university_study_programs.speciality_id',
            'edebo_university_study_programs.speciality_name',
            'edebo_university_study_programs.qualification_group_name',
            'edebo_university_study_programs.university_study_program_id',
            'edebo_university_study_programs.study_program_name'
        ])
        ->distinct()
        ->innerJoin(
            'edebo_dictionary_cdc_specialityalls',
            'edebo_university_study_programs.speciality_id =
edebo_dictionary_cdc_specialityalls.speciality_id
AND
(edebo_university_study_programs.university_specialization_id =
edebo_dictionary_cdc_specialityalls.specialization_id OR
(edebo_dictionary_cdc_specialityalls.specialization_id IS
NULL))'
        )
        ->where(['edebo_university_study_programs.speciality_id'
=> $speciality_id])
        ->orderBy(['edebo_university_study_programs.study_program_name' =>
SORT_ASC])
        ->asArray()
        ->all();

    return ArrayHelper::map(
        $studyPrograms,
        'university_study_program_id',
        function ($model) {
            return $model['study_program_name'] . ' (' .
$model['qualification_group_name'] . '; ID: ' .
$model['university_study_program_id'] . ')';
        }
    );
}

/**
 * @return array
 */
public function actionGetEducationDates()
{
    Yii::$app->response->format = Response::FORMAT_JSON;

```

		Коновал М.М.			ІІЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Вакалюк Т.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		79

```

        $program_id = Yii::$app->request->get('university_study_program_id');
        if (!$program_id) {
            return [];
        }
        $students = EdeboStudents::find()
            ->where(['university_study_program_id' => $program_id])
            ->select('education_date_end')
            ->distinct()
            ->orderBy(['education_date_end' => SORT_ASC])
            ->asArray()
            ->all();
        return array_values(array_unique(array_column($students,
            'education_date_end')));
    }

    /**
     * @param string $q
     * @return array
     */
    public function actionSearchStaff(string $q = ''): array
    {
        Yii::$app->response->format = Response::FORMAT_JSON;

        $query = Staff::find()
            ->select([
                'id' => 'id',
                'text' => 'full_name',
            ])
            ->where(['like', "full_name", $q])
            ->orderBy("full_name ASC")
            ->limit(20);

        $persons = $query->asArray()->all();

        return [
            'items' => array_map(function ($author) {
                return [
                    'id' => $author['id'],
                    'text' => $author['text']
                ];
            }, $persons)
        ];
    }

    /**
     * @param string $q
     * @return array
     */
    public function actionSearchStudents(string $q = ''): array
    {
        Yii::$app->response->format = Response::FORMAT_JSON;

        $query = EdeboStudents::find()
            ->select([

```

		Коновал М.М.			ІІЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Вакалюк Т.А.				80
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        'id' => 'education_id',
        'person_fio',
        'education_id'
    ])
    ->orderBy('person_fio ASC')
    ->limit(20);

if (strlen($q)) {
    $query->andWhere([
        'or',
        ['like', 'person_fio', $q],
        ['like', 'education_id', $q]
    ]);
}

$students = $query->asArray()->all();

return [
    'items' => array_map(function ($student) {
        return [
            'id' => $student['id'],
            'text' => "{$student['person_fio']}
({$student['education_id']})"
        ];
    }, $students)
];

};

public function actionIndex()
{
    $searchModel = new QualificationWorkViewSearch();
    Yii::$app->session->set('qualification_work_view_filters',
    Yii::$app->request->queryParams);

    $availableWorks = $this->qualificationWorkService-
    >getAvailableQualificationWorks();

    $params = Yii::$app->request->queryParams;
    if (!isset($params['QualificationWorkViewSearch'])) {
        $params['QualificationWorkViewSearch'] = [];
    }

    $workIds = array_column($availableWorks, 'id');

    $searchModel->qualification_work_ids = $workIds;

    $dataProvider = $searchModel->search($params);
    $years = (new Query())
        ->select(['year' => 'YEAR(education_date_end)'])
        ->from('edebo_students')
        ->where(['not', ['education_date_end' => null]])
        ->groupBy(['year'])
        ->orderBy(['year' => SORT_DESC])
        ->column();

    return $this->render('index', [

```

		Коновал М.М.			ІІЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Вакалюк Т.А.				81
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        'searchModel' => $searchModel,
        'dataProvider' => $dataProvider,
        'years' => $years,
    });
}

/**
 * @param $id
 * @return string
 * @throws NotFoundHttpException
 * @throws Exception
 */
public function actionView($id)
{
    $model = $this->findModel($id);

    $educationResults = ExamSheetLastPoint::find()
        ->alias('eslp')
        ->select(['eslp.*', 'usn.notation'])
        ->leftJoin('exam_sheet_notation usn', 'eslp.points_final =
usn.points')
        ->with([
            'examSheet.educationalComponent',
            'examSheet.formOfControl',
            'examSheet.educationalComponentType',
        ])
        ->where(['student_id' => $model->education_id])
        ->asArray()
        ->all();

