

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

ПОЯСНЮВАЛЬНА ЗАПИСКА

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

на тему:

**«Інформаційна система онлайн-менеджменту
освітніх процесів Digital University UA: модулі
авторизації та логування»**

Виконав студент групи ІПЗ-22-4
ЗУБИК Максим Романович

Керівник роботи:
МОРОЗОВ Андрій Васильович

Рецензент:
БАКАЛЮК Тетяна Анатоліївна

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
«28» лютого 2026 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **ЗУБИК Максим Романович**

Керівник роботи: **МОРОЗОВ Андрій Васильович**

Тема роботи: **«Інформаційна система онлайн-менеджменту освітніх процесів
Digital University UA: модулі авторизації та логування»**

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

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

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

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Морозов А.В.	01.02.2026р.	01.02.2026р.
2	Морозов А.В.	02.04.2026р.	02.04.2026р.
3	Морозов А.В.	04.05.2026р.	04.05.2026р.

РЕФЕРАТ

Кваліфікаційна робота складається з програмного комплексу для інформаційної системи онлайн-менеджменту освітніх процесів Digital University UA, зокрема модуля «Авторизація» та «Логування» та пояснювальної записки. Пояснювальна записка до випускної роботи містить 95 сторінок, 51 ілюстрацію та 1 таблицю.

Головною метою є створення модулю інформаційної системи, який дозволить користувачам авторизуватися різними способами у системі та логувати помилки, що трапляються у користувачів під час використання продукту.

У процесі виконання роботи було сформульовано ключові завдання розробки модуля, здійснено аналіз існуючих аналогів інформаційних систем і обрано технології, які забезпечують реалізацію необхідних вимог до програмного забезпечення.

КЛЮЧОВІ СЛОВА: BACKEND, ІНФОРМАЦІЙНА СИСТЕМА, АВТОРИЗАЦІЯ, PHP, Yii2, JS, LDAP, GOOGLE CLOUD, AZURE, OFFICE 365, OAUTH2, EMAIL.

					ІПЗ.КР.Б – 121 – 26 – ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	Інформаційна система онлайн-менеджменту освітніх процесів Digital University UA: модулі авторизації та логування	Літ.	Арк.	Аркушів
Розроб.		Зубик М.Р.						
Керівник		Морозов А. В.					3	95
Н. контр.						Житомирська політехніка, група ІПЗ-22-4		
Зав. каф.		Вакалюк Т.А.						

ABSTRACT

The qualification work consists of a software complex for the information system of online management of educational processes Digital University UA, in particular the Authorization and Logging modules, as well as an explanatory note. The explanatory note of the graduation work contains 95 pages, 51 illustrations and 1 table.

The main objective is to develop a module of the information system that will allow users to authenticate in the system using various methods and to log errors that occur during product usage.

In the course of the work, the key tasks for module development were defined, an analysis of existing analogues of information systems was conducted, and technologies were selected that ensure the implementation of the required software requirements.

KEYWORDS: BACKEND, INFORMATION SYSTEM, AUTHORIZATION, PHP, YII2, JS, LDAP, GOOGLE CLOUD, AZURE, OFFICE 365, OAUTH2, EMAIL.

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

РЕФЕРАТ	3
ABSTRACT	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	6
ВСТУП.....	7
РОЗДІЛ 1. АНАЛІЗ ВИМОГ КОРИСТУВАЧА ТА КОНЦЕПТУАЛЬНЕ МОДЕЛЮВАННЯ	10
1.1. Постановка задачі	10
1.2. Аналіз аналогів програмного продукту	12
1.3. Вибір архітектури системи	17
1.4. Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення	20
Висновки до першого розділу	23
РОЗДІЛ 2. ПРОЕКТУВАННЯ СИСТЕМИ ОНЛАЙН-МЕНЕДЖМЕНТУ ОСВІТНІХ ПРОЦЕСІВ	25
2.1. Визначення варіантів використання та структури системи	25
2.2. Обґрунтування вибору системи управління базами даних	28
2.3. Розробка бази даних системи	31
2.4. Реалізація функціоналу системи управління навчальним процесом	35
Висновки до другого розділу	47
РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З СИСТЕМОЮ ОНЛАЙН- МЕНЕДЖМЕНТУ ОСВІТНІХ ПРОЦЕСІВ	49
3.1. Порядок встановлення та налаштування параметрів додатку	49
3.2. Структура інтерфейсу додатку	54
3.3. Тестування роботи розробленого додатку	74
Висновки до третього розділу	77
ВИСНОВКИ	78
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	80
ДОДАТКИ	83
Додаток А	84

