

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:

**«Інформаційна система онлайн менеджменту
освітніх процесів "Digital University": Сесія»**

Виконав студент групи ВТ-21-1

ДЖУС Ігор Олександрович

Керівник роботи:

МОРОЗОВ Андрій Васильович

Рецензент:

ФАНТ Микола Олександрович

Житомир – 2025

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

«ЗАТВЕРДЖУЮ»

Зав. каф. інженерії
програмного забезпечення
Тетяна ВАКАЛЮК
«26» лютого 2025 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **ДЖУС Ігор Олександрович**

Керівник роботи: **МОРОЗОВ Андрій Васильович**

Тема роботи: **«Інформаційна система онлайн менеджменту освітніх процесів "Digital University": Сесія»**

затверджена Наказом закладу вищої освіти від **«14» лютого 2025 р., №61/с**
Вихідні дані для роботи: **розробити систему управління навчальними процесами в університеті, що покращить роботу його адміністрації та надасть функціонал для автоматизації рутинних завдань, зокрема розробити модулі для проведення сесії та ведення відомостей обліку успішності в електронному вигляді.**

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

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Морозов А.В.	27.02.2025р.	27.02.2025р.
2	Морозов А.В.	24.04.2025р.	24.04.2025р.
3	Морозов А.В.	02.05.2025р.	02.05.2025р.

РЕФЕРАТ

Кваліфікаційна робота складається з програмного комплексу для інформаційної системи онлайн-менеджменту освітніх процесів «Digital University», а конкретно – модуля «Сесія» та пояснювальної записки. Пояснювальна записка до випускної роботи містить 71 сторінок, 27 ілюстрацій та 1 таблицю.

Головною метою є створення модулю інформаційної системи, який дозволить цифровізувати увесь етап проведення залікових сесій у закладах вищої освіти та впровадити зручний інтерфейс для використання користувачами різних вікових категорій та посад.

У ході роботи були визначені основні завдання для розробки модулю, проведений аналіз аналогів інформаційної системи та обрано технології, що дозволять реалізувати низку вимог до програмного забезпечення.

КЛЮЧОВІ СЛОВА: BACKEND, ІНФОРМАЦІЙНА СИСТЕМА, СЕСІЯ, ВІДОМОСТІ, ВИКЛАДАЧІ, РНР, YII2.

					ІПЗ.КР.Б – 121 – 25 – ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	Інформаційна система онлайн менеджменту освітніх процесів "Digital University": Сесія Пояснювальна записка	Лім.	Арк.	Аркушів
Розроб.		Джус І.О.					3	71
Керівник		Морозов А. В.				Житомирська політехніка, група ВТ-21-1		
Н. контр.								
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", specifically the "Session" module and an explanatory note. The explanatory note to the graduation work contains 71 pages, 27 illustrations and 1 table.

The main goal is to create an information system module that will allow digitalizing the entire stage of conducting credit sessions in higher education institutions and implementing a convenient interface for use by users of different age categories and positions.

During the work, the main tasks for developing the module were identified, an analysis of analogues of the information system was conducted, and technologies were selected that will allow implementing a number of requirements for software.

KEYWORDS: BACKEND, INFORMATION SYSTEM, SESSION, NEWS, TEACHERS, PHP, Yii2.

		Джус І.О.			ІІЗ.КР.Б – 121 – 25 – ІІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

ЗМІСТ

РЕФЕРАТ	3
ABSTRACT	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	6
ВСТУП.....	7
РОЗДІЛ 1. АНАЛІЗ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ.....	8
1.1 Постановка задачі.....	8
1.2 Аналіз аналогів програмного продукту	9
1.3 Вибір архітектури системи.....	13
1.4 Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення	15
Висновки до першого розділу.....	16
РОЗДІЛ 2. ПРОЕКТУВАННЯ СИСТЕМИ ОНЛАЙН-МЕНЕДЖМЕНТУ ОСВІТНІХ ПРОЦЕСІВ.....	17
2.1 Визначення варіантів використання та структури системи	17
2.2 Розробка бази даних системи.....	22
2.3 Постановка задачі.....	25
2.4 Реалізація функціоналу системи управління навчальним процесом	26
Висновки до другого розділу	41
РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З СИСТЕМОЮ ОНЛАЙН-МЕНЕДЖМЕНТУ ОСВІТНІХ ПРОЦЕСІВ	43
3.1 Порядок встановлення та налаштування параметрів додатку.....	43
3.2 Структура інтерфейсу додатку	44
Висновки до третього розділу	57
ВИСНОВКИ.....	58
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	59
ДОДАТКИ.....	62
Додаток А	63

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД – база даних

UI – інтерфейс користувача

ЦП – центральний процесор

ЕЦП – електронний цифровий підпис

ЄДЕБО – Єдина державна електронна база з питань освіти

ОЗП – оперативна пам'ять

		Джус І.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

ВСТУП

Освіта була, є та залишиться однією з головних складових існування людства, без якого неможливий подальший розвиток технологій, пошук рішень для глобальних проблем, однак споконвіку освіта мала нерівного суперника – час. З кожним поколінням світ підносить ще більше викликів[1], залишаючи менше часу на отримання якісної освіти та відповідних знань, бо частиною цього є неосяжна побудова коректних освітніх процесів, формування матеріалу, оцінювання, яке в сьогоденні здійснюється не найзручнішим способом. Якщо не почати зміни зараз, то з часом бажання як викладачів, так і студентів отримувати освіту та безліч років зіштовхуватися з бюрократичними проблемами зовсім зникне. Саме тому об'єктом розгляду стає інформаційна система онлайн-менеджменту освітніх процесів.

Метою даної роботи є розробка модуля «Сесія». Встановлена мета обумовлює конкретні завдання, а саме аналіз даного освітнього процесу та уніфікація етапів задля універсального використання в інформаційній системі (надалі ІС); побудова оптимальної архітектури; обґрунтування засобів реалізації системи; розробка інформаційного та математичного забезпечення системи.

Об'єктом дослідження інформаційні системи онлайн-менеджменту освітніх процесів, що включають залікову сесію.

Предметом дослідження є використання ключових механік та підходів, що практикуються у цифровізації освітніх процесів.

		Джус І.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		7

РОЗДІЛ 1. АНАЛІЗ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

1.1 Постановка задачі

Модуль «Сесія» представляє собою частину веб платформи [2] зі зручним для користувача інтерфейсом та серверною складовою, що складається з алгоритмів обрахунку оцінок, отриманих студентами за сесію, контроль над ними зі сторони відповідальних користувачів (викладачі, декани тощо), створення відомостей, зведення відомостей, відображення результатів залікової сесії здобувачам у кабінетах. Модуль включає в собі підмодулі статистики для порівняння залікових сесій з різних навчальних семестрів, діаграми та генерування звітності у вигляді таблиць.

Необхідна реалізація таких функцій, як зручний інтерфейс, враховуючи специфіку працівників закладів освіти. Інтерфейс має бути простим та зрозумілим для усіх вікових категорій, а різні доступні дії повинні супроводжуватися відповідними повідомленнями про виконання.

Також важливою складовою є ядро модуля, що дозволяє створювати таблиці та моделі для зберігання даних, реалізовувати сервіси для обрахунку та пошуку. Невідмінною частиною є розробка контролерів для обробки сторінок та дій, а також конфігурація та налаштування сталих показників модулю (мінімальна/максимальна оцінка, терміни сесії). Ще одним етапом є реалізація кабінету здобувача освіти: інтерфейс, групування даних по навчальним семестрам, безпека та конфіденційність даних.

Наступною обов'язковою частиною є авторизація, побудована на загальних принципах безпеки з врахуванням особливості безпекових заходів в університеті.

Одним з ключових функціоналів буде генерування документації та звітності з використанням .docx шаблонів зі змінними для підстановки відповідних даних, конвертація у .pdf формат. Також генерування .xlsx файлів зі зведеними

		Джус І.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		8

відомостями для друку та оформлення статистики. Перевагою буде постійне динамічне генерування, що забезпечує оновленість даних.

1.2 Аналіз аналогів програмного продукту

Сьогодні можна знайти доволі багато програмних продуктів, що намагаються покрити певні частини освітнього процесу, але головним їхнім недоліком є подрібненість та залежність від самостійних сервісів. Найбільш схожі продукти перераховані нижче:

Інформаційна система управління освітнім процесом в Кам'янець-Подільському національному університеті імені Івана Огієнка [3]

Подана ІС є гарним прикладом централізації управління освітніми процесами та надає можливості цифровізувати адміністративні процеси та роботу із заліковими відомостями. Однак, що стає зрозуміло з відкритих джерел, ІС не надає всього спектру можливостей роботи з відомостями, а саме зведення відомостей, додаткові відомості, кабінет студента з оцінками та підписання відомостей. Також ІС не передбачає налаштувань області оцінок та іншого.

Посилання на ІС: <https://meridian.kpnu.edu.ua/2022/11/15/informatsiina-systema-upravlinnia-osvitnim-protsesom-v-universyteti-ohiiienka-stan-i-shliakhy-vdoskonalennia>

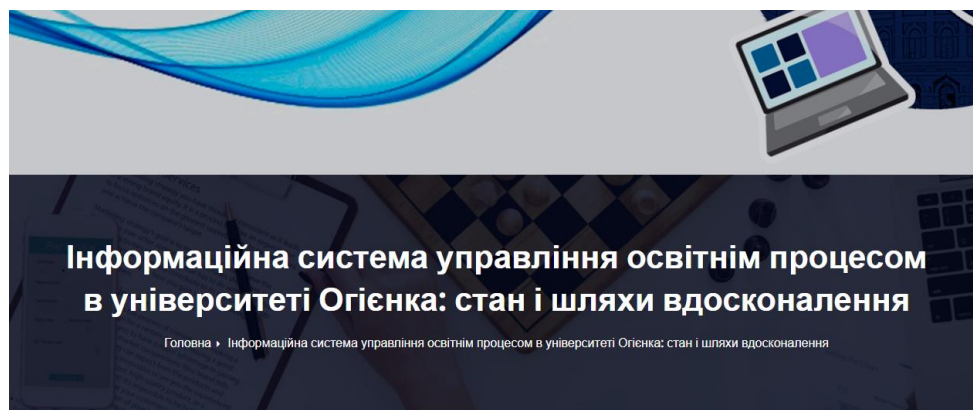


Рис. 1.1. Інформаційна система управління освітнім процесом в Кам'янець-Подільському національному університеті імені Івана Огієнка

		Джус І.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

До недоліків даної системи можна віднести відсутність зведених відомостей, а також відсутність можливості підписати відомість електронним підписом. Не передбачена можливість переглянути оцінки здобувачу.

iZETA [4]

Дана ІС є ще одним підвидом системи, що цифровізує освітні процеси, однак у ході дослідження було визначено, що програмний продукт більше адаптований до внутрішнього керування документами, та не покриває такі етапи як сесія у зв'язці з безпосередніми викладачами та здобувачами освіти.

Посилання на ІС: <http://www.izeta.puet.edu.ua/>

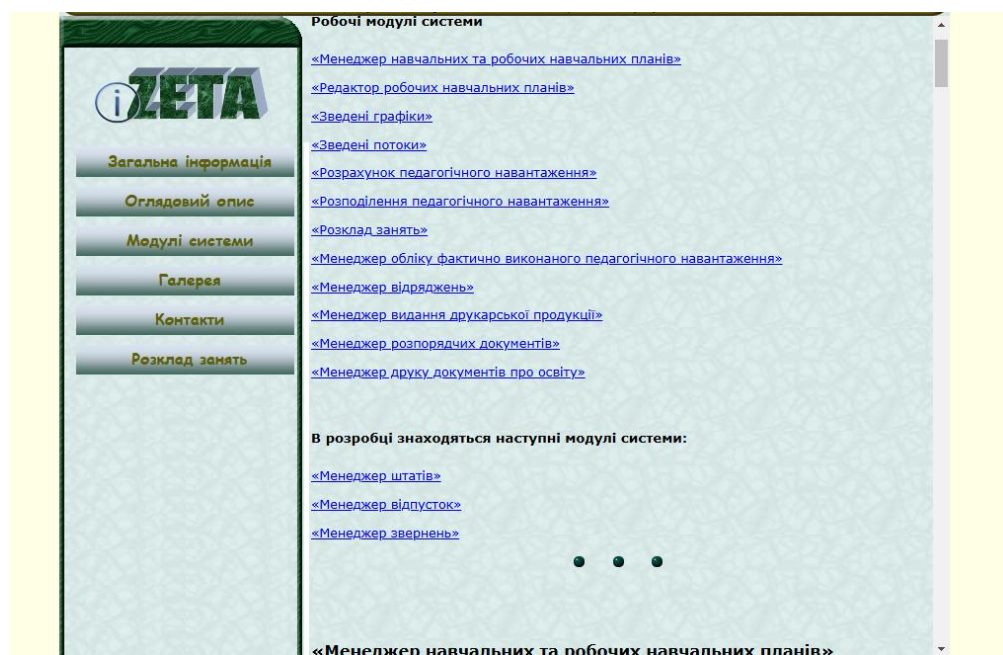


Рис. 1.2.1. iZETA

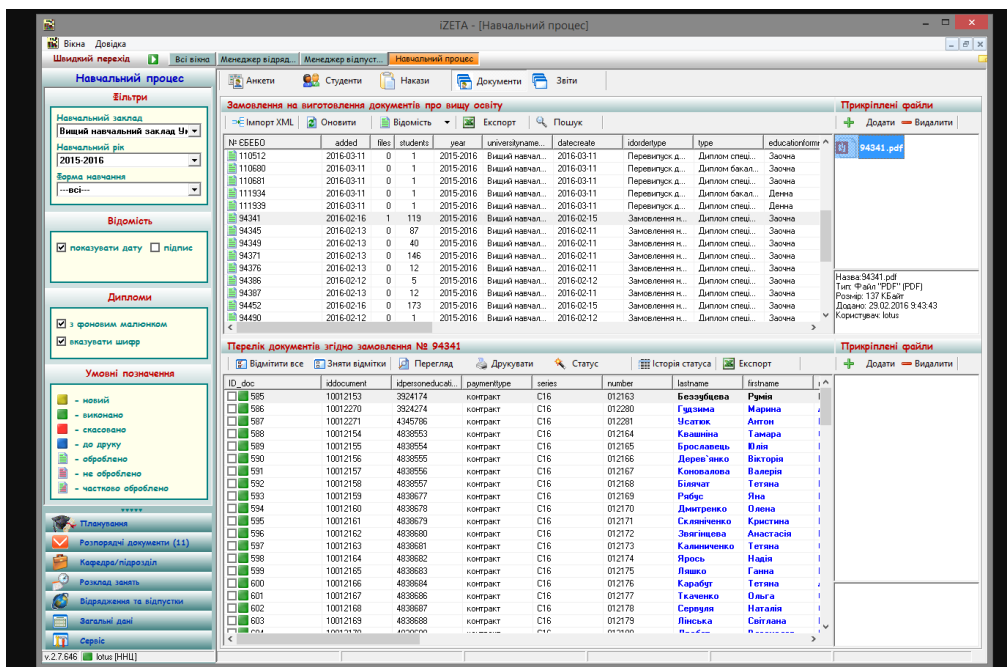


Рис. 1.2.2. iZETA

До переваг даної системи можна віднести великий спектр менеджерів для роботи з персоналом та наявність редакторів для побудови робочих планів та іншого.

До недоліків відносяться застарілий інтерфейс та відсутність інформації про сумісність з новими операційними системами. Не масштабоване створення документів та відсутність функціоналу для проведення залікових сесій.

Комплексна інформаційна система автоматизації управління навчальним процесом [5]

ІС є однією з небагатьох, функціонал яких передбачає проведення залікових сесій на базі вищих навчальних закладів та пропонує широкий спектр побічних можливостей, що супроводжують використання модулю, однак, враховуючи лише описану документацію (факт наявної інтегрованої системи не підтверджено), модуль залікових сесій обмежує користувача великою кількістю ручної роботи, у той час, поки розроблюваний модуль передбачатиме фактично повну автоматизацію формування відомостей та динаміку в залежності від специфіки груп та закладів освіти.

Посилання на документацію ІС:

<https://ena.lpnu.ua:8443/server/api/core/bitstreams/5eebd9d5-1abc-4637-b3b6-92de8b5e702e/content>

Рис. 1.3. Комплексна інформаційна система автоматизації управління навчальним процесом

До переваг даної системи можна віднести широкі можливості щодо генерації документів та нахил до ведення звітності.

Що ж до недоліків, то виявлено обмежений функціонал модулю сесії та відсутність реальних кейсів імплементації та роботи ІС

		Джус І.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця. 1.1

Характеристика	Інформаційна система управління освітнім процесом в Кам'янець-Подільському національному університеті імені Івана Огієнка	iZETA	Комплексна інформаційна система автоматизації управління навчальним процесом
Зручний інтерфейс	+	-	-
Мультиплатформність	+	-	-
Безпека	+	+	-
Робота з відомостями	+	-	+
Автоматизація роботи з відомостями	-	-	-
Генерування звітності	+	-	+
Кабінет студента	-	-	-

Після розгляду трьох аналогів ІС можна зробити висновок, що кожна з описаних покриває функціоналом певну частину великого модуля «Сесія», однак враховуючи досвід вищих навчальних закладів, досліджуваний модуль неможливо ділити на окремі частини, тому необхідно розробити програмний продукт, що забезпечить зручність використання та ефективний інструментарій для проведення сесії. Усі переваги та недоліки будуть враховані під час розробки.

1.3 Вибір архітектури системи

У сучасному світі активного розвитку ІТ-сфери виникла потреба в чіткому плануванні структури програмного забезпечення [6]. Один з найпоширеніших підходів у розробці програмного забезпечення — це монолітна архітектура, яка передбачає об'єднання всіх частин програми (інтерфейс користувача, логіка, база даних) в один єдиний блок.