    $reenrollmentResults = ExamSheetReenrollmentPoint::find()
        ->alias('esrp')
        ->select([
            'esrp.*',
            'usn.notation AS notation_reenrollment',
        ])
        ->leftJoin('exam_sheet_notation usn', 'esrp.points =
usn.points')
        ->leftJoin(
            'exam_sheet_reenrollment_institution esi',
            'esi.id =
esrp.exam_sheet_reenrollment_institution_id',
        )
        ->with([
            'examSheetReenrollmentInstitution',
            'educationalComponent',
            'formOfControl',
            'educationalComponentType',
        ])
        ->where(['esi.education_id' => $model->education_id])
        ->asArray()
        ->all();

    $educationResultsProvider = new ArrayDataProvider([
        'allModels' => $educationResults,
    ]);
}

```

		Коновал М.М.			ІІЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Вакалюк Т.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		82

```

        'pagination' => false,
    ]);

    $reenrollmentResultsProvider = new ArrayDataProvider([
        'allModels' => $reenrollmentResults,
        'pagination' => false,
    ]);

    $allStatuses = GraduationWorkStatus::find()->all();
    $messages = $model->getGraduationWorkMessages()-
>orderBy(['datetime' => SORT_DESC]);
    $allMessages = $messages->all();
    $statusHistory = $model->getGraduationWorkStatusHistories()-
>orderBy(['datetime' => SORT_DESC])->all();

    $statusHistoryProvider = new ArrayDataProvider([
        'allModels' => $statusHistory,
    ]);

    $userRoles = $this->qualificationWorkService-
>getUserRoles($id);

    $newMessages = false;
    if (!empty($statusHistory)) {
        $lastChangeTime = $statusHistory[0]?->datetime;
        $newMessages = $messages->where(['is_student' => 1])
            ->andWhere(['>=', 'datetime', $lastChangeTime])
            ->exists();
    }

    return $this->render('view', [
        'model' => $model,
        'allStatuses' => $allStatuses,
        'messages' => $allMessages,
        'newMessages' => $newMessages,
        'statusHistoryProvider' => $statusHistoryProvider,
        'educationResultsProvider' => [
            'base' => $educationResultsProvider,
            'reenrollment' => $reenrollmentResultsProvider,
        ],
        'userRoles' => $userRoles,
    ]);
}

public function getAvailableQualificationWorks(): array
{
    $result = [];

    if ($this->isAdmin()) {
        return $this->getAllQualificationWorks();
    }

    if ($this->isDean() || $this->isDeputyDean() || $this-
>isDeanOfficeEmployee()) {
        $result = array_merge($result, $this-
>getDeanQualificationWorks());
    }
}

```

		Коновал М.М.			ІІЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Вакалюк Т.А.				83
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }

    if ($this->isDepartmentHead()) {
        $result = array_merge($result, $this->getDepartmentHeadQualificationWorks());
    }

    if ($this->isCommissionMember()) {
        $result = array_merge($result, $this->getCommissionMemberQualificationWorks());
    }

    if ($this->isTeacher()) {
        $result = array_merge($result, $this->getTeacherQualificationWorks());
    }

    $uniqueWorks = [];
    foreach ($result as $work) {
        $uniqueWorks[$work['id']] = $work;
    }

    return array_values($uniqueWorks);
}

$(document).ready(function () {
    const qualificationWorkId = params.get('id');

    const showAgreeStep = () => {
        $('#signStep').hide();
        $('#agreeStep').show();
    };

    const openDiiaSignModal = diiaSignModal({
        deepLinkUrl: `/qualificationWork/qualification-work-view/get-deeplink?qualificationWorkId=${qualificationWorkId}`,
        checkSignStatusUrl: `/qualificationWork/qualification-work-view/check-sign-status?workId=${qualificationWorkId}`,
        serviceName: 'Накладання підпису на файл кваліфікаційної роботи',
        onSuccessSign: showAgreeStep,
    });

    const openKEPSignModal = (e) => openSignModal(e, showAgreeStep);

    $('#signWithKEP').on('click', openKEPSignModal);
    $('#signWithDiia').on('click', openDiiaSignModal);
});

function diiaSignModal(options) {
    const QR_CODE_EXPIRE_SECONDS = 180;
    const QR_CODE_POLLING_SECONDS = 2;

    const {
        deepLinkUrl,
        checkSignStatusUrl,
        serviceName,
        returnLink = window.location.href,
    } = options;

```

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Вакалюк Т.А.				84
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        filesInputSelector = null,
        onSuccessSign,
        onErrorSign,
    } = options;

    const $modal = $('#modal-diia-sign');
    const $status = $modal.find('#diia-status');
    const $qrContainer = $modal.find('#diia-qr-container');
    const $qrLoader = createLoader($qrContainer[0]);

    const isMobile = checkIsMobile();
    let pollingInterval = null;
    let timerInterval = null;

    function formatTime(seconds) {
        const min = String(Math.floor(seconds / 60)).padStart(2, '0');
        const sec = String(seconds % 60).padStart(2, '0');

        return `${min}:${sec}`;
    }

    function checkIsMobile() {
        return /Android|webOS|iPhone|iPad|iPod|BlackBerry|IEMobile|Opera
        Mini/i.test(navigator.userAgent);
    }

    function setStatusMessage(type, statusMessage) {
        const statusTypes = {
            'success': 'text-success',
            'error': 'text-danger',
            'info': 'text-info',
        };

        const statusClass = statusTypes?.[type];
        const $message = $('<span>');

        if (statusClass) {
            $message.addClass(statusClass);
        }

        $message.text(statusMessage);
        $status.html($message);
    }

    function resetStatusMessage() {
        $status.empty();
    }

    function setStatusGetDeepLinkError(message) {
        setStatusMessage('error', 'Помилка отримання QR-коду');
        console.error('Error get deep link:', message);
    }

    let loadedFileData = null;
    const params = new URLSearchParams(window.location.search);

```

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Вакалюк Т.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		85