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД – база даних

ЗВО – заклад вищої освіти

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

ВСТУП

Інтенсивний розвиток цифрових технологій ставить перед закладами вищої освіти нові завдання, які істотно змінюють підходи до організації освітнього процесу. Традиційні методи управління, зокрема використання паперової документації, втрачають ефективність через складність опрацювання значних обсягів даних, високу ймовірність людських помилок та обмежений доступ до інформації поза межами навчального закладу. Окрім цього, поширення дистанційного та змішаного форматів навчання вимагає впровадження сучасних технологічних рішень, здатних адаптувати освітні процеси до нових умов, забезпечуючи безперервність навчання та оперативну взаємодію між учасниками освітнього процесу.

Додатково слід враховувати зростання обсягів освітніх даних, які потребують систематизації, структурованого зберігання та аналітичної обробки для прийняття управлінських рішень. У таких умовах автоматизовані інформаційні системи стають ключовим інструментом цифрової трансформації закладів вищої освіти, оскільки дозволяють підвищити ефективність адміністрування, забезпечити прозорість процесів контролю успішності та оптимізувати комунікацію між студентами, викладачами й адміністрацією. Впровадження подібних систем також сприяє зменшенню навантаження на адміністративний персонал і мінімізації ризику помилок при обробці даних.

У зв'язку з цим зростає актуальність використання автоматизованих систем управління навчальною діяльністю, які забезпечують підвищення ефективності, прозорості та оптимізації освітнього процесу.

Метою дослідження є створення цифрової системи для управління освітнім процесом, яка дозволить автоматизувати адміністративні процедури, полегшити взаємодію між студентами, викладачами та адміністрацією, а також забезпечити зручні механізми моніторингу та аналізу навчальної діяльності. Розроблювана система має сприяти підвищенню ефективності управління освітніми процесами шляхом централізованого зберігання та обробки інформації, скорочення часу на виконання рутинних операцій та зменшення

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		7

кількості помилок, пов'язаних із ручним введенням і обробкою даних. Крім того, система повинна забезпечувати безпечний доступ до інформаційних ресурсів, підтримувати сучасні механізми автентифікації та створювати умови для подальшого розвитку й масштабування функціональних можливостей відповідно до потреб закладу вищої освіти.

Завдання, що потрібно виконати для досягнення поставленої мети, полягають у проведенні аналізу предметної області та існуючих рішень для організації авторизації користувачів і логування подій, визначенні функціональних та нефункціональних вимог до розроблюваних модулів, обґрунтуванні вибору архітектури програмного забезпечення, інструментальних засобів розробки та системи керування базами даних. Також необхідно спроектувати структуру бази даних для забезпечення роботи модулів авторизації та логування, реалізувати підтримку авторизації через LDAP, Office 365 та Google OAuth2, а також механізм автоматичного створення облікових записів користувачів. Окремим завданням є розробка системи централізованого логування помилок і подій.

Об'єктом дослідження є процес створення програмного забезпечення для автоматизованого управління освітнім процесом, що охоплює планування, організацію, контроль та моніторинг навчальної діяльності в закладах вищої освіти. У межах дослідження розглядаються особливості побудови інформаційних систем освітнього призначення, організація взаємодії між їхніми компонентами, а також методи забезпечення надійного зберігання та обробки даних. Особлива увага приділяється питанням автоматизації процесів, пов'язаних із керуванням користувачами, наданням доступу до інформаційних ресурсів та фіксацією системних подій.