Такий підхід є традиційним і особливо зручним на етапі запуску проєкту або для невеликих команд, адже дозволяє швидко розробити, протестувати й

		Джус І.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

розгорнути додаток. Простіше налаштовується моніторинг, налагодження та взаємодія між компонентами. Крім того, монолітна структура забезпечує високу продуктивність, оскільки всі елементи працюють у спільному середовищі без затримок на мережеву взаємодію.

Однак у масштабних системах моноліт втрачає ефективність: із зростанням обсягу коду він стає важким для підтримки, розширення та тестування. Будь-які зміни можуть мати непередбачувані наслідки, особливо коли команда велика. У порівнянні з мікросервісною архітектурою, яка дозволяє розділяти функціонал на незалежні сервіси, моноліт менш гнучкий і складніший для масштабування. Водночас, мікросервіси вимагають складнішої інфраструктури та кращої координації між модулями. Отже, вибір архітектури залежить від складності проєкту, доступних ресурсів та довгострокових планів щодо розвитку системи.

Мікросервісна архітектура [7] — це підхід до розробки програмного забезпечення, за якого система складається з невеликих незалежних сервісів, кожен з яких виконує чітко визначену функцію. Ці сервіси взаємодіють між собою через API або повідомлення, працюють автономно та можуть розроблятися і оновлюватися окремо. Такий підхід особливо зручний для великих проєктів або команд, де окремі модулі можуть створюватися і підтримуватися незалежно. Це значно підвищує масштабованість та гнучкість системи, а також полегшує впровадження нових функцій без ризику порушити роботу всієї програми.

Проте мікросервіси мають і свої недоліки. По-перше, для їх коректної роботи потрібна складна інфраструктура — з налагодженим моніторингом, балансуванням навантаження та безпекою взаємодії. По-друге, розподілена структура ускладнює тестування і може призводити до затримок у взаємодії між модулями. Мікросервіси вимагають більше ресурсів і технічної підготовки, однак у довгостроковій перспективі забезпечують кращу адаптивність і масштабованість. Обираючи між цими двома підходами, важливо враховувати

		Джус І.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				14
Змн.	Арк.	№ докум.	Підпис	Дата		

розмір команди, темпи розвитку продукту та очікуваний обсяг функціональності.

Ми обрали монолітний підхід для нашої системи, оскільки наразі не планується розширення на мобільні пристрої або значна масштабованість, яка б вимагала використання мікросервісів. Оскільки наш продукт орієнтований на роботу в межах однієї платформи та з однією інтегрованою кодовою базою, монолітна архітектура є оптимальним рішенням для забезпечення швидкої розробки та спрощеного розгортання. Такий підхід дозволяє зменшити складність інфраструктури та забезпечити високу продуктивність при мінімальних витратах на підтримку.

1.4 Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення

У процесі вибору технологій для розробки нашої системи ми розглянули безліч варіантів, що включали різні мови програмування та фреймворки. Кожен з них мав свої переваги та можливості, але після ретельного аналізу, зважаючи на специфіку проекту, ми зупинили свій вибір на PHP [8] як мові програмування та Yii2 як фреймворку для розробки. Цей вибір став результатом поєднання таких факторів, як швидкість розробки, зручність масштабування, висока продуктивність та безпека.

PHP має багаторічну історію та є однією з найбільш поширених мов програмування для створення веб-додатків. Його популярність пояснюється величезною кількістю готових рішень, бібліотек та фреймворків, що дозволяє ефективно зменшити час на розробку без шкоди для якості. Враховуючи потребу в ефективній роботі з серверною логікою та базами даних, PHP є ідеальним вибором для нашої системи. Що стосується фреймворку, Yii2[9] надає потужний набір інструментів для швидкої розробки масштабованих і безпечних веб-додатків. Завдяки вбудованим механізмам кешування, підтримці сучасних стандартів безпеки та зручному синтаксису, Yii2 забезпечує високу

		Джус І.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				15
Змн.	Арк.	№ докум.	Підпис	Дата		

продуктивність та гнучкість. Він ідеально підходить для створення CRUD-подібних бізнес-логік, що робить його найкращим вибором для нашої системи.

Висновки до першого розділу

У першому розділі роботи було проведено аналіз вимог до системи, розглянуто актуальні підходи до реалізації серверних рішень та здійснено порівняння доступних мов програмування і фреймворків.

З урахуванням потреб проєкту було ухвалено рішення використовувати PHP як основну мову програмування, а Yii2 — як фреймворк для побудови веб-системи. Такий вибір обґрунтовано високою швидкістю розробки, зрозумілою архітектурою, широким інструментарієм для безпечної та ефективної роботи з базами даних, а також добрим балансом між продуктивністю і простотою підтримки.

Під час технічного проєктування було також розглянуто варіанти побудови архітектури системи. Проаналізувавши плюси й мінуси мікросервісного підходу, було прийнято рішення реалізувати систему в межах монолітної архітектури. Це дозволило спростити логіку взаємодії компонентів, прискорити розробку та знизити витрати на розгортання й підтримку.

У результаті сформовано чітке технічне рішення, яке базується на перевірених технологіях, відповідає вимогам до безпеки, продуктивності та масштабованості, а також демонструє оптимальне співвідношення витрат часу, ресурсів і очікуваної функціональності.

		Джус І.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				16
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2. ПРОЕКТУВАННЯ СИСТЕМИ ОНЛАЙН-МЕНЕДЖМЕНТУ ОСВІТНІХ ПРОЦЕСІВ

2.1 Визначення варіантів використання та структури системи

Для забезпечення стабільного та безперервного виконання ключових функцій програмного забезпечення, а також дотримання встановлених стандартів і технічних специфікацій, сформульовано високорівневі вимоги[10] до розроблюваної системи. Ці вимоги охоплюють як функціональні, так і нефункціональні аспекти роботи програми, включаючи надійність, продуктивність, безпеку, масштабованість та сумісність із різними програмно-апаратними середовищами.

Бізнес вимоги

Основними цілями є створення проєкту з метою цифровізації та оптимізації управління освітніми процесами в закладах вищої освіти. Система має забезпечити прозоре оцінювання, автоматизацію зведення відомостей та зручний доступ здобувачів освіти до результатів сесії.

Представленням проєкту є інформаційна система, що буде реалізована у вигляді веб-застосунку з архітектурою "клієнт-сервер" [11]. Серверна частина буде побудована з використанням фреймворку PHP Yii2, що дозволить створити надійну, масштабовану та легко підтримувану систему.

Вимоги адміністрації

1. Можливість переглядати матеріал відповідно до ролі та прав доступу (здобувач, викладач, декан, адміністратор);
2. Зручний перегляд результатів сесії з групуванням по предметах і семестрах;
3. Можливість пошуку та сортування за будь-якими параметрами;
4. Контроль оцінок з боку викладача та керівників факультету;
5. Автоматичне формування відомостей та їх перевірка;
6. Генерація офіційної документації у форматах .docx, .pdf та .xlsx;

		Джус І.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				17
Змн.	Арк.	№ докум.	Підпис	Дата		

7. Доступність усіх функцій з мобільних пристроїв без втрати зручності інтерфейсу;
8. Повна звітність за сесією — формування аналітики успішності, кількості зданих предметів, динаміки оцінювання.

Вимоги студентів

1. Налаштування власного профілю
2. Можливість переглядати власні отримані оцінки за всі семестри;
3. Можливість погодження оцінки або подання заявки на участь у екзамені.

Вимоги власників

1. Гнучка система налаштувань мінімальних прохідних балів, термінів сесії, статусів оцінювання;
2. Контроль доступів користувачів на основі ролей та прав;
3. Логування дій користувачів для підвищення безпеки.

Характеристика об'єкта комп'ютеризації

Користувачами системи будуть адміністрація та працівники університету, яким буде доступна система менеджменту освітніх процесів, а також студенти, що матимуть можливість авторизуватися у кабінеті та переглядати власну успішність. Інтерфейс повинен бути інтуїтивно зрозумілим[12] для людей різного віку. Основний акцент зроблено на простоту, наочність та автоматизацію всіх рутинних дій, пов'язаних з проведенням сесії. Доступ до всіх функцій здійснюється через браузер[13] без необхідності встановлення додаткових програм.

Функціональні вимоги

1. Авторизація та розмежування прав доступу на основі ролей;
2. Завантаження, редагування та підтвердження оцінок;
3. Формування та перегляд відомостей та зведених документів;
4. Автоматичне обчислення загальних результатів за предметами, модулями, семестрами;
5. Експорт оцінок у документи відповідного формату;

		Джус І.О.			ПІЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				18
Змн.	Арк.	№ докум.	Підпис	Дата		

6. Можливість перегляду результатів у кабінетах здобувачів;
7. Надсилання сповіщень користувачам щодо дій, які потрібно виконати (заповнити, підписати тощо).

Нефункціональні вимоги

1. Сприйняття

- Час навчання користувача — до 60 хвилин;
- Час відгуку для типових дій — до 2 секунд, для генерації звітності — до 10 секунд;
- Пояснення та підказки до кожної дії в інтерфейсі.

2. Надійність

- Доступність системи — не менше 95% часу на добу;
- Середній час безвідмовної роботи — не менше 20 робочих днів;
- Максимальна норма помилок — 1 на 20 тисяч рядків коду.

3. Продуктивність

- Робота на пристроях зі слабкими технічними характеристиками;
- Адаптивний інтерфейс для екранів різного розміру;
- Максимально можливий вивід інформації на екрані.

4. Можливість експлуатації

- Система повинна бути модульною, з можливістю додавання нових функцій у майбутньому;
- Можливість легкої інтеграції з іншими внутрішніми сервісами ЗВО;
- Забезпечення резервного копіювання бази даних та звітної інформації.

Після здійснення повноцінного аналізу вимог до системи та виявлення ключових акторів [14] було розроблено діаграму акторів системи (рис.2.1):

		Джус І.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		19

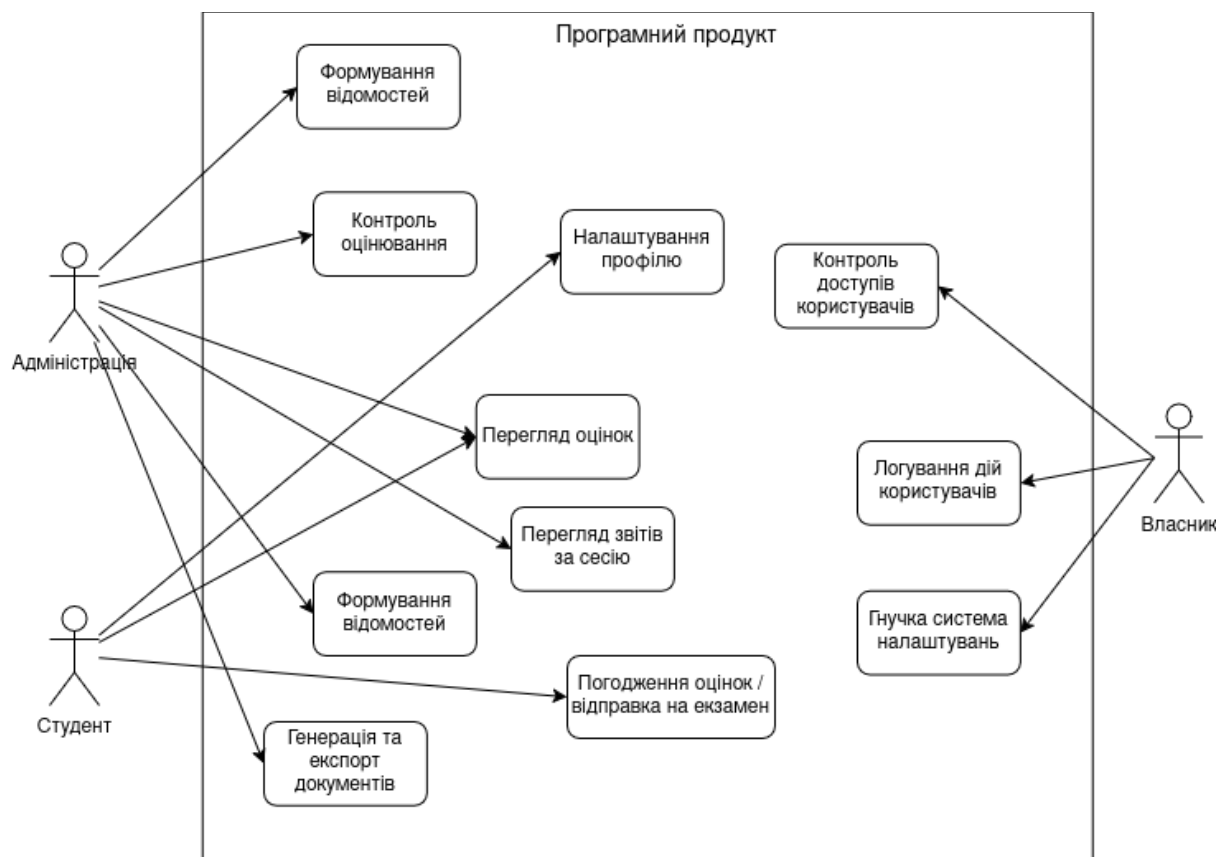


Рис. 2.1. Діаграма використання системи

У результаті аналізу функціональних вимог було створено діаграми активності, що відображають ймовірні сценарії розвитку подій, що виникають при використанні системи. Розглянемо сценарій використання підмодуля “Відомості” (рис. 2.2):

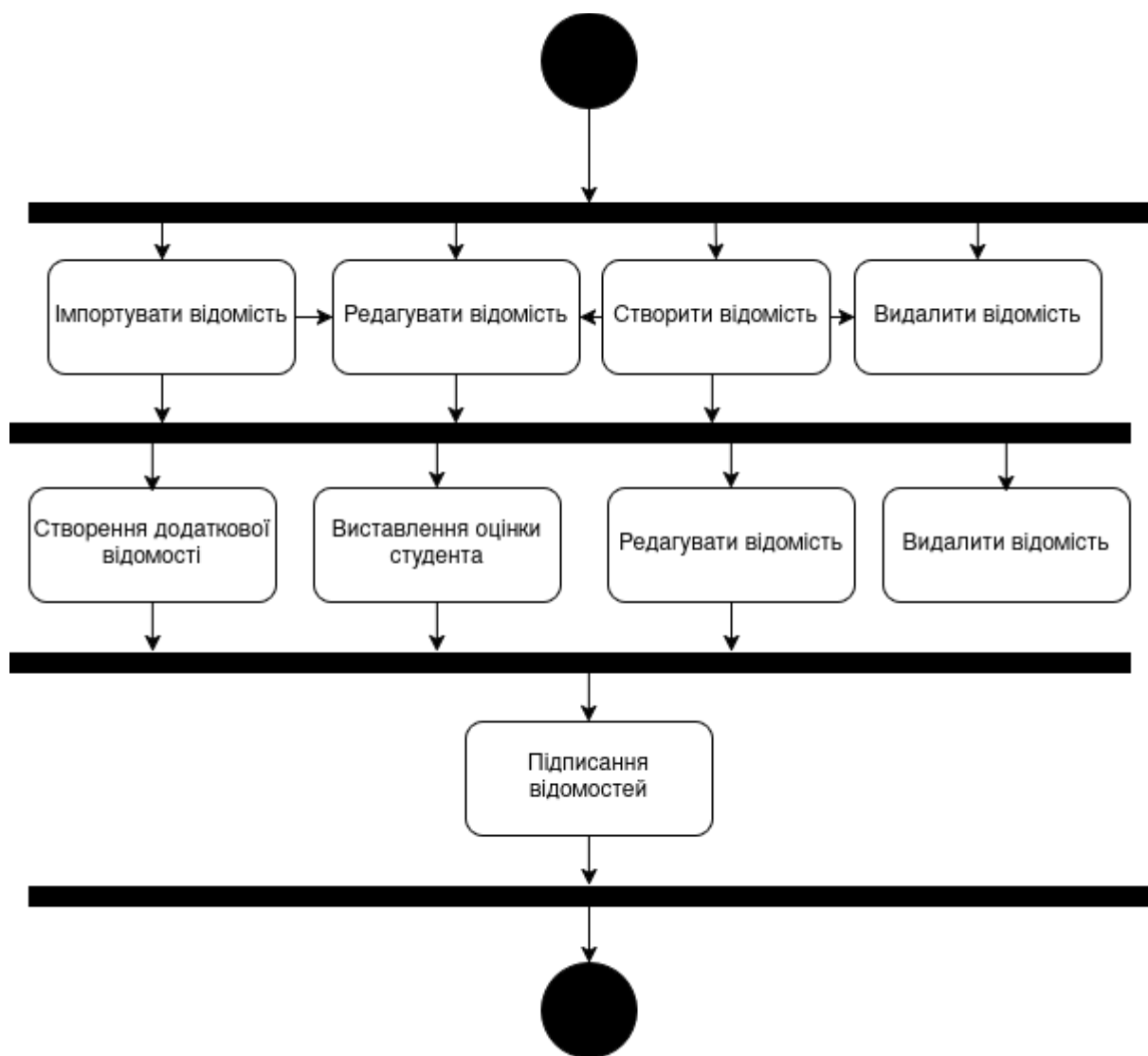


Рис. 2.2. Діаграма активності

Користувач при вході на індексову сторінку може бачити список усіх доступних до перегляду відомостей [15]. Доступ визначається залежно від його підрозділу, до якого він приналежний - таким чином, працівник може отримувати інформацію лише про відомості, які відносяться до його університетської структури. Працівник може виконати імпорт відмостей у .csv форматі, створити нову вручну або ж перейти до редагування/видалення існуючої.