```

const SIGN_WIDGET_PARENT_ID = "sign-widget-parent";
const SIGN_WIDGET_ID = "sign-widget";
const SIGN_WIDGET_URI = "https://eu.iit.com.ua/sign-
widget/v20240301_new/";

function initializeFileProcessing() {
    const signWidgetParent = document.querySelector("#sign-widget-
parent");
    const fileInfoContainer = document.querySelector('#file-info');
    const loader = createLoader(fileInfoContainer);

    loader.show();
    const currentId = params.get('id');

    fetch(`/qualificationWork/qualification-work-view/get-work-
file?qualificationWorkId=${currentId}`)
        .then(response => {
            if (!response.ok) {
                throw new Error("Файл не знайдено або не вдалося отримати.");
            }
            return response.json();
        })
        .then(fileData => {
            if (fileData.error) {
                loader.remove();
                console.error(fileData.error);
                alert("Помилка: " + fileData.error);
                return;
            }

            loadedFileData = {
                qualificationWorkId: currentId,
                ...fileData,
            };
            fileInfoContainer.innerHTML = `

            let fileSize = (atob(fileData.fileContent).length / 1024 /
1024).toFixed(2) + ' MB';

            let fileBlock = document.createElement('div');
            fileBlock.className = 'file-block';
            fileBlock.innerHTML = `
                <div class="file-card">
                    <div class="file-details">
                        <p class="file-name"><i class="fa-regular fa-
file-lines"></i> ${fileData.fileName}</p>
                        <p class="file-size">${fileSize}</p>
                        <a href="data:application/octet-
stream;base64,${fileData.fileContent}" download="${fileData.fileName}"
class="btn btn-dark download-button">
                            Завантажити файл
                        </a>
                    </div>
                </div>
            `;
        });

```

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Вакалюк Т.А.				86
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        `;

        fileInfoContainer.appendChild(fileBlock);

        loader.remove();
        signWidgetParent.style.display = "block";
        document.querySelector('#step2').style.display = 'block';
    })
    .catch(error => {
        loader.remove();
        console.error("Помилка:", error);
        alert("Виникла помилка при отриманні файлу.");
    });
}

var euSign = new EndUser(
    SIGN_WIDGET_PARENT_ID,
    SIGN_WIDGET_ID,
    SIGN_WIDGET_URI,
    EndUser.FormType.ReadPKey,
);

function onConfirmKSPOperation(event) {
    var node = '';
    node += '<a href="' + event.url + '>';
    node += '';
    node += '</a>';

    document.getElementById('qr-code-block').innerHTML = node;
    document.getElementById('qr-code-block').style.display = 'block';
}

function base64ToUint8Array(base64) {
    const binaryString = atob(base64);
    const length = binaryString.length;
    const bytes = new Uint8Array(length);
    for (let i = 0; i < length; i++) {
        bytes[i] = binaryString.charCodeAt(i);
    }
    return bytes;
}

function onSign(successCallback) {
    const signButton = document.querySelector("#buttonSignData");

    if (!loadedFileData) {
        alert("Файл не завантажено");
        return;
    }

    signButton.loadingState(true);

    let data = base64ToUint8Array(loadedFileData.fileContent);
    var external = true;

```

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Вакалюк Т.А.				87
Змн.	Арк.	№ докум.	Підпис	Дата		

```

var asBase64String = true;
var signAlgo = EndUser.SignAlgo.DSTU4145WithGOST34311;
var signType = EndUser.SignType.CAdES_X_Long;

euSign.SignData(data, external, asBase64String, signAlgo, null,
signType)
    .then(function (sign) {
        const signFormData = new FormData();
        signFormData.append('file', new Blob([sign]));
        signFormData.append('qualificationWorkId',
loadedFileData.qualificationWorkId);
        signFormData.append('fileName',
loadedFileData.uploadedFileName);

        $.ajax({
            url: '/qualificationWork/qualification-work-view/upload-sign-
file',
            type: 'POST',
            data: signFormData,
            processData: false,
            contentType: false,
            success: function () {
                successCallback?.();
            },
            error: function () {
                console.error(`Помилка при передачі підпису для файлу
${loadedFileData.fileName} на сервер`);
            },
        });
    })
    .catch(function (e) {
        alert('Виникла помилка при підписі файлу ' +
loadedFileData.fileName +
'. Опис помилки: ' + (e.message || e));
    });
}

function initializeSigningProcess() {
    euSign.ReadPrivateKey()
        .then(function () {
            return euSign.AddEventListener(
                EndUser.EventType.ConfirmKSPOperation,
                onConfirmKSPOperation,
            );
        })
        .then(function () {
            document.getElementById('qr-code-block').style.display = 'none';
            document.getElementById('sign-widget-parent').style.display =
'none';
            document.getElementById('sign-data-block').style.display =
'block';
        })
        .catch(function (e) {
            alert('Виникла помилка при зчитуванні ос. ключа. ' +
'Опис помилки: ' + (e.message || e));
        });
}

```

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				88
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    });
}

function startSignProcess() {
    initializeFileProcessing();
    initializeSigningProcess();
}

const axios = require('axios');
const crypto = require('crypto');
const fs = require('fs').promises; // Додано для асинхронного читання файлів
const path = require('path');      // Додано для роботи зі шляхами

// Припустимо, що getSignedFilesFromDB знаходиться там само, де і saveRequestId
const {getSignedFilesFromDB} = require("../../services/DatabaseService/DatabaseService");

class SignController {
    constructor() {
        this.sessionToken = null;
        this.sessionTokenTimestamp = null;
        this.tokenExpiryTime = 2 * 60 * 60 * 1000;

        this.acquirerToken = process.env.ACQUIRER_TOKEN;
        this.authAcquirerToken = process.env.AUTH_ACQUIRER_TOKEN;
        this.diiaHost = process.env.NODE_ENV === "production" ?
        "https://api2.diia.gov.ua" : "https://api2.diia.gov.ua";
    }

    async getSessionToken() {
        try {
            console.log('HOST:', this.diiaHost);
            console.log('TOKEN:', this.acquirerToken);
            console.log('AUTH:', this.authAcquirerToken);

            const response = await axios.get(
                `${this.diiaHost}/api/v1/auth/acquirer/${this.acquirerToken}`,
                {
                    headers: {
                        Authorization: `Basic ${this.authAcquirerToken}`,
                        Accept: "application/json",
                    },
                }
            );

            this.sessionToken = response.data.token;
            this.sessionTokenTimestamp = Date.now();

            return this.sessionToken;
        } catch (error) {
            console.error("Error getting session token:",
            error.response?.data || error.message);
        }
    }
}

```

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		89

```

        throw new Error("Failed to obtain session token");
    }
}

async ensureSessionToken() {
    if (!this.sessionToken || Date.now() -
this.sessionTokenTimestamp > this.tokenExpiryTime) {
        await this.getSessionToken();
    }
}

/**
 * Хешування файлу для підпису
 * @param {Buffer} fileBuffer - Буфер файлу
 * @returns {string} - Base64-закодований хеш
 */
hashFile(fileBuffer) {
    const hash = crypto.createHash('sha256');
    hash.update(fileBuffer);
    return hash.digest('base64');
}

/**
 * Генерація deeplink для підпису файлів
 * @param {string} branchId - Ідентифікатор відділення
 * @param {string} serviceName - Назва послуги
 * @param files
 * @returns {Object} - Об'єкт з deeplink, qrCode, та метаданими
 */
async generateDeeplink(branchId, serviceName, files) {
    try {
        await this.ensureSessionToken();

        let offerId = await this.createOffer(branchId,
serviceName);
        let requestId = crypto.randomBytes(16).toString("hex");

        // Хешуємо всі файли
        const hashedFiles = files.map(file => {
            const fileHash = this.hashFile(file.buffer);

            return {
                fileName: file.originalname,
                fileHash: fileHash
            };
        });

        const payload = {
            offerId,
            requestId,
            signAlgo: "ECDSA",
            data: {
                hashedFilesSigning: {
                    hashedFiles: hashedFiles
                }
            }
        }
    }
}

```

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				90
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }
    };

    console.log('Dynamic request payload:',
JSON.stringify(payload, null, 2));

    // Відправляємо запит
    const response = await axios.post(
`${this.diiaHost}/api/v2/acquirers/branch/${branchId}/offer-
request/dynamic`,
    payload,
    {
        headers: {
            Authorization: `Bearer ${this.sessionToken}`,
            "Content-Type": "application/json",
            Accept: "application/json",
        },
    }
);

    if (!response.data.deeplink) {
        throw new Error("No deeplink returned from Diia API");
    }

    const timestamp = new Date().toISOString();

    const url = new URL(response.data.deeplink);
    const otp = url.pathname.split('/').pop();
    return {
        success: true,
        deeplink: response.data.deeplink,
        otp,
        requestId,
        offerId,
        name: serviceName,
        timestamp,
        date: timestamp.split('T')[0]
    };

    } catch (error) {
        console.error("Error generating deeplink:",
error.response?.data || error.message);
        throw new Error(`Failed to generate deeplink:
${error.message}`);
    }
}

/**
 * @param {string} branchId - Ідентифікатор відділення
 * @param {string} name - Назва послуги
 * @returns {string} - Ідентифікатор створеного оффера
 */
async createOffer(branchId, name) {
    try {