При перегляді відмості відкриватиметься таблиця з усіма оцінками студентів даної групи. Користувач отримає можливість виставляти бали кожному студенту відповідно до встановленої шкали в налаштуваннях системи університету, якщо термін виставлення ще не минув. Ввід автоматично

валідуватиметься і конвертуватиметься у національну шкалу та ЄКТС[16]. На базі основної відомості, що переглядається, можна створити додаткову, таким чином даючи студентам із боргами шанс на виправлення оцінки.

Всі створені відомості користувач може підписати за допомогою ЕЦП. Для повноцінного підписання документу необхідно зібрати підписи від викладача, завідувача та декана, і в залежності від ролі поточного користувача у системі накладатиметься відповідний підпис. При внесенні будь-яких змін до завіреної підписом відомості цю процедуру доведеться повторити знову з метою попередження небажаних наслідків.

2.2 Розробка бази даних системи

У процесі розробки інформаційної системи було обрано систему управління базами даних MySQL [17], а для зручності керування — вебінтерфейс phpMyAdmin [18]. Такий вибір не є випадковим, адже MySQL зарекомендувала себе як одна з найнадійніших та найпопулярніших СКБД у сфері веброзробки. Вона використовується у великій кількості успішних проєктів по всьому світу, що свідчить про її стабільність, широке поширення та активну підтримку з боку як спільноти, так і розробників.

Однією з головних переваг MySQL є її висока продуктивність і здатність до масштабування. Це означає, що система може ефективно працювати як з невеликими обсягами даних, так і з великими навантаженнями, що є надзвичайно важливим для сучасних вебдодатків. У нашому випадку це дозволяє обробляти значну кількість інформації, що стосується навчального процесу, студентів, оцінювання, викладачів та інших пов'язаних сутностей.

Ще однією суттєвою перевагою є чудова інтеграція MySQL із мовою програмування PHP, яка є основною мовою реалізації проєкту. Це значно спрощує процес взаємодії між серверною логікою системи та базою даних. У свою чергу, phpMyAdmin забезпечує зручний і зрозумілий інтерфейс для керування базою даних, що дає змогу швидко виконувати необхідні операції — створення таблиць, редагування даних, побудову зв'язків між сутностями тощо.

		Джус І.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		22

Під час проєктування бази даних було створено діаграму класів, що відображає основні логічні сутності системи та взаємозв'язки між ними:



Весь розглянутий сегмент бази даних працює з даними сесії та успішності студентів. Центральним об'єктом є таблиця students, яка є проміжним пунктом між даними, що були імпортованими з ЄДЕБО[19], посилаючись на їх

внутрішній ключ `education_id`, що є персональним для кожного студента та таблицями для зберігання результатів оцінювання.

Одиницею зберігання відомостей є `exam_sheet`, кожна з яких містить інформацію про час створення, час, протягом якого вона буде відкритою до редагування, групу, курс, семестр, дисципліну та форму навчання. Тип відомості вказує, чи є вона основною чи додатковою - основна відомість може мати безліч додаткових, і отриманим фінальним балом буде вважатися останній, що не є `null`. Для додавання таких відомостей існує таблиця `exam_sheet_child`.

Також за відомістю закріплений певний викладач у таблиці `exam_sheet_staff` для забезпечення його доступу.

Так як в одній відомості може зберігатися безліч оцінок студентів, було створено таблицю балів `exam_sheet_points`. Вона також зберігає інформацію про студента, якому належить ця оцінка, дата отримання, а також передбачає можливість розділення балів на дві частини - наприклад, окрема оцінка за практику та модульний контроль.

Вищезгаданий алгоритм виставлення оцінок передбачає знаходження останньої, що буде вважатися остаточною і видимою в кабінеті студента. Ця функція була доволі важкою в обчисленнях та потребувала багато часу, тому додавання таблиці `exam_sheet_last_point` вирішило питання оптимізації.

При виставленні було передбачено можливість вписання умовних позначень, які додаються до таблиці `exam_sheet_notation` - таким чином, будь-який символ може означати певну кількість балів.

Так як кожен студент має можливість погодити або відхилити оцінку, ця інформація зберігається у таблиці `exam_sheet_point_agreement` та містить інформацію про статус і дату зміни стану.

Не менш важливим є і використання системи підписання відомостей. Вона складається із двох таблиць, `exam_sheet_sign_person` та `exam_sheet_sign`, перша з яких містить інформацію про посаду людини, що підписує документ, порядок та вимоги до підписання документів в цифровій системі. Сам факт

		Джус І.О.			ПІЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		24

підписання певної відомості буде внесений до exam_sheet_sign, завірений файловим документом електронно-цифрового підпису та з вказаним шляхом до нього у папці проекту.

2.3 Постановка задачі

При створенні додатку насамперед потрібно детально спланувати його ключові аспекти. Одним із таких моментів у нашій системі є процес виставлення оцінок студентам. На перший погляд, це проста задача, однак враховуючи масштабність порталу, величезну кількість студентів, дисциплін та одночасних користувачів потрібно подумати про оптимізований[20] шлях реалізації.

Так як процес виставлення оцінок є непростим та ресурсозатратним, наступна діаграма послідовності відображає процес присвоєння оцінки користувачеві. Розглянемо її більш детально:

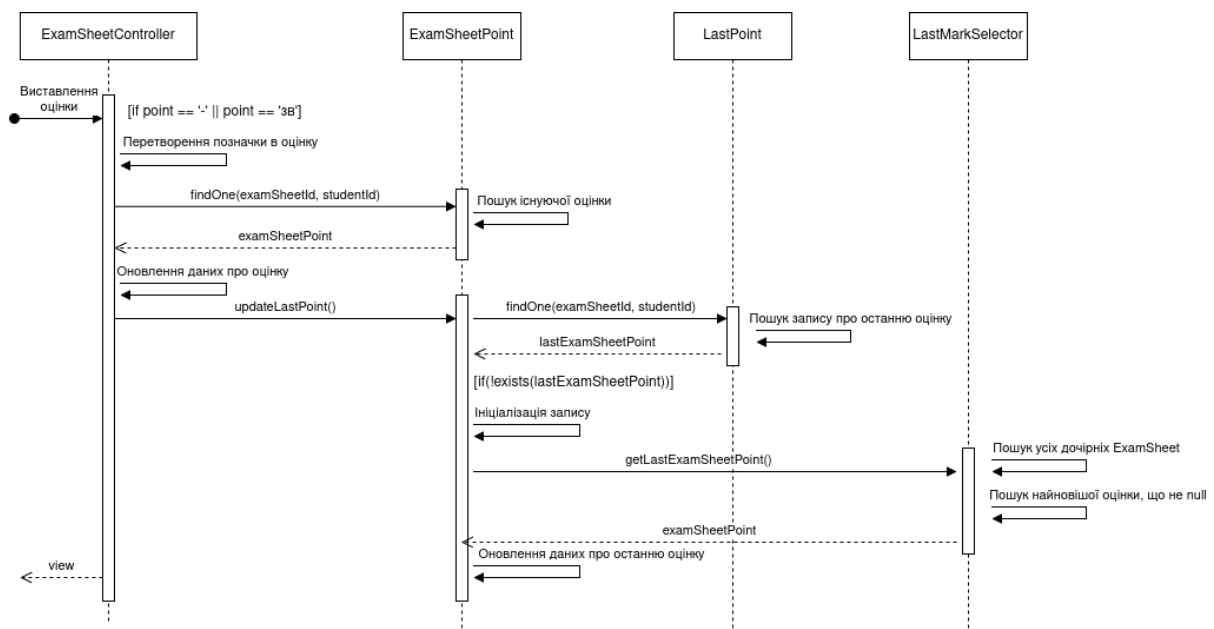


Рис. 2.4. Діаграма послідовності

Викладачі мають можливість виставляти не лише числове, а й умовне позначення, наприклад, “-”, що означає неатестацію студента, або “зв”, що означає “звільнення”. В такому разі, алгоритм звернеться до словника і шукатиме відповідне числове значення в системі оцінювання.

Якщо у студента вже існувала попередня оцінка і відбувається її зміна - запис в бд не створюватиме новий рядок, а оновить наявний.

Зважаючи на те, що університет має можливість створювати основні та додаткові відомості, потрібно було розробити логіку, завдяки якій ми зможемо знаходити останню виставлену оцінку, що не є null - для зберігання такої інформації було створено таблицю та модель LastPoint.

Після збереження оцінки в БД спрацьовує подія в ExamSheetPoint, яка розпочне процес оновлення запису в LastPoint, якщо такий є, інакше ініціалізує його.

Пошуком останньої оцінки займається сервіс LastMarkSelector, який вмикає циклічний пошук усіх дочірніх відомостей, а також одним побудованим SQL-запитом знаходить оцінку із останньої відомості, що не є null.

Далі саме ця оцінка буде відображатися студенту у його кабінеті для погодження.

2.4 Реалізація функціоналу системи управління навчальним процесом

Додаток складається із монолітної архітектури, реалізованої на PHP фреймворку Yii2, що поділяється на дві окремі частини для адміністрації та студентів відповідно. Сайт побудовано за схемою патерну MVC та з використанням рендерінгу на серверній частині.

Розпочнемо огляд із наявних контролерів, що забезпечують функціонування модуля “Сесія”. Одним із них є ExamSheetController:

```
public function actionIndex(): string
{
    $searchModel = new ExamSheetSearch();

    $filter = Yii::$app->request->get('filter', 'all');
    $signFilter = Yii::$app->request->get('signFilter', 'all');
    $dataProvider = $searchModel->search(Yii::$app->request->queryParams,
    $filter, $signFilter);

    $examSheetIds = array_map(function ($model) {
        return $model->id;
    }, $dataProvider->models);
```

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

```

        foreach ($dataProvider->models as $model) {
            $model->is_sign = $model->isSign;
        }

        $signPersons = ExamSheetSignPerson::find()->where(['is_active' =>
'1'])->orderBy(['position' => SORT_DESC])->all();
        $signPersonStatuses = null;

        if ($signPersons) {
            $signPersonStatuses = ExamSheetSignService::getSignPersonStatuses($examSheetIds, $signPersons);
        }

        return $this->render('index', [
            'searchModel' => $searchModel,
            'dataProvider' => $dataProvider,
            'signPersonStatus' => $signPersonStatuses,
        ]);
    }

```

Згідно правил фреймворку Yii2, контролер та метод мають таку ж назву з приставками Controller та action відповідно, якою вони будуть представлені у url адресі веб-додатку. У даному методі відбувається пошук всіх відомостей, що належать людині, яка зараз здійснює перегляд індексової сторінки. Моделі пошуку містять логіку пошуку певних записів на основі отриманих даних із GET-параметрів. Також за допомогою ExamSheetSignService здійснюється пошук статусів підписання кожної з відображених відомостей:

```

public static function getSignPersonStatuses(array $examSheetIds, array
$signPersons): array
{
    $signPersonStatus = [];

    foreach ($examSheetIds as $examSheetId) {
        $signPersonStatus[$examSheetId] =
self::getSignPersonStatus($examSheetId, $signPersons);
    }

    return $signPersonStatus;
}

```

Метод приймає дані у вигляді масив ідентифікаторів усіх відомостей, які зараз доступні в перегляді, а також масив усіх ролей, від кого необхідний підпис – наприклад, це може бути викладач, завідувач кафедри та декан. Далі для кожної із відомостей запускається такий алгоритм:

		Джус І.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				27
Змн.	Арк.	№ докум.	Підпис	Дата		

```

public static function getSignPersonStatus($examSheetId, $signPersons):
array
{
    $currentUserId = Yii::$app->user->id;
    $currentExamSheet = ExamSheet::findOne($examSheetId);
    $currentComponentType = EducationalComponentType::find()->where(['id' =>
    $currentExamSheet->educational_component_type_id])->one();

    if (!$currentComponentType->is_need_to_sign) {
        return [];
    }

    $persons = self::getPersonsForExamSheet($examSheetId);
    $signPersonStatus = [];

    foreach ($signPersons as $person) {
        $personStatuses = array_filter($persons, function ($p) use ($person)
        {
            return $p['role'] === $person->type;
        });

        $isSigned = false;

        if ($person->behavior === 'all') {
            $isSigned = !empty($personStatuses) &&
array_reduce($personStatuses, function ($carry, $p) {
                return $carry && $p['sign_status'];
            }, true);
        } elseif ($person->behavior === 'one') {
            $isSigned = !empty($personStatuses) &&
array_reduce($personStatuses, function ($carry, $p) {
                return $carry || $p['sign_status'];
            }, false);
        }

        $signByCurrentUser = ExamSheetSign::find()
            ->where([
                'person_id' => $currentUserId,
                'exam_sheet_id' => $examSheetId,
                'person_type' => $person->type
            ])
            ->one();

        $signPersonStatus[] = [
            'label' => $person->label,
            'status' => $isSigned,
            'is_sign_by_current_user' => $signByCurrentUser !== null,
            'sign_date' => $signByCurrentUser->datetime ?? null,
            'position' => $person->position,
        ];
    }

    usort($signPersonStatus, function ($a, $b) {
        return $a['position'] <=> $b['position'];
    });

    return $signPersonStatus;
}

```

		Джус І.О.			ІІЗ.КР.Б – 121 – 25 – ІІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		28

Метод виконує декілька важливих функцій - задля забезпечення можливостей підписання документів було розроблено спеціальну систему зміни порядку підписання, вимог щодо підписання, а також систему відображення вже зібраних електронно цифрових підписів [22].

Згідно з побудованою системою всередині бази даних, адміністрація університету може змінювати порядок та потребу підписання відомостей - наприклад, за налаштувань за замовчуванням відомість мають підписати спочатку усі викладачі, що пов'язані з виставленням оцінок для певної групи студентів з певної дисципліни, далі один із завідувачів кафедри і після цього завершує підписання декан факультету, до якого прив'язана подана відомість. За бажання користувач, що має доступ до налаштування системи, може змінювати ці правила - наприклад, дозволити підписання лише одним із викладачів, змінити порядок підписання або ж прибрати необхідність підписання певною уповноваженою особою взагалі.

Метод `getSignPersonStatus` визначає, чи є необхідність у підписанні документа, перевіряючи поточні налаштування, а також визначає поточну чергу підписання за допомогою перевірки, була позначка на підписання документу усіма членами поточної ролі 'all' чи дозволено підписати відомість однією людиною 'one'. Якщо цей тип освітнього компонента не потребує підписання і відповідна інформація зберігатиметься всередині поля `is_need_to_sign` - метод завчасно завершить виконання і цикл виконання перейде до наступної відомості. усіма Для визначення порядку використовується сортування усіх пунктів в черзі підписання за полем 'position'. Після цього метод поверне список статусів підписання поданої відомості, зовнішній метод виконає це для усіх переданих записів у базі даних й це можна буде відобразити візуально у вигляді активних і неактивних html-елементів.

Перейдемо до методу перегляду окремої відомості, де також є можливість побачити всі оцінки:

		Джус І.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		29

```

$model = $this->findModel($id);
$model->date_end = $model->dateEnd;
$academicGroup = AcademicGroup::findOne(['id' => $model->group_id]);

$students = Student::find()
    ->select('education_id')
    ->where(['academic_group_id' => $academicGroup->id])
    ->column();

```

Метод виконує пошук студентів по групі, що прив'язана до поданої відомості.

Далі відбувається сортування студентів за алфавітом із врахуванням особливих літер української мови (і, ї, є, г):

```

$collator = new Collator('uk_UA');
usort($points, function ($a, $b) use ($collator) {
    return $collator->compare($a->student_name, $b->student_name);
});

```

Знаходимо всі дані про оцінки:

```

$points_id = ExamSheetPoint::find()
    ->select('id')
    ->where(['exam_sheet_id' => $model->id])
    ->column();

```

Для коректного відображення оцінок, а також всієї інформації про те, хто так коли її виставив, відбувається пошук усіх викладачів, прив'язаних до поточної відомості, а також у циклах відбувається пошук погоджених оцінок:

```

$teachers = ExamSheetStaff::find()->select('staff_id')-
>andWhere(['exam_sheet_id' => $model->id])->column();
$expandedTeachers = Staff::find()->andWhere(['id' => $teachers])->all();

$restructuredData = [];
foreach ($points_id as $point_id) {
    $agreedPoints = ExamSheetPointAgreement::find()
        ->where(['point_id' => $point_id])
        ->all();

    foreach ($agreedPoints as $agreement) {
        $restructuredData[$point_id] = [
            'choice_type' => $agreement->choice_type,
            'datetime' => $agreement->datetime,
        ];
    }
}

```

Усі ці дані будуть використані при рендерінгу для правильного відображення інформації про успішність студентів:

		Джус І.О.			ПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		30

```

$signPersons = ExamSheetSignService::getPersonsForExamSheet($model->id);
return $this->render('view', [
    'points' => $points,
    'teachers' => $expandedTeachers,
    'model' => $model,
    'academicGroup' => $academicGroup,
    'providerExamSheetPoint' => $providerExamSheetPoint,
    'providerExamSheetStaff' => $providerExamSheetStaff,
    'agreedPoints' => $restructuredData,
    'course' => $course,
    'signPersons' => $signPersons,
    'is_sign' => $isSign,
    'examSheetsWithSameParent' => $this->getExamSheetsWithSameParent($id),
    'isCanGenerate' =>
ExamSheetSummaryService::canGenerateAdditionalExamSheet($id),
]);

```

В функцію рендерінгу передається інформація про те, чи була ця відомість підписана, чи не вимкнена генерація додаткових відомостей на її основі, а також усі попередньо зібрані масиви даних, які будуть виведені у правильному порядку.