```

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				91
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        await this.ensureSessionToken();

        if (!branchId || !name) {
            throw new Error("Missing required parameters: branchId,
name");
        }

        const payload = {
            name: name,
            scopes: {
                diiaId: ["hashedFilesSigning"]
            }
        };

        console.log('Create offer payload:',
JSON.stringify(payload, null, 2));

        const response = await axios.post(
`${this.diiaHost}/api/v1/acquirers/branch/${branchId}/offer`,
        payload,
        {
            headers: {
                Authorization: `Bearer ${this.sessionToken}`,
                "Content-Type": "application/json",
                Accept: "application/json",
            },
        }
    );

    if (!response.data._id) {
        throw new Error("No offer ID returned from Diia API");
    }

    return response.data._id;

    } catch (error) {
        console.error("Error creating offer:",
error.response?.data || error.message);
        throw new Error(`Failed to create offer:
${error.message}`);
    }
}

/**
 * Перевірка статусу підписання
 * @param {string} requestId - параметр, який передавався Партнером
в запиті на deeplink
 * @param {string} otp - частина від deeplink, яку партнер отримує
в запиті на підписання онлайн
 */
async checkStatus(requestId, otp) {
    try {
        const response = await axios.get(

```

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				92
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        `${this.diiaHost}/api/v1/acquirers/offer-
request/status`,
        {
            params: {
                otp,
                requestId,
            },
            headers: {
                Authorization: `Bearer ${this.sessionToken}`,
                Accept: "application/json",
            }
        }
    );

    const signedFilesList = [];

    const filePaths = await getSignedFilesFromDB(requestId);

    if (filePaths && filePaths.length > 0) {
        console.log(`[CheckStatus] Found ${filePaths.length}
files. Reading...`);

        for (const filePath of filePaths) {
            try {
                const fileBuffer = await
fs.readFile(filePath);

                const base64String =
fileBuffer.toString('base64');

                const dataUri = `data:application/pkcs7-
signature;base64,${base64String}`;

                signedFilesList.push({
                    fileName: path.basename(filePath),
                    data: dataUri
                });

            } catch (readErr) {
                console.error(`Error reading file
${filePath}:`, readErr.message);
            }
        }

        return {
            ...response.data,
            signed_files: signedFilesList
        };

    } catch (error) {
        console.error("Error check sign status:",
error.response?.data || error.message);
        throw new Error(`Failed to check sign status:
${error.message}`);
    }
}

```

		Коновал М.М.			ІІЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Вакалюк Т.А.				93
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }
  }
}

module.exports = new SignController();
app.post("/diia-user-file-upload", upload.none(), async (req, res) =>
{
  console.log("\n=====");
  ;
  console.log("NEW /diia-user-file-upload REQUEST");

  console.log("=====");

  try {
    console.log("Incoming headers:");
    console.log(JSON.stringify(req.headers, null, 2));

    console.log("Incoming body keys:", Object.keys(req.body));

    const requestId = req.headers["x-document-request-trace-id"];
    const diiaAction = req.headers["x-diia-id-action"];

    console.log("requestId:", requestId);
    console.log("diiaAction:", diiaAction);

    if (!requestId) {
      console.log("Missing requestId");
      return res.status(400).json({
        success: false,
        message: "Missing X-Document-Request-Trace-Id"
      });
    }

    console.log("Fetching requestData from DB...");
    // getRequestById шукає тільки в таблиці кваліфікаційних робіт
    (стара логіка)
    const requestData = await getRequestById(requestId);

    console.log("requestData found:", !!requestData);
    if (diiaAction !== "hashedFilesSigning") {
      console.log("Invalid action:", diiaAction);
      return res.status(400).json({
        success: false,
        message: "Invalid X-Diia-Id-Action"
      });
    }
  }
  const encodedDataB64 = req.body.encodeData;
  console.log("encodeData present:", !!encodedDataB64);

  if (!encodedDataB64) {
    console.log("encodeData missing");
    return res.status(400).json({
      success: false,
      message: "encodeData is required"
    });
  }
}

```

		Коновал М.М.			ІІЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Вакалюк Т.А.				94
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    });
}

console.log("Decoding Base64...");
const decodedJson = Buffer.from(encodedDataB64,
"base64").toString("utf-8");

let encodeData;
try {
    encodeData = JSON.parse(decodedJson);
} catch (jsonErr) {
    console.log("JSON decode error:", jsonErr);
    return res.status(400).json({
        success: false,
        message: "Invalid JSON after base64 decoding"
    });
}

console.log("Decoded JSON keys:", Object.keys(encodeData));

if (!encodeData.signedItems ||
!Array.isArray(encodeData.signedItems)) {
    console.log("signedItems missing or invalid");
    return res.status(400).json({
        success: false,
        message: "signedItems is missing or invalid"
    });
}

const signedItems = encodeData.signedItems;
console.log(`Received ${signedItems.length} signatures`);

// =====
// РОЗГАЛУЖЕННЯ ЛОГІКИ
// =====

if (requestData) {
    // -----
    // ВАРІАНТ 1: СТАРА ЛОГІКА (Кваліфікаційні роботи)
    // -----
    console.log(">>> MODE: LEGACY (Qualification Works)");

    for (const signed of signedItems) {
        const fileName = signed.name;
        const signatureB64 = signed.signature;

        console.log("\nProcessing signed item (Legacy):",
fileName);

        if (!signatureB64 || !fileName) {
            console.log("Invalid signed item:", signed);
            continue;
        }

        const result = await
decodeVerifyAndMakeCades(signatureB64);