При виставленні оцінки спрацьовує подія пошуку тієї, що необхідна для занесення в ExamSheetLastPoint:

```

protected function updateLastPoint($event): void
{
    /** @var ExamSheetPoint $sender */
    $sender = $event->sender;

    $examSheetId = $sender->exam_sheet_id;
    $originalExamSheetId =
AdditionalExamSheetGenerator::getOriginalExamSheetIdOf($examSheetId);
    $studentId = $sender->student_id;

    $lastExamSheetPoint = $this->initLastPoint($studentId, $examSheetId,
$originalExamSheetId);

    $selector = new LastMarkSelector();
    $point = $selector->getLastExamSheetPointOf($sender->student_id,
$originalExamSheetId);

    if ($point) {
        $lastExamSheetPoint->exam_sheet_id = $point->exam_sheet_id;
        $lastExamSheetPoint->points_part1 = $point->points_part1;
        $lastExamSheetPoint->points_part2 = $point->points_part2;
        $lastExamSheetPoint->points_final = $point->points_final;

        $lastExamSheetPoint->save(false);
    } else {
        $lastExamSheetPoint->delete();
    }
}

```

		Джус І.О.			ПІЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				31
Змн.	Арк.	№ докум.	Підпис	Дата		

Подану логіку було розроблено з метою можливості додавання нового функціоналу – наприклад, якщо умова для фінальної оцінки студента буде не “остання можлива”, а “найкраща” - це буде досить швидко та просто реалізувати.

Наразі при виставленні оцінки відбувається пошук її основної відомості, якщо бал був доданий до додаткової. При відсутності запису про останній бал для поточного студента з поточної дисципліни рядок в базі даних буде ініціалізовано. За допомогою сервісу LastMarkSelector відбувається пошук інших оцінок цього студента:

```
public function getLastExamSheetPointOf(int $studentId, int $originalExamSheetId): ExamSheetPoint|null
{
    $childrenIds = AdditionalExamSheetGenerator::getExamSheetWithAncestorsIds($originalExamSheetId);
    $allExamSheetIds = array_merge([$originalExamSheetId], $childrenIds);

    $point = $this->getExamSheetPointOfStudent($studentId, $allExamSheetIds);

    if (is_null($point)) {
        $point = ExamSheetPoint::find()->where(["exam_sheet_id" => $allExamSheetIds, 'student_id' => $studentId])->one();
    }

    return $point;
}
```

Відбувається збір усіх дочірніх відомостей до основної, до масиву додається основна відомість, і за допомогою ітерування по ньому буде визначатися остання оцінка, що є в межах заданого діапазону системи оцінювання університету, заданого в налаштуваннях, і не є позначкою про неатестацію. Пошук було побудовано за допомогою використання спеціального запиту:

```
public function getExamSheetPointOfStudent(int $studentId, array $examSheetIds): ExamSheetPoint|null
{
    $query = ExamSheetPoint::find()
        ->andWhere(['exam_sheet_id' => $examSheetIds])
        ->andWhere(['student_id' => $studentId])
        ->andWhere(['not', ['points_final' => null]],
    )
        ->orderBy(['exam_sheet_id' => SORT_DESC]);
    return $query->one();
}
```

		Джус І.О.			ПІЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				32
Змн.	Арк.	№ докум.	Підпис	Дата		

Якщо жодна із оцінок не входить в перелік вищезгаданих правил, метод поверне null – тоді в моделі ExamSheetLastPoint збережеться відповідна інформація і студент буде вважатися неатестованим, якщо йому не випаде ще одна нагода виправити оцінку в новій додатковій відомості.

Однак не обов’язково заходити на сторінку перегляду і редагувати оцінки вручну – для великих обсягів даних було створено систему імпортування існуючих балів із .csv файлів:

```
private function validateAndImport(string $filePath)
{
    Yii::$app->response->format = Response::FORMAT_JSON;

    $linker = new ExcelExamSheetPointsDoubleLinker();
    $errors = $linker->validate($filePath);
    $isEmpty = $this->areAllInnerArraysEmpty($errors);

    Yii::$app->response->format = Response::FORMAT_JSON;
    if (count($errors) > 0 && !$isEmpty) {

        $sheetNames = ExcelParser::getSheetNames($filePath);
        return ['errors' => $errors, 'sheetNames' => $sheetNames];
    }

    $linker->import($filePath);

    return [
        'success' => true,
        'message' => "Оцінки успішно імпортовано",
    ];
}
```

При відсиланні файлу для імпорту успішності студентів використовується наступний сервіс ExcelExamSheetPointsDoubleLinker – назва свідчить про можливість реалізації як імпорту, так і експорту в майбутньому. Першочергово запускається процес валідації – всередині інформаційної системи було створено інструкцію щодо перевірки окремих груп полів в файлі Excel на валідність інформації. Ця інструкція є статичною, однак за необхідності її можна змінити. Інструкція має такий вигляд:

```
private function getDisciplineValidationRules(string $filePath, int $page,
string $letter): ExcelParser
{
    $parser = new ExcelParser($filePath, $page);

    $parser
        ->addRule($letter . "5", "subject_name", new EmptyExcelValidator(),
ExcelErrorType::ERROR)
```

		Джус І.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		33

```

        ->addRule($letter . "6", "count_credits", new NumberRangeValidator(0,
240, false), ExcelErrorType::ERROR)
        ->addRule($letter . "7", "form_control", new FormControlValidator(),
ExcelErrorType::ERROR)
        ->addRule($letter . "8", "education_component_type", new
EducationalComponentTypeNameOrShortNameValidator(), ExcelErrorType::ERROR)
        ->addRule($letter . "9", "control_date", new DateValidator(),
ExcelErrorType::ERROR)
        ->addRule($letter . "10", "number_information", new
EmptyExcelValidator(), ExcelErrorType::ERROR)
        ->addRule($letter . "11", "teacher", new EmptyExcelValidator(),
ExcelErrorType::ERROR);

    return $parser;
}

```

Це один із різновидів інструкцій, який потребує на вході назву стовпця, в якому йде перевірка. Правила перевірки описуються у вигляді реалізації патерну Builder – кожна з функцій повертає об'єкт ExcelParser. Функція приймає клітинку або стовпець, який підпадає під перевірку, назву поля в масиві після збору даних, валідатор, що контролює поле, а також тип помилки - це може бути Error (помилка, що не дозволяє подальші дії із файлом) або Warning (рекомендація щодо покращення якості файлу).

Кожен із валідаторів наслідує абстрактний клас ExcelValidator. Його ключовою особливістю є здатність накопичення інформації під час перевірки – один валідатор може повернути не одну, а декілька помилок завдяки таким функціям:

```

public function clearErrors(): void
{
    $this->errors = [];
}

protected function addError(string $error): void
{
    $this->errors[] = $error;
}

```

Це розширює можливості валідаторів та позбавляє необхідності використовувати return під час знаходження першої несправності у файлі.

Розглянемо клас EducationalComponentTypeNameOrShortNameValidator - подана конструкція дозволяє перевірити, чи існує в системі запис про тип освітнього компоненту, взятий із імпортованого файлу. Якщо такого типу не знайдено в базі даних, отже, це є помилкою:

		Джус І.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		34

```

class EducationalComponentTypeNameOrShortNameValidator extends
ExcelValidator
{
    /**
     * @param $value
     * @return void
     */
    public function validate($value): void
    {
        $isExists = EducationalComponentType::find()
            ->where(['or',
                ['name' => $value],
                ['short_name' => $value]
            ])
            ->exists();

        if (!$isExists) {
            $this->addError("Неправильно вказано тип освітнього компонента");
        }
    }
}

```

Аналогічним чином, використовуючи об'єктно-орієнтоване програмування та однаковий підхід, можна створити будь-яке правило та використати його при додаванні до черги виклику:

```

public function addRule(string $cell, string $fieldName, ExcelValidator
$validator, int $errorType): ExcelParser
{
    $this->rules[] = [
        'cell' => $cell,
        'field' => $fieldName,
        'validator' => $validator,
        'errorType' => $errorType,
    ];
    return $this;
}

```

Для ініціювання процесу валідації необхідно викликати функцію, що пробіжитья по масиву rules. Перевірки клітинки та стовпців в документі мають різні функції, однак їх принцип дії однаковий, тому розглянемо його детальніше на прикладі перевірки комірок:

```

private function validateCell(array $rule): void
{
    $cellValue = $this->getCellString($rule['cell']);

    /** @var $validator ExcelValidator */
    $validator = $rule['validator'];
    $validator->validate($cellValue);

    $errors = $validator->getErrors();
}

```

		Джус І.О.			ПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		35

```

if (!empty($errors)) {

    $uniqueErrors = array_unique($errors);

    foreach ($uniqueErrors as $error) {
        $this->addError($rule['cell'], $error, $rule['errorType']);
    }
}
}

```

Відбувається отримання значення клітинки, беремо переданий екземпляр класу валідатора, з якого викликається метод validate. Всі його накопичені помилки під час перевірки зберігаються у змінній errors, які далі будуть додані до загального переліку помилок. Це відбувається для кожного створеного правила валідації, і подана система перевірки неймовірно спрощує подальшу підтримку функціоналу - за її допомогою можливо швидко створити перевірку імпорту інших даних без жорстких прив'язок та величезних нечитабельних обчислень масивів даних, що вже знайшло застосування у інших модулях системи.

Правила діють на одну сторінку в Excel-документі [22], що також додає гнучкості до використання і дозволяє обробляти файли з різною структурою аркушів. При роботі з імпортуванням оцінок було створено окрему функцію, що містить усі підфункції-інструкції на кожну сторінку:

```

$groupRules = $this->getGroupValidationRules($filePath, $page);
$studentRules = $this->getStudentValidationRules($filePath, $page,
$groupId);
$disciplineRules = [];
$pointsRules = [];

foreach ($lettersArr as $letter) {
    $disciplineRules[] = $this->getDisciplineValidationRules($filePath,
$page, $letter);
    $pointsRules[] = $this->getPointsValidationRules($filePath, $page,
$letter);
}

```

Кожна група правил була окремо викликана:

```

$groupRulesErrors = $groupRules->validate()->getErrors();

$studentRulesErrors = $studentRules->validate()->getErrors();
for ($i = 0; $i < count($disciplineRulesArr); $i++) {
    $disciplineRule = $disciplineRulesArr[$i];
    $pointsRule = $pointsRulesArr[$i];

    $disciplineErrors = $disciplineRule->validate()->getErrors();
}

```

		Джус І.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				36
Змн.	Арк.	№ докум.	Підпис	Дата		

```

$pointsErrors = $pointsRule->validate()->getErrors();

$allDisciplinesErrors      =      array_merge($allDisciplinesErrors,
$disciplineErrors);
$allPointsErrors = array_merge($allPointsErrors, $pointsErrors);
}

```

Після чого усі помилки були з'єднані між собою в єдиний масив:

```

return      array_merge($groupRulesErrors,      $studentRulesErrors,
$allDisciplinesErrors, $allPointsErrors);

```

Якщо валідація була успішна – файл можна імпортувати в систему:

```

if (count($errors) > 0 && !$isEmpty) {

    $sheetNames = ExcelParser::getSheetNames($filePath);
    return ['errors' => $errors, 'sheetNames' => $sheetNames];
}

$linker->import($filePath);

```

Перед імпортуванням відбувається ще один не менш важливий етап - парсинг даних і їх збереження в асоціативному масиві в пам'яті з назвами, що були визначені при створенні валідаційних правил:

```

$rulesArr      =      $this->getExamSheetPointsValidationRules($filePath,
$page);
/** @var ExcelParser $groupRules */
$groupRules = $rulesArr['group_rules'];
$studentRules = $rulesArr['student_rules'];

if ($groupRules->isEmpty()) {
    return [];
}

$groupRulesData = $groupRules->parse();
$studentRulesData = $studentRules->parse();

```

Парсинг являє собою зчитування даних по комірках із файлу і їх додавання в асоціативний масив із заданою назвою ключа:

```

public function parseDataCells(array $cellRules): array
{
    $rowData = [];

    foreach ($cellRules as $rule) {
        $cellValue = $this->getCellString($rule['cell']);

        $rowData[$rule['field']] = $cellValue;
    }

    return $rowData;
}

```

		Джус І.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		37

Сам безпосередній імпорт виконується через функцію `importExamSheet`, що приймає накопичені у асоціативні масиви дані і номер листа. Спочатку необхідно впевнитись, що таких відомостей не існує в системі:

```
$existingExamSheet = $this->findExistingExamSheet($data, $i);

if ($existingExamSheet) {
    return $existingExamSheet;
}
```

Далі після проходження попередньої перевірки необхідно імпортувати відомість:

```
$semester = $this->calculateSemester(
    $this->getGroupDataValue($dataArray, 'semester'),
    $this->getGroupDataValue($dataArray, 'study_year')
);

$examSheet->dictionary_semester_id = $semester->id;
$examSheet->educational_component_id = $this->getEducationComponent(
    $this->getDisciplineDataValue($dataArray, $i, 'subject_name')
);
```

Відбувається пошук поточного року та поточного семестру в базі даних, на основі зчитаної назви предмету знаходимо цю дисципліну у системі.

Аналогічно, на основі інших даних, що були отримані з комірок файлу успішності, збирається новий екземпляр моделі `ExamSheet` та зберігається:

```
$examSheet->form_of_control_id = $this->getFormControlId(
    $this->getDisciplineDataValue($dataArray, $i, 'form_control')
);
$examSheet->registration_date = $this->getDate($this->
    >getDisciplineDataValue($dataArray, $i, 'control_date'))->format('Y-m-d');
$examSheet->sheet_number = trim($this->getDisciplineDataValue($dataArray,
    $i, 'number_information')) ?: "-";
$examSheet->control_date = $this->getDate($this->
    >getDisciplineDataValue($dataArray, $i, 'control_date'))->format('Y-m-d');
$examSheet->control_time = null;
$examSheet->days_open = 0;

$academicGroup = AcademicGroup::findOne(["name" => $this->
    >getGroupDataValue($dataArray, 'group_name')]);
$examSheet->group_id = $academicGroup->id;
$examSheet->exam_sheet_type_id = ExamSheetType::find()->select('id')-
    >where(['short_name' => "Основна"])->scalar();

$year = $semester->academicYear;
$course = DateParserService::calculateGroupCourse($academicGroup-
    >study_start, $year->year_start);

$examSheet->course = $course;
$examSheet->queue = 1;
$examSheet->save(false);
```

		Джус І.О.			ІПЗ.КР.Б – 121 – 25 – ІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		38

Важливими даними є група, курс, форма контролю, ідентифікаційний номер відомості, час та дата контролю, термін до закриття відомості. За допомогою цих параметрів кожен запис можна вважати унікальною і в подальшому розпочати імпорт оцінок:

```
public function importPointsForDiscipline(array $students, ExamSheet
$examSheet, array $points, Staff $staff)
{
    for ($i = 0; $i < count($students); $i++) {
        $man = $students[$i];
        $mark = $points[$i];

        $point = ExamSheetPoint::findOne(['exam_sheet_id' => $examSheet-
>id, 'student_id' => $man]);

        if (!$point) {
            $point = new ExamSheetPoint();
            $point->exam_sheet_id = $examSheet->id;
            $point->student_id = $man->education_id;
            $point->student_name = $man->person_fio;
            $point->staff_id = $staff->id;

            $point->points_final = $mark;

            if (is_null($mark)) {
                $point->staff_name = null;
                $point->datetime = null;
            } else {
                $point->staff_name = $staff->full_name;
                $point->datetime = $examSheet->control_date;
            }

            $point->save(false);
        }
    }
}
```

Метод отримує масив зчитаних студентів з листа і починає ітеруватися по ньому. По обраним іменам студентів було отримано моделі Student, звідки ми дізналися їх student_id. Також було передбачено, що існуючі оцінки не потрібно затирати, тому завчасно перевіряється, чи існує такий запис в базі даних.

Створюється нова модель ExamSheetPoint, яка заповнюється раніше отриманою інформацією про відомість та студента. Якщо оцінку не було виставлено - ПІБ викладача та час виставлення оцінки залишається пустим.

Кожен студент має можливість переглянути власні оцінки із персонального кабінета. Для зручності та оптимізації роботи серверу було прийнято рішення реалізувати API-взаємодію між сервером та згенерованими

		Джус І.О.			ПІЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		39

на бекенді сторінками. Ось так виглядає отримання усіх сторінок користувачем – веб-сторінка посилає асинхронний запит на отримання даних про успішність:

```
Yii::$app->response->format = Response::FORMAT_JSON;
$lastMarkSelector = new LastMarkSelector();

$lastPoints = $lastMarkSelector->getAllLastExamSheetPointsOfStudent($educationId);
```

Відповідь буде надіслана у вигляді асоціативного масиву із статусом відповіді та іншими накопиченими необхідними даними в ході виконання обробки запиту:

```
return [
    'status' => 'success',
    'data'   => $result
];
```

Також студент може погодити виставлену йому оцінку – для цього на панелі керування йому потрібно буде натиснути відповідну кнопку, яка надішле запит на представлений метод:

```
public function actionAgreePoint(): Response
{
    $data = Yii::$app->request->getRawBody();
    $data = json_decode($data, true);

    $user = User::findOne(['id' => Yii::$app->user->identity->getId()]);
    $personId = $user->person_id;

    if (isset($data['point_id']) && isset($data['choice_type'])) {
        $agreement = new ExamSheetPointAgreement();
        $agreement->point_id = $data['point_id'];
        $agreement->choice_type = $data['choice_type'];
        $datetime = new DateTime('now', new DateTimeZone('Europe/Kiev'));
        $agreement->datetime = $datetime->format('Y-m-d H:i:s');
        $point = ExamSheetPoint::findOne($data['point_id']);
        $students = ActiveStudent::find()->where(['person_id' => $personId])->all();
        if ($point && in_array($point->student_id, array_column($students, 'education_id'))) {
            if ($agreement->save()) {
                return $this->asJson([
                    'status' => 'success',
                    'message' => 'Agreement saved successfully.'
                ]);
            }
        }
        return $this->asJson([
            'status' => 'error',
            'message' => 'Invalid data.'
        ]);
    }
}
```

		Джус І.О.			ПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				40
Змн.	Арк.	№ докум.	Підпис	Дата		

Внаслідок такої дії буде створено запис у таблиці ExamSheetPointAgreement, що збереже статус підтвердження у БД. Після знаходження користувача і перевірки його ідентифікатора також додано валідацію, чи належить оцінка поточному користувачеві – студент має можливість підтверджувати лише власні оцінки.

Скасування підтвердження має подібну логіку, однак відбувається видалення даних:

```
if ($agreement->delete()) {  
    return $this->asJson([  
        'status' => 'success',  
        'message' => 'Agreement canceled successfully.'  
    ]);  
}
```

Таким чином всередині одного проєкту відбувається взаємодія користувачів, що являються адміністрацією та здійснюють оцінювання студентів, а також користувачів, що визначені як студенти і отримують виставлені бали.

Висновки до другого розділу

Під час проєктування та розробки системи керування освітнім процесом було сформульовано як функціональні, так і нефункціональні вимоги, визначено ключові цілі проєкту, а також сформовано перелік основного функціоналу для кожної категорії користувачів — адміністраторів, викладачів та студентів. Особливу увагу приділено забезпеченню зручного доступу до навчальних матеріалів, веденню обліку успішності та автоматизації організаційних процесів.

У процесі аналізу було створено структуровану модель даних, що охоплює всі сутності освітнього середовища, після чого розроблено діаграму класів, яка відображає взаємозв'язки між компонентами системи. Архітектура додатку розроблена відповідно до затвердженого технічного завдання та враховує потребу в масштабованості, безпеці та простоті підтримки.

Результати реалізації свідчать про відповідність створеної системи початковим вимогам: забезпечено функціональність для ведення електронного

		Джус І.О.			ПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				41
Змн.	Арк.	№ докум.	Підпис	Дата		

журналу, погодження та відхилення оцінок, імпорту звітів за минулі роки та накладання електронного підпису. Таким чином, проєктна ціль була досягнута — створено ефективний інструмент для організації та оптимізації освітнього процесу з урахуванням сучасних вимог до цифрових навчальних платформ.

		Джус І.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		42

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

3.1 Порядок встановлення та налаштування параметрів додатку

Для більшого розуміння архітектури майбутнього програмного забезпечення було прийнято рішення відобразити його схематичний принцип дії на діаграмі розгортання [23] (рис. 3.1):

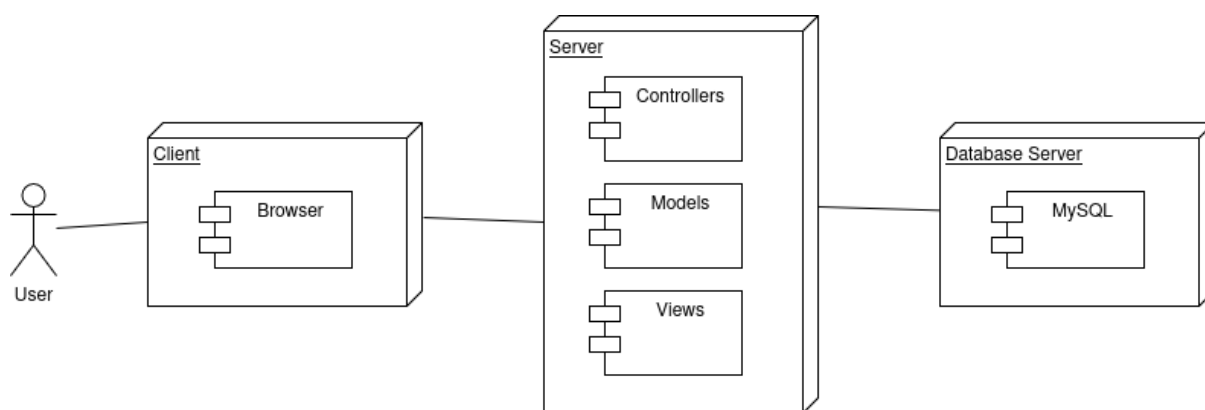


Рис. 3.1. Діаграма розгортання

Діаграма розгортання демонструє, як компоненти вебзастосунку адміністрації університету, побудованого за архітектурою MVC [24], розподіляються між фізичними вузлами в рамках монолітної структури. Клієнт використовує власний пристрій і здійснює вхід через браузер. Уся логіка програми виконується на стороні сервера в межах єдиного додатку, що розгортається на одному вебсервері і здійснює підключення до бази даних.

Серверна частина включає такі основні елементи:

- **Controllers:** отримують та обробляють HTTP-запити [25] користувачів, відповідають за маршрутизацію, виклик відповідних сервісів і передачу даних до представлень.
- **Models (Entities):** описують основні бізнес-сутності системи (наприклад, Student, Teacher, Discipline, SessionRecord) та реалізують логіку взаємодії з базою даних згідно з ORM-підходом Yii2 (Active Record).

- Views: відповідають за відображення інформації користувачеві, генерують HTML-код сторінок, використовуючи дані, передані контролерами.

Цілісний характер додатку спрощує його розгортання, оскільки всі компоненти функціонують у межах єдиного програмного середовища, що працює на веб сервері, з підключенням до реляційної бази даних MySQL.

Розгортання здійснюється на сервері, характеристики якого відповідають вимогам, представлених у пункті 2.1. Розроблена інформаційна система орієнтована на зручність і доступність для кінцевого користувача. Для цього передбачено повну сумісність із більшістю сучасних веб-браузерів. Система коректно функціонує в таких браузерах, як Google Chrome (версія 45 і вище), Mozilla Firefox (версія 38 і вище), Microsoft Edge, Safari, а також інших популярних оглядачах. Це забезпечує гнучкість у використанні та дозволяє користувачам взаємодіяти із системою на будь-якому пристрої без необхідності встановлення додаткового програмного забезпечення.

Завдяки дотриманню стандартів веб-розробки інтерфейс залишається стабільним, адаптивним [26] і зрозумілим незалежно від вибраного браузера, що суттєво підвищує зручність роботи з системою.

3.2 Структура інтерфейсу додатку

Система онлайн менеджменту освітніх процесів має інтуїтивно зрозумілий інтерфейс та підтримує адаптивність - користувачі можуть заходити на сайт як з мобільних пристроїв, так із ноутбуків чи комп'ютерів з моніторами з великою роздільною здатністю [27]. Перед входом на головну сторінку порталу користувач має увійти у свій обліковий запис (рис. 3.2):

		Джус І.О.			ПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		44

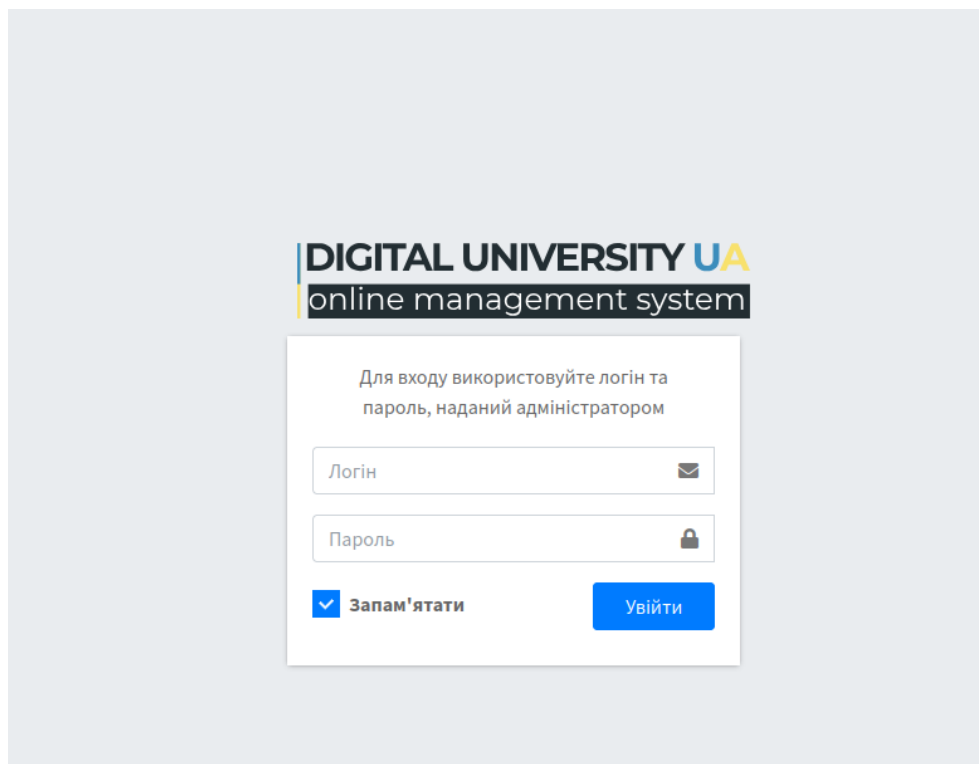


Рис. 3.2. Вхід на портал

З метою безпеки даних, на сайті неможливо зареєструватися - в університеті, що використовує подану платформу, має бути відповідна структура, що займається видачею логінів та паролів для отримання доступу.

Після успішної авторизації користувач потрапляє на головну сторінку, де велику кількість модулів і підмодулів було згруповано у лівій бічній панелі для зручності (рис.3.3):

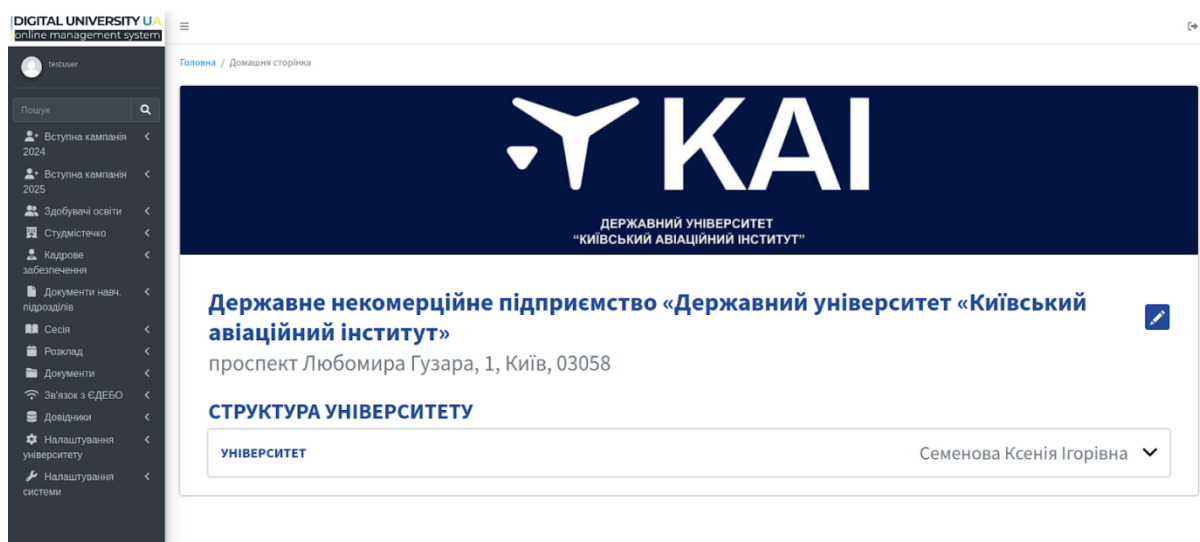


Рис. 3.3. Головна сторінка

		Джус І.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				45
Змн.	Арк.	№ докум.	Підпис	Дата		

Серед усіх представлених модулів ми сконцентруємо увагу на компоненті “Сесія” (рис. 3.4):

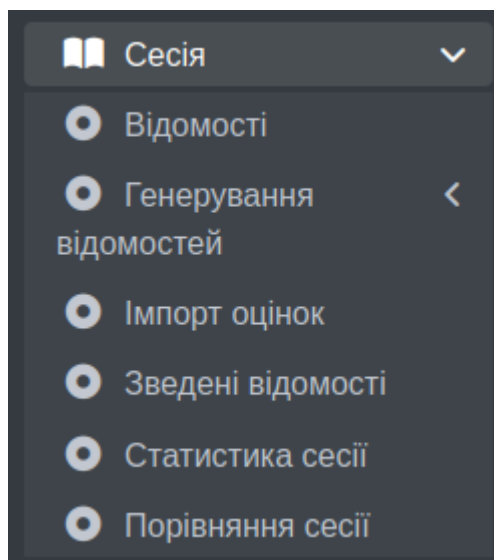


Рис. 3.4. Модуль “Сесія”

Перейдемо до сторінки перегляду відомостей. Початкова сторінка міститиме перелік усіх документів, що доступні для перегляду користувачеві з поточною роллю. Ми ведемо перегляд від імені адміністратора, тож бачитимемо усі відомості, наявні у системі:

Головна / Сесія / Відомості

Відомості обліку успішності

Показані 1-15 із 31 429 записів.

Всі статуси

Усі відомості

Усі відомості

Підписати усі доступні відомості

#	Тип відомості	Курс	КЕП	Дата контролю	Час контролю	Група	Форма навчання	К-ть здобувачів	К-ть оцінок	Дисципліна	Тип освітнього о
1	Основна відомість	3	ВЗД	09.06.2025	-	Б-122-21-2-УС	Денна	1	0	Комп'ютеризовані інформаційні управлінські системи	Навчальна ди
2	Основна відомість	3	ВЗД	09.06.2025	-	Б-122-21-2-УС	Денна	26	0	Комп'ютеризовані інформаційні управлінські системи	Навчальна ди
3	Основна відомість	1	ВЗД	26.05.2025	-	Б-134-24-1-ОС	Денна	1	0	Ділова українська мова	Навчальна ди
4	Основна відомість	4	ВЗД	27.05.2025	-	Б-141-21-1-МН	Денна	1	1	Електричне обладнання трансформаторних підстанцій	Навчальна ди
5	Основна відомість	4	ВЗД	27.05.2025	-	Б-141-21-1-МН	Денна	1	1	Електричне обладнання трансформаторних підстанцій	Курсова ри
6	Додаткова відомість	4	ВЗД	28.05.2025	-	Б-141-21-1-МН	Денна	1	1	Електросвітлотехнічне обладнання аеропортів	Навчальна ди
7	Додаткова відомість	2	ВЗД	28.05.2025	-	Б-172-23-1-ТО	Денна	1	1	Енергосистеми повітряних суден	Навчальна ди
8	Додаткова відомість	4	ВЗД	28.05.2025	-	Б-134-21-1-ОС	Денна	1	0	Системи повітряних суден	Курсова ри
9	Додаткова відомість	2	ВЗД	28.05.2025	-	Б-053-23-1-ПС	Заочна	2	0	Практикум з загальної психології	Навчальна ди
10	Додаткова відомість	2	ВЗД	28.05.2025	-	Б-053-23-1-ПС	Заочна	1	0	Юридична психологія	Навчальна ди
11	Додаткова відомість	2	ВЗД	28.05.2025	-	Б-053-23-1-ПС	Заочна	6	0	Психодіагностика	Навчальна ди
12	Додаткова відомість	2	ВЗД	28.05.2025	-	Б-053-23-1-ПС	Заочна	1	0	Гендерна психологія	Навчальна ди
13	Додаткова відомість	2	ВЗД	28.05.2025	-	Б-053-23-1-ПС	Заочна	1	0	Основи суїцидології	Навчальна ди
14	Додаткова відомість	2	ВЗД	28.05.2025	-	Б-053-23-1-ПП	Заочна	2	0	Юридична психологія	Навчальна ди
15	Додаткова відомість	2	ВЗД	28.05.2025	-	Б-053-23-1-ПП	Заочна	3	0	Психологія особистості	Навчальна ди

1

2

3

4

5

6

...

Рис. 3.5. Відомості обліку успішності

На сторінці відображається інформація, отримана з БД, виводячи тип відомості, групу, курс, дисципліну, кількість студентів у групі, викладачів, яким вона належить, а також статус підписання у вигляді блоків із першою літерою назви посади людини, що має підписати відомість - Викладач, Завідувач

		Джус І.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		46

кафедри та Декан. Зелена відмітка свідчить про те, що підпис відповідної особи чи групи осіб було здійснено, і черга переходить до того, в кого відмітка червона.

Також на сторінці присутні фільтри та пагінація, що є необхідністю для здійснення зручної навігації.

Перейдемо до перегляду відомості (рис. 3.6):

до 20.12.2021

Основна відомість | 481 від 20.12.2021

Викладачі

Перенесення оцінки

ФПКП | 053 Психологія

Б-053-21-1-ПП | 1 КУРС

+

✓

📄

Психологія особистісного життєпроекування

Навчальна дисципліна | Диференційований залік | К-сть кредитів: 3

Дата контролю: 20.12.2021 / -

Стипендія: Враховано

Навч. семестр:

1 сем. 2021/2022

Семестр:

1

Осв. ступінь:

Бакалавр

Форма навчання:

Денна

Осв. програма:

Практична психологія

Статус групи:

Активна

№	ПІБ	ОЦІНКА	ЄКТС	ЗА НАЦІОНАЛЬНОЮ ШКАЛОЮ	ДАТА	НІП
1	Белік Юлія Олександрівна	75	C	Добре	20.12.2021 00:00:00	Перенесення оцінки
2	Білявська Дар'я Віталіївна	76	C	Добре	20.12.2021 00:00:00	Перенесення оцінки
3	Богачова Валерія Юріївна	90	A	Відмінно	20.12.2021 00:00:00	Перенесення оцінки
4	Вдровню Крістіна Олександрівна	76	C	Добре	20.12.2021 00:00:00	Перенесення оцінки
5	Гребенікова Наталя Дмитрівна	90	A	Відмінно	20.12.2021 00:00:00	Перенесення оцінки
6	Жарко Аліна Олександрівна	82	B	Добре	20.12.2021 00:00:00	Перенесення оцінки

Рис. 3.6. Перегляд відомості

Під час перегляду можна змінювати оцінки студентів за умови, якщо відомість є відкритою. Як ми бачимо на рисунку 3.6, час на редагування, зазначений у верхньому лівому куті, вийшов, тож наразі бали змінити неможливо. Крім балу у числовій системі, автоматично вираховується його позначка за шкалою ЄКТС, а також національною шкалою. Останніми двома стовпцями є дата виставлення оцінки та викладач, який виконав дану дію. Як ми бачимо на рисунку 3.6, у останньому стовпці замість ПІБ викладача міститься запис “Перенесення оцінки”, що свідчить про те, що дана відомість була імпортована.