```

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				95
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        const upgradedSignatureB64 = result.p7s;
        const signatureBuffer =
Buffer.from(upgradedSignatureB64, "base64");

        const savedFileName = `${fileName}.p7s`;

        // Старий шлях
        const savedFilePath = path.join(
            BASE_UPLOAD_DIRECTORY,
            String(requestData.qualification_work_id),
            savedFileName
        );

        console.log("Saving upgraded .p7s file:",
savedFilePath);

        await
ensureDirectoryExists(path.dirname(savedFilePath));
        await fs.writeFile(savedFilePath, signatureBuffer);

        console.log("Signature saved:", savedFilePath);

        const fieldName =
            fileName.replace(/\.([^\./]+)$/ , "") .replace(/\. /g,
            "_") + "_p7s";

        console.log("Mapped fieldName:", fieldName);

        await updateQualificationWorkFilePath(
            requestData.qualification_work_id,
            fieldName,
            savedFilePath
        );
    }

    console.log("Updating request_ids...");
    await updateRequestSuccess(requestId);

} else {
    // -----
    // ВАРІАНТ 2: НОВА ЛОГІКА (Універсальна / Фін. документи)
    // -----
    console.log(">>> MODE: NEW (Universal)");

    const savedFilePaths = [];

    for (const signed of signedItems) {
        const fileName = signed.name;
        const signatureB64 = signed.signature;

        console.log("\nProcessing signed item (Universal):",
fileName);

```

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Вакалюк Т.А.				96
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        if (!signatureB64 || !fileName) {
            console.log("Invalid signed item:", signed);
            continue;
        }

        const result = await
decodeVerifyAndMakeCades(signatureB64);

        const upgradedSignatureB64 = result.p7s;
        const signatureBuffer =
Buffer.from(upgradedSignatureB64, "base64");
        const savedFileName = `${fileName}.pdf.p7s`;

        const savedFilePath = path.join(
            BASE_UPLOAD_DIRECTORY_UNIVERSAL,
            String(requestId),
            savedFileName
        );

        console.log("Saving upgraded .p7s file:",
savedFilePath);

        await
ensureDirectoryExists(path.dirname(savedFilePath));
        await fs.writeFile(savedFilePath, signatureBuffer);

        savedFilePaths.push(savedFilePath);
        if (savedFilePaths.length > 0) {
            await processUniversalSignedFiles(requestId,
savedFilePaths);

            try {
                await updateRequestSuccess(requestId);
            } catch (e) {
                console.log("Universal logic:
updateRequestSuccess failed (maybe ID not in legacy table),
continuing...");
            }
            } else {
                console.warn("No valid files processed in Universal
mode");
            }
        }
    }

    console.log("Request successfully processed");
    return res.status(200).json({ success: true });

} catch (error) {
    console.log("Unexpected error:", error);
    return res.status(500).json({
        success: false,
        message: "Server error",
        error: error.message || error
    });
}

```

		Коновал М.М.			ІІЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Вакалюк Т.А.				97
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }
});
    makeCAdESSign(signature, data, signType) {
        let pThis = this;
        let euSign = pThis.euSign;

        return new Promise(function(resolve, reject) {
            pThis._initialize()
                .then(function() {
                    // Отримання кількості підписувачів для існуючого
підпису
                    let signsCount = euSign.CtxGetSignsCount(
                        pThis.context, signature);

                    // Встановлення типу підпису

                    euSign.SetRuntimeParameter(EU_SIGN_TYPE_PARAMETER, signType);

                    // Створення нового контейнеру з підписом
                    let newSignature = euSign.CreateEmptySign(data);

                    for (let i = 0; i < signsCount; i++) {
                        // Отримання інформації та сертифікату
підписувача
                        let signerInfo = euSign.CtxGetSignerInfo(
                            pThis.context, i, signature);
                        let signerCert = signerInfo.GetData();
                        // Отримання підписувача з підпису
                        let signer = euSign.GetSigner(i, signature);
                        // Отримання інформації про поточний тип
підпису
                        let curSignType = euSign.GetSignType(i,
signature);

                        // Додавання додаткових даних для перевірки
підпису
                        // лише у випадку якщо їх не має
                        if (signType > curSignType) {
                            signer =
euSign.AppendValidationDataToSigner(
                                signer, signerCert);
                        }

                        // Додавання підписувача до контейнеру з
новим підписом
                        newSignature = euSign.AppendSigner(
                            signer, signerCert, newSignature,
true);
                    }

                    resolve(newSignature);
                })
                .catch(function(e) {
                    reject(e);
                })
        });
    }

```

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ІЗ	Арк.
		Вакалюк Т.А.				98
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        });
    }
    public function actionPreview($id, $disciplineId = null):
Response|string
    {
        $pollQuery = Poll::find()
            ->alias('p')
            ->joinWith('semester')
            ->where(['p.id' => $id]);

        $pollModel = $pollQuery->one();
        if (!$pollModel) {
            throw new NotFoundException('Опитування не
знайдено.');
```

```

        }

        $educationId = $this->getMockPollStudent($pollModel);
        $poll = $pollQuery->asArray()->one();

        $isByDisciplines = (int)$poll['type'] ===
Poll::TYPE_BY_EDUCATIONAL_COMPONENTS;
        $isSingle = (int)$poll['display_type'] ===
Poll::DISPLAY_TYPE_SINGLE;
        $isMultiple = (int)$poll['display_type'] ===
Poll::DISPLAY_TYPE_MULTIPLE;

        [, $groupId] = $this->getEducationDetails($educationId);

        $student = EdeboStudent::find()->where(['education_id' =>
$educationId])->asArray()->one();

        $response = [
            'poll' => $poll,
            'disciplines' => null,
            'discipline' => null,
            'student' => $student,
        ];

        if ($isByDisciplines) {
            $disciplines = $this->getDisciplinesWithStatus($poll,
$educationId, $groupId);

            if (!$isMultiple) {
                $response['disciplines'] = $disciplines;
            } else {
                if (!$disciplineId) {
                    $response['disciplines'] = $disciplines;
                    $response['students'] = $student;

                    return $this->render('preview', $response);
                }

                $currentDiscipline = null;
                foreach ($disciplines as $d) {
                    if ((int)$d['id'] === (int)$disciplineId) {
```

		Коновал М.М.			ІІЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Вакалюк Т.А.				99
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        $currentDiscipline = $d;
        break;
    }
}

if (!$currentDiscipline) {
    throw new NotFoundException('Освітній
компонент не знайдено.');
```

```

    }

    if (!empty($currentDiscipline['isCompleted'])) {
        return $this->redirect(['preview', 'id' =>
$poll['id'], 'student' => $student['id'] ?? null]);
    }

    $response['discipline'] = $currentDiscipline;
}

$rawQuestions = PollFormQuestion::find()
    ->alias('pfq')
    ->joinWith([
        'pollFormQuestionAnswers pfqa',
        'pollFormQuestionAnswers.answer',
    ])
    ->where(['pfq.is_active' => 1, 'pfq.poll_form_id' =>
$poll['poll_form_id']])
    ->orderBy(['pfq.position' => SORT_ASC, 'pfqa.position' =>
SORT_ASC])
    ->asArray()
    ->all();

$processedQuestions = array_map(function ($rawQuestion) {
    $answers = array_map(fn($qa) => array_merge(
        $qa['answer'] ?? [],
        ['position' => $qa['position'] ?? 0],
    ), $rawQuestion['pollFormQuestionAnswers'] ?? []);

    ArrayHelper::multisort($answers, 'position', SORT_ASC);

    return [
        'id' => $rawQuestion['id'],
        'text' => $rawQuestion['text'] ?? $rawQuestion['name']
        ?? '',
        'type' => $rawQuestion['type'],
        'position' => $rawQuestion['position'],
        'answers' => $answers,
    ];
}, $rawQuestions);

if ($isSingle) {
    $chunkedQuestions = [];
    $currentSignature = null;

    foreach ($processedQuestions as $cleanQuestion) {
```

		Коновал М.М.			ІІЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Вакалюк Т.А.				100
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        $signature = implode('-',
ArrayHelper::getColumn($cleanQuestion['answers'], 'id', 0));

        if ($signature !== $currentSignature) {
            $currentSignature = $signature;
            $chunkedQuestions[] = [];
        }

$chunkedQuestions[array_key_last($chunkedQuestions)][] =
$cleanQuestion;
    }
    $response['questions'] = $chunkedQuestions;
} else {
    $response['questions'] = $processedQuestions;
}

    return $this->render('preview', $response);
}

    public static function getEducationalDisciplineByPoll(Poll $poll,
array $filters = []): array
    {
        $groupIds = self::getTargetGroupIds($poll, $filters);
        return empty($groupIds) ? [] : self::getDisciplines($groupIds,
$poll->semester_id);
    }

    public static function getDisciplines(array $groupIds, int
$semesterId): array
    {
        return Studyload::find()
            ->alias('s')
            ->select([
                'ded.id',
                'ded.name',
                'GROUP_CONCAT(DISTINCT st.full_name SEPARATOR ", ") as
teachers',
            ])
            ->innerJoin('studyload_distribution sd', 'sd.studyload_id
= s.id AND sd.staff_department_id IS NOT NULL')
            ->innerJoin('staff_department std', 'std.id =
sd.staff_department_id')
            ->innerJoin('staff st', 'st.id = std.staff_id')
            ->innerJoin('workplan_educational_component wec', 'wec.id
= s.workplan_educational_component_id')
            ->innerJoin('dictionary_educational_discipline ded',
'ded.id = wec.educational_component_id')
            ->where(['s.group_id' => $groupIds, 's.semester_id' =>
$semesterId])
            ->groupBy(['ded.id', 'ded.name'])
            ->asArray()
            ->all();
    }

    public static function getTargetGroupIds(Poll $poll, array $filters
= []): array