За необхідності, можна відредагувати відомість або створити нову (рис.3.7):

		Джус І.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		47

Дата реєстрації 12 / 20 / 2021	Номер відомості 481
Семестр 1 сем. 2021/2022	Курс 1
Дата контролю 12 / 20 / 2021	Час контролю --:--:--
Група 6-653-21-3-01	Враховувати для стипендії Враховувати
Освітній компонент Психологія особистісного матеріального	Тип ОК Навчальна дисципліна
К-сть кредитів 3	Вибірковий Не вибірковий
К-сть днів на виставлення оцінок 0	Форма контролю Диференційований залік

Студенти Оберть... Бетік Юлія Олександрівна Білявська Дар'я Віталіївна Богачова Валерія Юріївна Вдовенко Крістіна Олександрівна	Викладачі Оберть... Перенесення оцінки
---	---

Зберегти Відмінити

Рис. 3.7. Створення та редагування відомості

Для редагування доступні усі параметри, що відображаються на перегляді. При додаванні викладача, що існує в системі, ми зможемо отримати можливість підписати відомість (рис. 3.8):

Електронні підписи	
СТАТУС	ВИКЛАДАЧ
Підписати	Паскал Жанна Валеріївна, викладач
	Сібрук Анастасія Володимирівна, завідувач кафедри
	Мельник Наталія Іванівна, декан факультету

Рис. 3.8. Доданий викладач

Наразі черга підписання викладача, тож увійшовши в систему від його імені, ми маємо можливість поставити його підпис. Натискаємо на кнопку “Підписати” і переходимо на наступну сторінку (рис. 3.9):

Головна / Підпис електронної відомості

Підписання відомостей

Крок 1.

01. 39171_orig.pdf 0.03 MB Завантажити файл

Крок 2.

На цьому кроці файли будуть підписані електронним підписом. Вибір електронний підпис та слідуйте вказівкам, які будуть відображатися.

Зчитайте ключ

Файловий Токен Хмарний

Копіювання надання електронних довірчих послуг

Визначити автоматично

Перетягніть сюди файл ключа або завантажте його зі свого носія (Зазначений його назва "key-6.dat" або ".pfx", ".pk8", ".p12", ".jks")

Пароль захисту ключа

Зчитати

Що таке файловий носій?

Файловий носій – це спеціальний файл, який містить ваш особистий ключ.

Зазвичай цей файл має назву Key-6 з розширенням ".dat" (зустрічаються також розширення ".pfx", ".pk8", ".p12", ".jks").

Для того, щоб скористатись файловим носієм, вам необхідно:

1. Обрати зі списку свого надавача електронних довірчих послуг – суб'єкта, до якого ви зверталися для отримання електронного підпису.
2. Завантажити із зовнішнього носія

Рис. 3.9. Сторінка підписання

		Джус І.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				48
Змн.	Арк.	№ докум.	Підпис	Дата		

Користувач бачить взятий з публічного джерела віджет Eusign [28], куди потрібно передати ключ та пароль захисту ключа підпису. Відбувається генерація PDF файлу документу відомості, віджет бере на себе обробку алгоритму підписання, і поверне на сервер підписаний файл та його метадані. Файли зберігатимуться в директорії проекту та будуть доступні до завантаження зі сторінки перегляду відомостей.

Будь-яка подальша зміна відомості або виставлення нової оцінки анулюють підписи, і процедуру підписання потрібно буде повторити. Відомість та її оцінки можна імпортувати, маючи файл із відповідною структурою (рис. 3.10):

А		В	С	Д	Е	Ф	Г	Н	І	Ж
		Академічний рік	Б-272-21-1-ТО							
		Навчальний рік	2023/2024							
		Семестр	2							
		Висхідна відомість	Висхідна відомість	Висхідна відомість	Висхідна відомість	Висхідна відомість	Висхідна відомість	Висхідна відомість	Висхідна відомість	Висхідна відомість
		Номер відомості	7	1.5	5.5	4.5	4.5	4	4	6
		Формат контролю	диф.звіт	диф.звіт	диф.звіт	диф.звіт	диф.звіт	диф.звіт	диф.звіт	диф.звіт
		Тип ОК	диф.звіт	диф.звіт	диф.звіт	диф.звіт	диф.звіт	диф.звіт	диф.звіт	диф.звіт
		Дата	14.05.2024	25.04.2024	25.04.2024	23.06.2024	14.06.2024	26.04.2024	19.04.2024	5/2/2024
		№ відомості	587/3	420/3	424/3	461/3	434/3	232/3	419/3	239/3
		Висхідна	Степанов Сергій	Хомік Сергій	Хомік Сергій	Хомік Андрій	Довго Дмитро	Степан Сергій	Штан Сергій	Михайлов
		Висхідна	Висхідна	Висхідна	Висхідна	Висхідна	Висхідна	Висхідна	Висхідна	Висхідна
		1	ВІШІЙ Вадим Вячеславович	90	90	85	85	82	91	90
		2	БОНДАРЕНКО Олександр Романович	68	70	85	75	62	60	90
		3	ДЗОВЕНКО Олена Ігорівна	-	90	-	90	96	92	91
		4	ДРОБИЧУК Максим Валерійович	60	82	82	77	65	70	70
		5	ЗУРНАДЖИ Олександр Михайлович	75	86	82	82	78	80	91
		6	КАЛЕНКО Дмитро Олександрович	68	80	85	82	76	94	85
		7	КОВАЛЕНКО Дмитро Андрійович	93	92	98	92	95	94	93
		8	КОРМАН Максим Олександрович	69	68	67	82	78	79	67
		9	КОСТЮК Олександр Михайлович	-	75	90	75	60	65	70
		10	ЛЕВЧЕНКО Олександр Дмитрович	88	82	75	90	60	95	92
		11	МОРГУЛЕЦЬ Віктор Олександрович	69	82	80	85	82	85	60
		12	ПЕТРИШАК Андрій Васильович	85	85	82	90	95	94	90
		13	ТАРАСЕНКО Назар Юрійович	75	82	85	75	75	60	75
		14	ТИДІНОК Валентин Юрійович	94	88	75	82	97	95	96
		15	ТУРЧАНИНКО Михайло Костянтинович	98	98	98	95	98	96	96
		16	ЦІВІК Сергій Васильович	-	-	-	61	60	-	65
		17	ЩАЛАСВ Сергій Олександрович	65	73	82	70	93	80	86

Рис. 3.10. Файл для імпортування оцінок

Згідно правил імпорту, усі аркуші повинні мати однакову структуру, де в зазначених комірках вписується назва групи, курс, роки навчання, а нижче представляється таблиця із студентами, оцінками кожного з них та викладачами, що їх виставили.

Перейшовши до відповідної сторінки та завантаживши файл Excel, кнопка “Імпортувати” буде активною (рис.3.11-3.12):

Завантаження файлу з оцінками

Оберіть файл ...

Обрати

Імпортувати

Завантаження файлу з оцінками

семестр 2023-2024 _6_.xlsx

Видалити

Обрати

Імпортувати

Рис. 3.11.-3.12. Завантаження файлу

Після натискання на кнопку запуститься процес валідації. Якщо у файлі виявлено помилки - виведеться відповідне повідомлення (рис. 3.13):

Помилки валідації

Аркуш Б-142-21-1-ГУ

С1: Група 'Б-142-21-1-ГУ85' не існує в таблиці академічних груп.

В12: Не знайдено студента/студентку: БОЙЧУК Максим Ігорович

В13: Не знайдено студента/студентку: БОРОВІК Микита Юрійович

В14: Не знайдено студента/студентку: БУДНІК Дарина Ростиславівна

В15: Не знайдено студента/студентку: ГОРДІЙЧУК Віктор Сергійович

Закрити

Рис. 3.13. Помилки валідації

Кожен валідатор спрацював і обробив призначену йому групу комірок. Як видно з рисунку 3.12, файл не пройшов перевірку, тож перед імпортом його потрібно виправити. Після усунення всіх несправностей та успішної валідації було створено відомість по кожній групі оцінок, підв'язано відповідних викладачів та виставлені бали, а також здійснено переадресацію на головну сторінку відомостей. Розглянемо одну із успішно імпортованих відомостей групи Б-142-21-1-ГУ, а саме з дисципліни “Теорія газотурбінних установок” (рис. 3.14):

Головна / [Сесія](#) / [Відомості](#) / Теорія газотурбінних установок / Б-142-21-1-ГУ

Основна відомість (24.05.2024 - 24.05.2024)

до 24.05.2024

Основна відомість | 416/3 від 24.05.2024

Викладачі
Перенесення оцінки

АКФ | 142 Енергетичне машинобудування

Б-142-21-1-ГУ | 3 КУРС

Теорія газотурбінних установок

Навчальна дисципліна | Екзамен | К-сть кредитів: 6

Дата контролю: 24.05.2024 / -

Стипендія: **Враховано**

Навч. семестр:
2 сем. 2023/2024
Семестр:
6

Осв. ступінь:
Бакалавр
Форма навчання:
Денна

Осв. програма:
Газотурбінні установки і компресорні станції
Статус групи:
Активна

Електронні підписи

СТАТУС	ВИКЛАДАЧ
	Корольов Петро Васильович, викладач
	Терещенко Юрій Матвійович, завідувач кафедри
	Юцкевич Святослав Сергійович, декан факультету

№	ПІБ	ОЦІНКА	ЕКТС	ЗА НАЦІОНАЛЬНОЮ ШКАЛОЮ	ДАТА	НПП
1	Бойчук Максим Ігорович	71	D	Задовільно	24.05.2024 00:00:00	Перенесення оцінки
2	Боровік Микола Юрійович	73	D	Задовільно	24.05.2024 00:00:00	Перенесення оцінки
3	Буднік Дарина Ростиславівна	92	A	Відмінно	24.05.2024 00:00:00	Перенесення оцінки
4	Гордійчук Віктор Сергійович	90	A	Відмінно	24.05.2024 00:00:00	Перенесення оцінки

Рис. 3.14. Імпортована відомість

На сторінці перегляду зведених відомостей можна побачити інформацію по всіх відомостях вибраної групи (рис 3.15):

01.09.2021 - 30.06.2025

Кількість студентів: 11

142 Енергетичне машинобудування

Газотурбінні установки і компресорні станції

Б-142-21-1-ГУ | 3 КУРС

Осв. ступінь:
Бакалавр

Форма навчання:
Денна

Бюджетна основа:
8

Контрактна основа:
3

Ваучери:
0

Зведена відомість обліку успішності

Створити відомість

		Теорія газотурбінних установок (Курсовий проєкт)	Теорія газотурбінних установок	Ремонтно-виробнича практика	Першеві двигуни і компресори	Тривимірне проектування елементів турбінних двигунів	Конструкція і міцність газотурбінних установок і компресорів	Система автоматизованого проектування енергетичних машин
Кредитів:		1	6	3	4	4	5.5	6.5
Дата контролю:		24.05.2024	24.05.2024	16.06.2024	22.05.2024	16.05.2024	22.05.2024	30.05.2024
Форма контролю:		диф.залік	екзамен	диф.залік	диф.залік	диф.залік	екзамен	екзамен
Тип ОК:		КП	Д	П	Д	Д	Д	Д
Стипендіальна:		✓	✓	✓	✓	✓	✓	✓
Основні відомості:		+ 0	+ 0	+ 0	+ 0	+ 0	+ 0	+ 0
Додаткові відомості:		+	+	+	+	+	+	+
1	Бойчук Максим Ігорович	Контракт	71 / D	71 / D	-	61 / E	72 / D	61 / E
2	Боровік Микита Юрійович	Бюджет	73 / D	73 / D	-	60 / E	72 / D	63 / E
3	Буднік Дарина Ростиславівна	Бюджет	93 / A	92 / A	90 / A	91 / A	60 / E	87 / B
4	Гордійчук Віктор Сергійович	Бюджет	91 / A	90 / A	92 / A	82 / B	67 / D	92 / A
5	Зубов Микита Сергійович	Бюджет	80 / C	80 / C	-	62 / E	60 / E	62 / E

Рис. 3.15. Зведені відомості

Власні отримані оцінки доступні для перегляду тим, хто навчається. Студенти також можуть заходити в систему, однак через іншу адресу – ідея розгалуження точок входу адміністрації університету та здобувачів освіти було реалізовано задля попередження ризиків отримання несанкціонованого доступу до панелі керування [29] освітнім процесом. Сторінка для студентів має такий вигляд (рис.3.16):

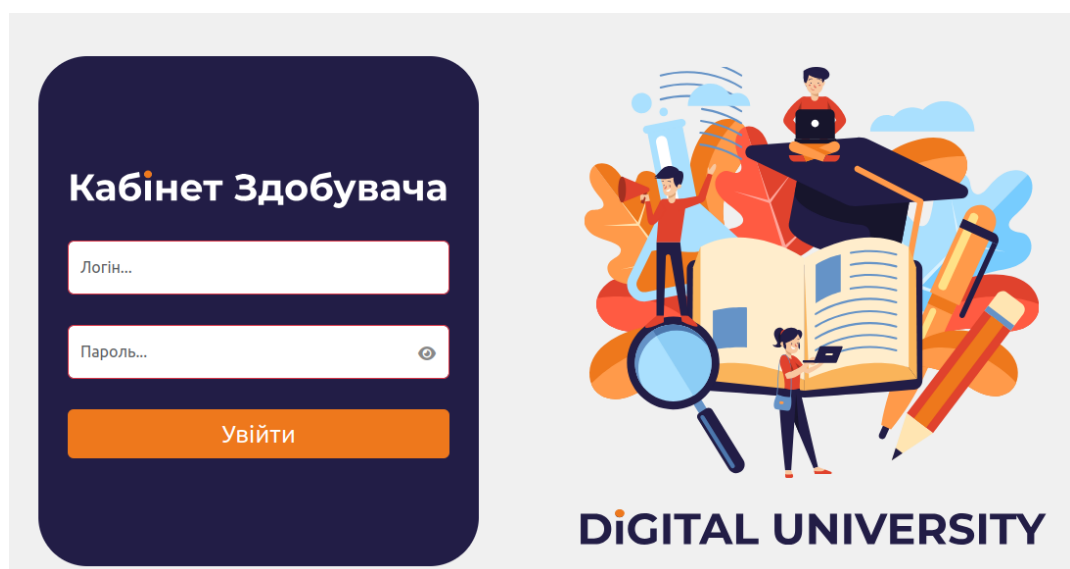


Рис. 3.16. Сторінка входу студента

		Джус І.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				52
Змн.	Арк.	№ докум.	Підпис	Дата		

Після успішної авторизації здобувача освіти перенесе на головну сторінку (рис. 3.17):

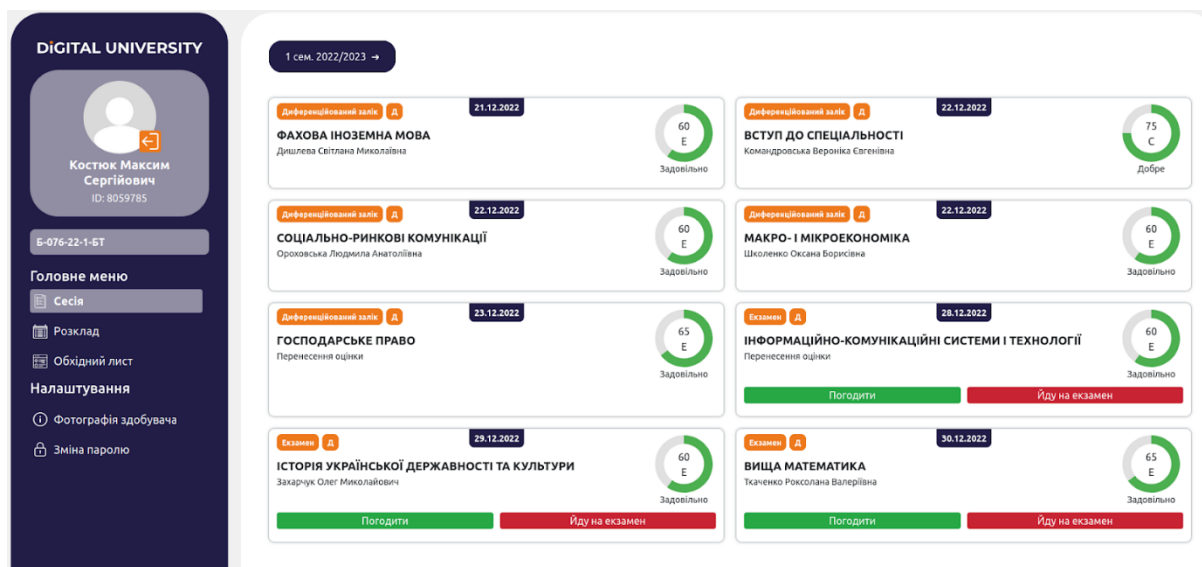


Рис. 3.17. Головна сторінка

Розглянемо детальніше картку дисципліни (рис. 3.18):