```

		Коновал М.М.			ІІЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Вакалюк Т.А.				101
Змн.	Арк.	№ докум.	Підпис	Дата		

```

{
    $cacheKey = $poll->id . md5(serialize($filters));
    if (isset(self::$_groupIdsCache[$cacheKey])) {
        return self::$_groupIdsCache[$cacheKey];
    }

    $query = AcademicGroup::find()->alias('ag')->select('ag.id');

    if (!empty($poll->pollStudents)) {
        $studentIds = array_column($poll->pollStudents,
'student_id');
        $query->innerJoin('student s', 's.academic_group_id =
ag.id')
            ->andWhere(['s.education_id' => $studentIds]);
    } elseif (!empty($poll->pollGroups)) {
        $query->andWhere(['ag.id' => array_column($poll->
pollGroups, 'group_id')]);
    } elseif (!empty($poll->pollStudyPrograms)) {
        $query->andWhere(['ag.study_program_id' =>
array_column($poll->pollStudyPrograms, 'study_program_id')]);
    } else {
        if ($poll->faculty_id) {
            $facultyEdeboId = UniversityStructure::find()
                ->select('edebo_university_structure_id')
                ->where(['id' => $poll->faculty_id])
                ->scalar();
            $query->andWhere(['ag.faculty_id' => $facultyEdeboId
?: null]);
        }
        if ($poll->study_form_id) $query->
        >andWhere(['ag.study_form_id' => $poll->study_form_id]);
        if ($poll->education_level_id) $query->
        >andWhere(['ag.degree_id' => $poll->education_level_id]);
    }

    $query->andFilterWhere([
        'ag.degree_id' => $filters['education_level_id'] ?? null,
        'ag.study_form_id' => $filters['study_form_id'] ?? null,
        'ag.faculty_id' => $filters['faculty_id'] ?? null,
        'ag.study_program_id' => $filters['study_program_id'] ??
null,
        'ag.course' => $filters['course'] ?? null,
        'ag.id' => $filters['group_id'] ?? null,
    ]);

    return self::$_groupIdsCache[$cacheKey] = $query->column();
}

public static function getTargetStudentIds(Poll $poll, array
$filter = []): array
{
    $groupIds = self::getTargetGroupIds($poll, $filters);
    if (empty($groupIds)) return [];
}

```

```

        $query = Student::find()->select('education_id')-
>where(['academic_group_id' => $groupIds]);
        if (!empty($poll->pollStudents)) {
            $query->andWhere(['education_id' => array_column($poll-
>pollStudents, 'student_id')]);
        }

        return $query->column();
    }
$(document).ready(function() {
    let currentQuestionIndex = 0;

    $(document).on('click', '#nextBtn', function() {
        const $currentQuestion = $('.question-
container').eq(currentQuestionIndex);
        const $nextQuestion = $currentQuestion.next();

        clearError();

        if ($nextQuestion.length && validateAnswer()) {
            $currentQuestion.removeClass('active');
            $nextQuestion.addClass('active');
            $(this).prop('disabled', !$nextQuestion.next().length);
            $('#prevBtn').prop('disabled', false);
            currentQuestionIndex++;
            showFinishButton();
            updateProgress();
        }
    })

    $(document).on('click', '#prevBtn', function() {
        const $currentQuestion = $('.question-
container').eq(currentQuestionIndex);
        const $prevQuestion = $currentQuestion.prev();

        clearError();

        if ($prevQuestion.length) {
            $currentQuestion.removeClass('active');
            $prevQuestion.addClass('active');
            $(this).prop('disabled', !$prevQuestion.prev().length);
            $('#nextBtn').prop('disabled', false);
            currentQuestionIndex--;
            showFinishButton();
            updateProgress();
        }
    })

    $(document).on('input', '.form-textarea', function() {
        const $textarea = $(this);
        let value = $textarea.val();
        const maxLength = 1000;
        const $symbolCounter = $textarea.siblings('.symbol-counter');

        if (value.length > maxLength) {

```

		Коновал М.М.			ІІЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Вакалюк Т.А.				103
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        value = value.slice(0, maxLength);
        $textarea.val(value);
    }

    $symbolCounter.text(`${value.length} / ${maxLength} символів`);
});

function validateAnswer() {
    const $activeQuestion = $('.question-container.active');

    if ($activeQuestion.has('.option-list').length &&
    !$activeQuestion.has('input:checked').length) {
        $activeQuestion.find('.error-message').text('Будь ласка, оберіть
    варіант відповіді');
        return false;
    }

    return true;
}

function clearError() {
    $('.question-container .error-message').text('');
}

function showFinishButton() {
    const $nextBtn = $('#nextBtn');
    const $finishBtn = $('#finishBtn');

    const questionCount = $('.question-container').length;

    if (currentQuestionIndex === questionCount - 1) {
        $nextBtn.hide();
        $finishBtn.show();
    } else {
        $nextBtn.show();
        $finishBtn.hide();
    }
}

function updateProgress() {
    const questionCount = $('.question-container').length;
    const progress = (currentQuestionIndex / questionCount *
    100).toFixed(1);

    $('#progressText').text(`Запитання ${currentQuestionIndex + 1} з
    ${questionCount}`);
    $('#progressPercent').text(`${progress}%`);
    $('#progressBar').css('width', `${progress}%`);
}
})

```

		Коновал М.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А.				104
Змн.	Арк.	№ докум.	Підпис	Дата		