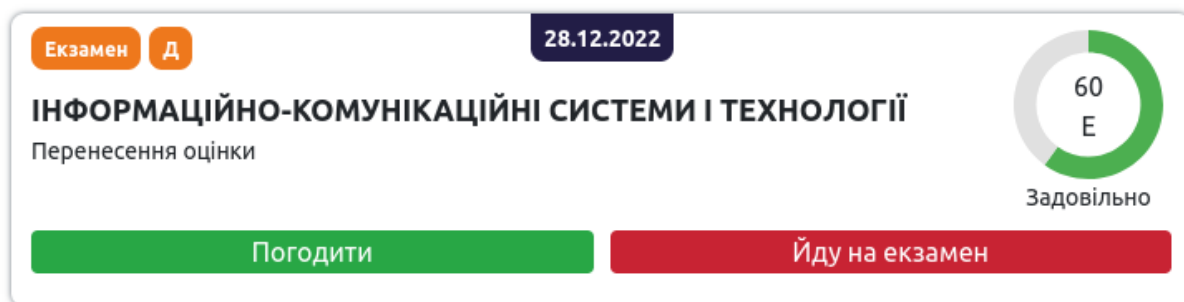


Рис. 3.18. Картка дисципліни

Студент має можливість погоджувати оцінки, або ж йти на екзамен. Така можливість властива лише на дисциплінах, відомість яких не була закрыта. У верхньому лівому куті зазначається інформація про форму контролю, а також тип освітнього компоненту - “Д” означає “дисципліна”, а “КР” - “курсова робота”. По центру виводиться дата проведення іспиту, а у правій частині картки - кількість отриманих балів та їх кількість у інших шкалах оцінювання. Ми можемо вибрати перегляд оцінок за будь-який інший семестр. Перейдемо до першого семестру 2024-2025 років (рис. 3.19-3.20):

		Джус І.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		53

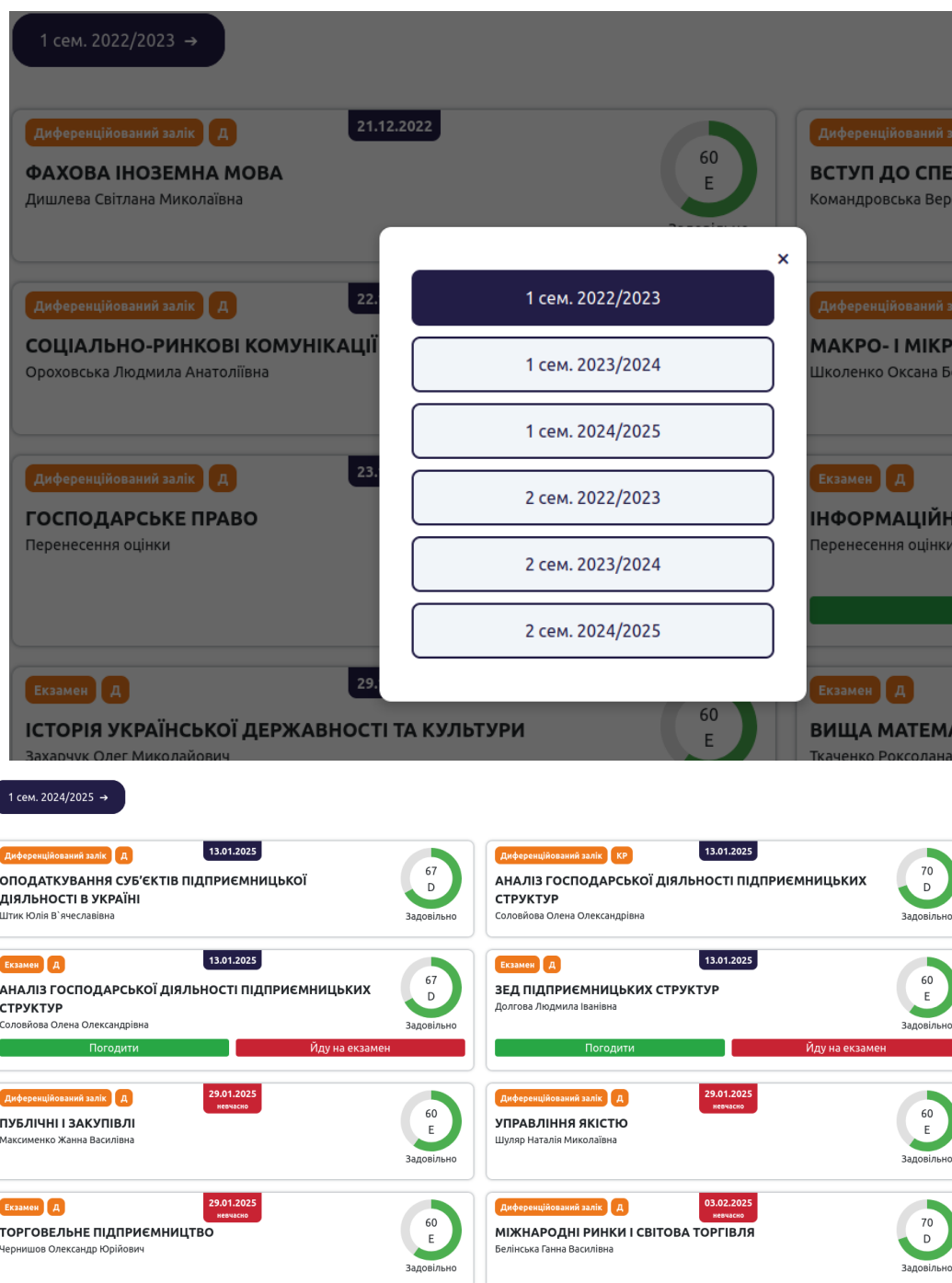


Рис. 3.19-3.20. Картка дисципліни

Адміністрація університету потребує статистичні дані задля розуміння стану успішності та можливості прийняття подальших рішень задля її покращення. Тому було розроблено функціонал для створення та перегляду статистики з фільтруванням (рис.3.25):

		Джус І.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		54

Перегляд статистики станом на 06.05.2025 16:36



Рис. 3.25. Статистика успішності студентів по університету

Гістограма [30] демонструє кількість успішних складань сесії студентами кожного з факультетів, а горизонтальна пряма показує середнє значення по університету. Нижче можна отримати більш детальні показники за кожним з критеріїв вимірювання.

Аналогічно, працює і можливість порівняння статистики за декілька обраних періодів - користувач може обирати будь-які часові проміжки, за які було створено збереження статистики, та візуально переглядати зміну у параметрах (рис.3.26-3.27).

Порівняння статистики

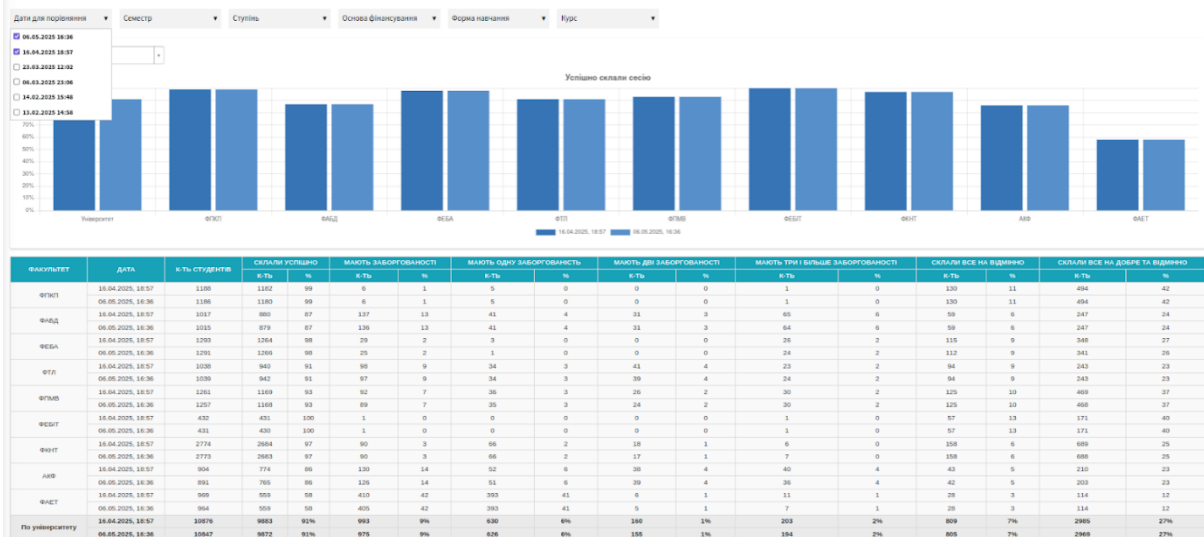


Рис. 3.26. Статистика за декілька проміжків

Порівняння статистики

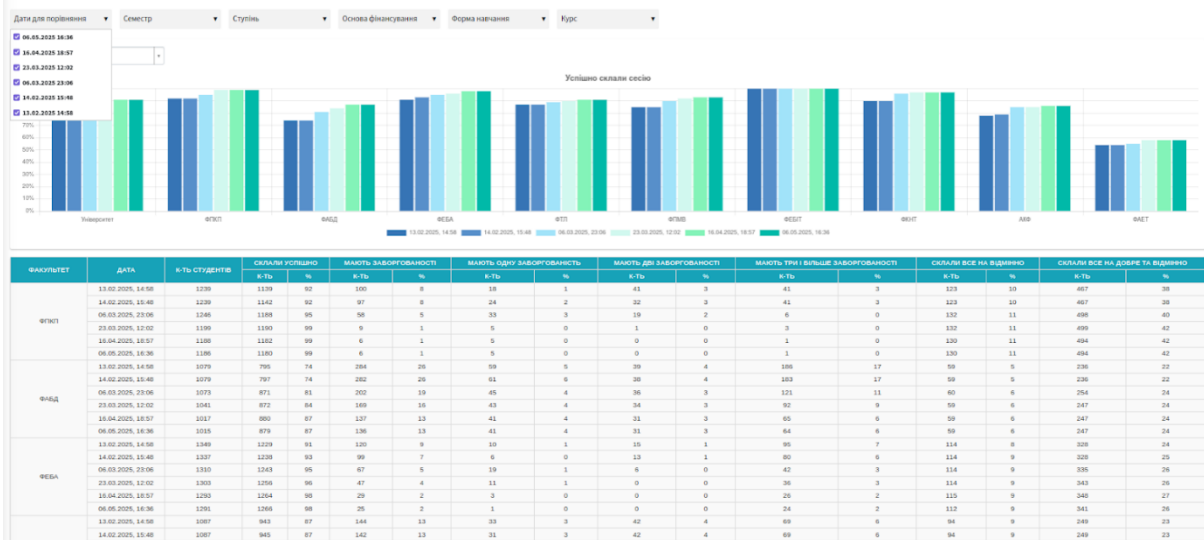


Рис. 3.27. Статистика за декілька проміжків

Таким чином, модуль “Сесія” надає можливості для керівників освітнього процесу здійснювати оцінювання студентів, імпортувати результати за попередні роки та використовувати набуті дані для аналітичних потреб, а також представляє можливості для здобувачів освіти вчасно отримувати інформацію про своє навчання і завчасно повідомляти університет про присутність на екзамені у вигляді схвалення чи відхилення власної оцінки у кабінеті.

Висновки до третього розділу

У третьому розділі було розгорнуто додаток на девайсі, що відповідає вимогам, а також протестовано отриманий функціонал – користувачі, що представляють адміністрацію, мають можливість логінитись та виконувати дії над оцінками студентів, створювати нові відомості та імпортувати попередні, а також формувати та переглядати статистику успішності студентів за обраний проміжок часу. Здобувачі освіти мають персональний кабінет, де виводяться отримані оцінки за обраний семестр, а також надається можливість завантажувати фото з очікуванням на схвалення від модератора. Додаток було протестовано, і як висновок – програма успішно виконує поставлені задачі.

		Джус І.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		57

ВИСНОВКИ

У межах виконання курсової роботи було реалізовано повноцінний цикл розробки модуля «Сесія» для інформаційної системи онлайн-менеджменту освітніх процесів, що дозволяє автоматизувати ключові аспекти організації та контролю заліково-екзаменаційної сесії.

На етапі аналітичного дослідження було проведено огляд актуальних технологій розробки, оцінено їхні переваги та недоліки. На основі цього обґрунтовано вибір мови програмування PHP та фреймворку Yii2, що дозволило досягти високої швидкості розробки, підтримувати гнучку архітектуру і забезпечити безпечну роботу з базами даних. Прийнята монолітна архітектура системи оптимізувала процес реалізації та зменшила складність обслуговування.

У процесі проектування сформульовано чіткі функціональні та нефункціональні вимоги, окреслено основні ролі користувачів і визначено їхній функціонал. Розроблені діаграми класів і варіантів використання наочно відобразили структуру та логіку взаємодії в системі. Приділено значну увагу зручності доступу до функцій оцінювання, імпорту/експорту звітності, підтвердження результатів та інтеграції з цифровими інструментами, такими як електронний підпис.

На заключному етапі було проведено розгортання і тестування системи. Перевірено коректність авторизації, роботу з відомостями, функціонал студентських кабінетів і модерацію завантажених даних. Усі заявлені функції реалізовано у повному обсязі, що підтверджує відповідність результату початковим технічним вимогам і завданням проекту.

Таким чином, розроблений модуль «Сесія» демонструє дієве рішення для цифровізації освітнього процесу, підвищення його прозорості, ефективності та зручності як для адміністрації, так і для здобувачів освіти, і може бути інтегрований у повноцінну систему управління навчальним закладом.

		Джус І.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		58

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Проблеми та перспективи цифрової трансформації освіти: психолого-педагогічний аспект/ Петришин Л.Й., Почуєва О.О., Лемещук М.А., Звоздецька В.Г. – Академічні Візії, 2023. – 7 с.
2. Цифровізація неперервної професійної освіти. Неперервна професійна освіта: теорія і практика / Сисоєва С., 2021, С. 24–32.
3. Інформаційна система управління освітнім процесом в університеті Огієнка [Електронний ресурс] – Режим доступу до ресурсу: <https://meridian.kpnu.edu.ua>
4. iZETA [Електронний ресурс] – Режим доступу до ресурсу: <http://www.izeta.puet.edu.ua/>
5. Комплексна інформаційна система автоматизації управління навчальним процесом [Електронний ресурс] – Режим доступу до ресурсу: <https://ena.lpnu.ua:8443/server/api/core/bitstreams/5eebd9d5-1abc-4637-b3b6-92de8b5e702e/content>
6. Алгоритми та структури даних. Основи алгоритмізації / Кублій Л.І. - Київ : КПІ ім. Ігоря Сікорського, 2022. – 528 с.
7. Мікросервіси. Від архітектури до релізу / Іраклі Надареїшвілі, Ронні Мітра - O'Reilly Media, 2022 – 318 с.
8. PHP Cookbook: Modern Code Solutions for Professional Developers / Eric Mann - O'Reilly Media, 2023 – 420 с.
9. The Definitive Guide to Yii 2.0 [Електронний ресурс] – Режим доступу до ресурсу: <https://www.yiiframework.com>
10. Особливості формулювання вимог до програмного забезпечення / Грицюк Ю. І., Нємова О. А. - Національний університет "Львівська політехніка", 2018 – 14с.
11. GeeksForGeeks Client-Server Model [Електронний ресурс] – Режим доступу до ресурсу: <https://www.geeksforgeeks.org/client-server-model/>

		Джус І.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		59

- 12.UX Design – Основи UI/UX [Електронний ресурс] – Режим доступу до ресурсу: <https://www.interaction-design.org/literature/topics/ui-design>
- 13.MDN Web Docs: Browser Compatibility [Електронний ресурс] – Режим доступу до ресурсу: https://developer.mozilla.org/en-US/docs/Web/HTTP/Browser_compatibility
- 14.Об’єктно-орієнтований аналіз та проектування з прикладами на UML 2.5 / Граді Буч, Джеймс Рамбо, Айвор Джейкобсон – Діалектика, 2020. – 512 с.
- 15.Організація обліку успішності студентів у закладах вищої освіти [Електронний ресурс] – Режим доступу до ресурсу: <https://mon.gov.ua/ua/npa/pro-zatverdzhennya-formi-vedomosti-obliku-uspishnosti-studentiv-ta-instrukcii-shodo-yiyi-zapovnennya>
- 16.Європейська кредитна трансферна система (ЄКТС) [Електронний ресурс] – Режим доступу до ресурсу: <https://osvita.ua/vnz/reform/45690/>
- 17.MySQL 8.0 Reference Manual [Електронний ресурс] – Режим доступу до ресурсу: <https://dev.mysql.com/doc/refman/8.0/en/>
- 18.phpMyAdmin Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://www.phpmyadmin.net/docs/>
- 19.ЄДЕБО – Єдина державна електронна база з питань освіти [Електронний ресурс] – Режим доступу до ресурсу: <https://info.edbo.gov.ua/>
- 20.High Performance PHP Programming / Rasmus Lerdorf, Cristian Darie – Wrox Press, 2003. – 432 с.
- 21.Криптографічні засоби інформаційного захисту. Електронний цифровий підпис / О. В. Семенець, А. В. Тимошенко – Київ: КНУ імені Тараса Шевченка, 2020. – 208 с.
- 22.Microsoft Excel 2021. Самовчитель / Галина Шелестова – ВHV, 2022. – 384 с.
- 23.UML 2.0 Візуальне моделювання та конструювання програмних систем / Граді Буч, Джеймс Рамбо, Айвар Якобсон – Діалектика, 2007. – 496 с.

		Джус І.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				60
Змн.	Арк.	№ докум.	Підпис	Дата		

- 24.PHP Objects, Patterns, and Practice / Mika Schwartz, Matt Zandstra – Apress, 2021. – 500 с.
- 25.HTTP: The Definitive Guide / David Gourley, Brian Totty – O'Reilly Media, 2002. – 656 с.
- 26.Responsive Web Design with HTML5 and CSS / Ben Frain – Packt Publishing, 2022. – 404 с.
- 27.Дизайн для різних екранів: роздільна здатність, щільність пікселів і адаптивні сітки [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/en-US/docs/Web/Guide/Responsive/Overview>
- 28.EU Sign – Кваліфікований електронний підпис [Електронний ресурс] – Режим доступу до ресурсу: <https://www.eusign.com.ua/>
- 29.PHP & MySQL: Server-side Web Development / Jon Duckett – Wiley, 2022. – 832 с.
- 30.Data Visualization: A Practical Introduction / Kieran Healy – Princeton University Press, 2018. – 296 с.

		Джус І.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				61
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

		Джус І.О.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		62

```

public function actionIndex(): string
{
    $searchModel = new ExamSheetSearch();

    $filter = Yii::$app->request->get('filter', 'all');
    $signFilter = Yii::$app->request->get('signFilter', 'all');
    $dataProvider = $searchModel->search(Yii::$app->request->queryParams,
    $filter, $signFilter);

    $examSheetIds = array_map(function ($model) {
        return $model->id;
    }, $dataProvider->models);

    foreach ($dataProvider->models as $model) {
        $model->is_sign = $model->isSign;
    }

    $signPersons = ExamSheetSignPerson::find()->where(['is_active' =>
    '1'])->orderBy(['position' => SORT_DESC])->all();
    $signPersonStatuses = null;

    if ($signPersons) {
        $signPersonStatuses =
ExamSheetSignService::getSignPersonStatuses($examSheetIds, $signPersons);
    }

    return $this->render('index', [
        'searchModel' => $searchModel,
        'dataProvider' => $dataProvider,
        'signPersonStatus' => $signPersonStatuses,
    ]);
}

public static function getSignPersonStatuses(array $examSheetIds, array
$signPersons): array
{
    $signPersonStatus = [];

    foreach ($examSheetIds as $examSheetId) {
        $signPersonStatus[$examSheetId] =
self::getSignPersonStatus($examSheetId, $signPersons);
    }

    return $signPersonStatus;
}

public static function getSignPersonStatus($examSheetId, $signPersons):
array
{
    $currentUserId = Yii::$app->user->id;
    $currentExamSheet = ExamSheet::findOne($examSheetId);
    $currentComponentType = EducationalComponentType::find()->where(['id' =>
    $currentExamSheet->educational_component_type_id])->one();

    if (!$currentComponentType->is_need_to_sign) {
        return [];
    }
}

```

		Джус І.О.			ІПЗ.КР.Б – 121 – 25 – ІЗ	Арк.
		Морозов А.В.				63
Змн.	Арк.	№ докум.	Підпис	Дата		

```

$persons = self::getPersonsForExamSheet($examSheetId);
$signPersonStatus = [];

foreach ($signPersons as $person) {
    $personStatuses = array_filter($persons, function ($p) use ($person)
    {
        return $p['role'] === $person->type;
    });

    $isSigned = false;

    if ($person->behavior === 'all') {
        $isSigned = !empty($personStatuses)
array_reduce($personStatuses, function ($carry, $p) {
    return $carry && $p['sign_status'];
}, true);
    } elseif ($person->behavior === 'one') {
        $isSigned = !empty($personStatuses)
array_reduce($personStatuses, function ($carry, $p) {
    return $carry || $p['sign_status'];
}, false);
    }

    $signByCurrentUser = ExamSheetSign::find()
        ->where([
            'person_id' => $currentUserId,
            'exam_sheet_id' => $examSheetId,
            'person_type' => $person->type
        ])
        ->one();

    $signPersonStatus[] = [
        'label' => $person->label,
        'status' => $isSigned,
        'is_sign_by_current_user' => $signByCurrentUser !== null,
        'sign_date' => $signByCurrentUser->datetime ?? null,
        'position' => $person->position,
    ];
}

usort($signPersonStatus, function ($a, $b) {
    return $a['position'] <=> $b['position'];
});

return $signPersonStatus;
}

$model = $this->findModel($id);
$model->date_end = $model->dateEnd;
$academicGroup = AcademicGroup::findOne(['id' => $model->group_id]);

$students = Student::find()
    ->select('education_id')
    ->where(['academic_group_id' => $academicGroup->id])
    ->column();

$collator = new Collator('uk UA');

```

		Джус І.О.			ІІЗ.КР.Б – 121 – 25 – ІІЗ	Арк.
		Морозов А.В.				64
Змн.	Арк.	№ докум.	Підпис	Дата		

```

usort($points, function ($a, $b) use ($collator) {
    return $collator->compare($a->student_name, $b->student_name);
});

$points_id = ExamSheetPoint::find()
->select('id')
->where(['exam_sheet_id' => $model->id])
->column();

$teachers = ExamSheetStaff::find()->select('staff_id')-
>andWhere(['exam_sheet_id' => $model->id])->column();
$expandedTeachers = Staff::find()->andWhere(['id' => $teachers])->all();

$restructuredData = [];
foreach ($points_id as $point_id) {
    $agreedPoints = ExamSheetPointAgreement::find()
        ->where(['point_id' => $point_id])
        ->all();

    foreach ($agreedPoints as $agreement) {
        $restructuredData[$point_id] = [
            'choice_type' => $agreement->choice_type,
            'datetime' => $agreement->datetime,
        ];
    }
}

$signPersons = ExamSheetSignService::getPersonsForExamSheet($model->id);
return $this->render('view', [
    'points' => $points,
    'teachers' => $expandedTeachers,
    'model' => $model,
    'academicGroup' => $academicGroup,
    'providerExamSheetPoint' => $providerExamSheetPoint,
    'providerExamSheetStaff' => $providerExamSheetStaff,
    'agreedPoints' => $restructuredData,
    'course' => $course,
    'signPersons' => $signPersons,
    'is_sign' => $isSign,
    'examSheetsWithSameParent' => $this->getExamSheetsWithSameParent($id),
    'isCanGenerate' => ExamSheetSummaryService::canGenerateAdditionalExamSheet($id),
]);

protected function updateLastPoint($event): void
{
    /** @var ExamSheetPoint $sender */
    $sender = $event->sender;

    $examSheetId = $sender->exam_sheet_id;
    $originalExamSheetId = AdditionalExamSheetGenerator::getOriginalExamSheetIdOf($examSheetId);
    $studentId = $sender->student_id;

    $lastExamSheetPoint = $this->initLastPoint($studentId, $examSheetId, $originalExamSheetId);

    $selector = new LastMarkSelector();

```

		Джус І.О.			ІІЗ.КР.Б – 121 – 25 – ІІЗ	Арк.
		Морозов А.В.				65
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    $point = $selector->getLastExamSheetPointOf($sender->student_id,
$originalExamSheetId);

    if ($point) {
        $lastExamSheetPoint->exam_sheet_id = $point->exam_sheet_id;
        $lastExamSheetPoint->points_part1 = $point->points_part1;
        $lastExamSheetPoint->points_part2 = $point->points_part2;
        $lastExamSheetPoint->points_final = $point->points_final;

        $lastExamSheetPoint->save(false);
    } else {
        $lastExamSheetPoint->delete();
    }
}

public function getLastExamSheetPointOf(int $studentId, int
$originalExamSheetId): ExamSheetPoint|null
{
    $childrenIds = AdditionalExamSheetGenerator::getExamSheetWithAncestorsIds($originalExamSheetId);
    $allExamSheetIds = array_merge([$originalExamSheetId], $childrenIds);

    $point = $this->getExamSheetPointOfStudent($studentId, $allExamSheetIds);

    if (is_null($point)) {
        $point = ExamSheetPoint::find()->where(["exam_sheet_id" =>
$allExamSheetIds, 'student_id' => $studentId])->one();
    }

    return $point;
}

public function getExamSheetPointOfStudent(int $studentId, array
$examSheetIds): ExamSheetPoint|null
{
    $query = ExamSheetPoint::find()
        ->andWhere(['exam_sheet_id' => $examSheetIds])
        ->andWhere(['student_id' => $studentId])
        ->andWhere(
            ['not', ['points_final' => null]],
        )
        ->orderBy(['exam_sheet_id' => SORT_DESC]);
    return $query->one();
}

private function validateAndImport(string $filePath)
{
    Yii::$app->response->format = Response::FORMAT_JSON;

    $linker = new ExcelExamSheetPointsDoubleLinker();
    $errors = $linker->validate($filePath);
    $isEmpty = $this->areAllInnerArraysEmpty($errors);

    Yii::$app->response->format = Response::FORMAT_JSON;
    if (count($errors) > 0 && !$isEmpty) {

        $sheetNames = ExcelParser::getSheetNames($filePath);
    }
}

```

		Джус І.О.			ІІЗ.КР.Б – 121 – 25 – ІІЗ	Арк.
		Морозов А.В.				66
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        return ['errors' => $errors, 'sheetNames' => $sheetNames];
    }

    $linker->import($filePath);

    return [
        'success' => true,
        'message' => "Оцінки успішно імпортовано",
    ];
}

private function getDisciplineValidationRules(string $filePath, int $page,
string $letter): ExcelParser
{
    $parser = new ExcelParser($filePath, $page);

    $parser
        ->addRule($letter . "5", "subject_name", new EmptyExcelValidator(),
ExcelErrorType::ERROR)
        ->addRule($letter . "6", "count_credits", new NumberRangeValidator(0,
240, false), ExcelErrorType::ERROR)
        ->addRule($letter . "7", "form_control", new FormControlValidator(),
ExcelErrorType::ERROR)
        ->addRule($letter . "8", "education_component_type", new
EducationalComponentTypeNameOrShortNameValidator(), ExcelErrorType::ERROR)
        ->addRule($letter . "9", "control_date", new DateValidator(),
ExcelErrorType::ERROR)
        ->addRule($letter . "10", "number_information", new
EmptyExcelValidator(), ExcelErrorType::ERROR)
        ->addRule($letter . "11", "teacher", new EmptyExcelValidator(),
ExcelErrorType::ERROR);

    return $parser;
}

public function clearErrors(): void
{
    $this->errors = [];
}

protected function addError(string $error): void
{
    $this->errors[] = $error;
}

class EducationalComponentTypeNameOrShortNameValidator extends
ExcelValidator
{
    /**
     * @param $value
     * @return void
     */
    public function validate($value): void
    {
        $isExists = EducationalComponentType::find()
            ->where(['or',
                ['name' => $value],
                ['short_name' => $value]
            ])
    }
}

```

		Джус І.О.			ІІЗ.КР.Б – 121 – 25 – ІІЗ	Арк.
		Морозов А.В.				67
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        ->exists();

        if (!$isExists) {
            $this->addError("Неправильно вказано тип освітнього компонента");
        }
    }
}

public function addRule(string $cell, string $fieldName, ExcelValidator
$validator, int $errorType): ExcelParser
{
    $this->rules[] = [
        'cell' => $cell,
        'field' => $fieldName,
        'validator' => $validator,
        'errorType' => $errorType,
    ];
    return $this;
}

private function validateCell(array $rule): void
{
    $cellValue = $this->getCellString($rule['cell']);

    /** @var $validator ExcelValidator */
    $validator = $rule['validator'];
    $validator->validate($cellValue);

    $errors = $validator->getErrors();

    if (!empty($errors)) {
        $uniqueErrors = array_unique($errors);

        foreach ($uniqueErrors as $error) {
            $this->addError($rule['cell'], $error, $rule['errorType']);
        }
    }

    $groupRules = $this->getGroupValidationRules($filePath, $page);
    $studentRules = $this->getStudentValidationRules($filePath, $page,
    $groupId);
    $disciplineRules = [];
    $pointsRules = [];

    foreach ($lettersArr as $letter) {
        $disciplineRules[] = $this->getDisciplineValidationRules($filePath,
        $page, $letter);
        $pointsRules[] = $this->getPointsValidationRules($filePath, $page,
        $letter);
    }

    $groupRulesErrors = $groupRules->validate()->getErrors();

    $studentRulesErrors = $studentRules->validate()->getErrors();
    for ($i = 0; $i < count($disciplineRulesArr); $i++) {

```

		Джус І.О.			ПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				68
Змн.	Арк.	№ докум.	Підпис	Дата		

```

$disciplineRule = $disciplineRulesArr[$i];
$pointsRule = $pointsRulesArr[$i];

$disciplineErrors = $disciplineRule->validate()->getErrors();
$pointsErrors = $pointsRule->validate()->getErrors();

    $allDisciplinesErrors = array_merge($allDisciplinesErrors,
$disciplineErrors);
    $allPointsErrors = array_merge($allPointsErrors, $pointsErrors);
}

return array_merge($groupRulesErrors, $studentRulesErrors,
$allDisciplinesErrors, $allPointsErrors);

if (count($errors) > 0 && !$isEmpty) {

    $sheetNames = ExcelParser::getSheetNames($filePath);
    return ['errors' => $errors, 'sheetNames' => $sheetNames];
}

$linker->import($filePath);

    $rulesArr = $this->getExamSheetPointsValidationRules($filePath,
$page);
    /** @var ExcelParser $groupRules */
    $groupRules = $rulesArr['group_rules'];
    $studentRules = $rulesArr['student_rules'];

    if ($groupRules->isEmpty()) {
        return [];
    }

    $groupRulesData = $groupRules->parse();
    $studentRulesData = $studentRules->parse();

public function parseDataCells(array $cellRules): array
{
    $rowData = [];

    foreach ($cellRules as $rule) {
        $cellValue = $this->getCellString($rule['cell']);

        $rowData[$rule['field']] = $cellValue;
    }

    return $rowData;
}

$existingExamSheet = $this->findExistingExamSheet($data, $i);

if ($existingExamSheet) {
    return $existingExamSheet;
}

$semester = $this->calculateSemester(

```

		Джус І.О.			ІІЗ.КР.Б – 121 – 25 – ІІЗ	Арк.
		Морозов А.В.				69
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        $this->getGroupDataValue($dataArray, 'semester'),
        $this->getGroupDataValue($dataArray, 'study_year')
    );

    $examSheet->dictionary_semester_id = $semester->id;
    $examSheet->educational_component_id = $this->getEducationComponent(
        $this->getDisciplineDataValue($dataArray, $i, 'subject_name')
    );

    $examSheet->form_of_control_id = $this->getFormControlId(
        $this->getDisciplineDataValue($dataArray, $i, 'form_control')
    );
    $examSheet->registration_date = $this->getDate($this->
        >getDisciplineDataValue($dataArray, $i, 'control_date'))->format('Y-m-d');
    $examSheet->sheet_number = trim($this->getDisciplineDataValue($dataArray,
        $i, 'number_information')) ? "-";
    $examSheet->control_date = $this->getDate($this->
        >getDisciplineDataValue($dataArray, $i, 'control_date'))->format('Y-m-d');
    $examSheet->control_time = null;
    $examSheet->days_open = 0;

    $academicGroup = AcademicGroup::findOne(["name" => $this->
        >getGroupDataValue($dataArray, 'group_name')]);
    $examSheet->group_id = $academicGroup->id;
    $examSheet->exam_sheet_type_id = ExamSheetType::find()->select('id')-
        >where(['short_name' => "Основна"])->scalar();

    $year = $semester->academicYear;
    $course = DateParserService::calculateGroupCourse($academicGroup-
        >study_start, $year->year_start);

    $examSheet->course = $course;
    $examSheet->queue = 1;
    $examSheet->save(false);

    public function importPointsForDiscipline(array $students, ExamSheet
    $examSheet, array $points, Staff $staff)
    {
        for ($i = 0; $i < count($students); $i++) {
            $man = $students[$i];
            $mark = $points[$i];

            $point = ExamSheetPoint::findOne(['exam_sheet_id' => $examSheet-
            >id, 'student_id' => $man]);

            if (!$point) {
                $point = new ExamSheetPoint();
                $point->exam_sheet_id = $examSheet->id;
                $point->student_id = $man->education_id;
                $point->student_name = $man->person_fio;
                $point->staff_id = $staff->id;

                $point->points_final = $mark;

                if (is_null($mark)) {
                    $point->staff_name = null;
                    $point->datetime = null;
                } else {
                    $point->staff_name = $staff->full_name;

```

		Джус І.О.			ІІЗ.КР.Б – 121 – 25 – ІІЗ	Арк.
		Морозов А.В.				70
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        $point->datetime = $examSheet->control_date;
    }

    $point->save(false);
}

}

Yii::$app->response->format = Response::FORMAT_JSON;
$lastMarkSelector = new LastMarkSelector();

$lastPoints = $lastMarkSelector->
>getAllLastExamSheetPointsOfStudent($educationId);

return [
    'status' => 'success',
    'data' => $result
];

public function actionAgreePoint(): Response
{
    $data = Yii::$app->request->getRawBody();
    $data = json_decode($data, true);

    $user = User::findOne(['id' => Yii::$app->user->identity->getId()]);
    $personId = $user->person_id;

    if (isset($data['point_id']) && isset($data['choice_type'])) {
        $agreement = new ExamSheetPointAgreement();
        $agreement->point_id = $data['point_id'];
        $agreement->choice_type = $data['choice_type'];
        $datetime = new DateTime('now', new DateTimeZone('Europe/Kiev'));
        $agreement->datetime = $datetime->format('Y-m-d H:i:s');
        $point = ExamSheetPoint::findOne($data['point_id']);
        $students = ActiveStudent::find()->where(['person_id' => $personId])-
>all();
        if ($point && in_array($point->student_id, array_column($students,
'education_id'))) {
            if ($agreement->save()) {
                return $this->asJson([
                    'status' => 'success',
                    'message' => 'Agreement saved successfully.'
                ]);
            }
        }
        return $this->asJson([
            'status' => 'error',
            'message' => 'Invalid data.'
        ]);
    }

    if ($agreement->delete()) {
        return $this->asJson([
            'status' => 'success',
            'message' => 'Agreement canceled successfully.'
        ]);
    }
}

```

		Джус І.О.			ІПЗ.КР.Б – 121 – 25 – ІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		71