Предметом дослідження є технології, алгоритмічні рішення та програмні засоби, що застосовуються під час розробки цифрових систем управління освітнім процесом, а також їх адаптація до вимог сучасних ЗВО. Предмет дослідження охоплює методи реалізації модулів авторизації та логування, підходи до проектування архітектури веб-застосунків, засоби роботи з базами

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				8
Змн.	Арк.	№ докум.	Підпис	Дата		

даних, механізми інтеграції із зовнішніми сервісами автентифікації та інструменти забезпечення інформаційної безпеки. Також розглядаються технології контейнеризації, адміністрування серверної інфраструктури та способи підвищення продуктивності й надійності програмного забезпечення.

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		9

РОЗДІЛ 1. АНАЛІЗ ВИМОГ КОРИСТУВАЧА ТА КОНЦЕПТУАЛЬНЕ МОДЕЛЮВАННЯ

1.1. Постановка задачі

Сучасні інформаційні системи закладів вищої освіти вимагають надійних та безпечних механізмів доступу до даних, а також ефективних засобів контролю стабільності роботи програмного забезпечення. Зі зростанням кількості користувачів, зокрема вступників, студентів, викладачів та адміністративного персоналу, актуальною стає потреба у впровадженні гнучких методів авторизації, які забезпечують зручність використання, високий рівень безпеки та інтеграцію з зовнішніми сервісами.

Багато освітніх установ досі використовують застарілі або обмежені механізми входу до систем, що ґрунтуються лише на локальних облікових записах. Такий підхід ускладнює адміністрування, створює додаткове навантаження на технічний персонал та не відповідає сучасним вимогам кібербезпеки. Крім того, відсутність централізованого логування подій і помилок значно ускладнює виявлення несправностей, аналіз інцидентів та забезпечення стабільної роботи системи.

Для вирішення зазначених проблем доцільним є впровадження модулів авторизації та логування в інформаційній системі «Цифровий університет». Модуль авторизації повинен підтримувати декілька способів автентифікації користувачів, зокрема через LDAP, Office 365, Google OAuth2, а також шляхом підтвердження входу через електронну пошту. Такий підхід дозволяє інтегрувати систему з корпоративними та хмарними сервісами, підвищуючи зручність доступу для різних категорій користувачів.

Окрему увагу приділено автоматичному створенню облікових записів для вступників і студентів, що дає змогу мінімізувати ручну роботу адміністраторів та забезпечити коректне управління користувачами в системі. Це особливо важливо під час вступної кампанії та на початку навчального року, коли кількість

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

нових користувачів значно зростає. Також створення нових акаунтів для персоналу є важливим кроком до спрощення робочого процесу.

Не менш важливим є модуль логування, який забезпечує фіксацію помилок та подій, що виникають під час роботи системи. Логування реалізується шляхом перехоплення винятків та критичних помилок за допомогою механізмів, що дозволяють виявляти як оброблені, так і фатальні помилки. Усі помилки зберігаються у БД та файлах з можливістю подальшого перегляду та аналізу.

Впровадження зазначених модулів у складі інформаційної системи «Цифровий університет» сприятиме підвищенню рівня безпеки, стабільності та керованості системи, а також забезпечить зручний і сучасний доступ користувачів до освітніх сервісів.

Основними завданнями розроблюваної системи є реалізація модуля авторизації та модуля логування, які забезпечують безпечний доступ користувачів до інформаційної системи й контроль стабільності її роботи. Модуль авторизації повинен підтримувати декілька способів входу, зокрема автентифікацію через LDAP, Office 365 та Google OAuth2. Такий підхід дає можливість використовувати як корпоративні облікові записи університету, так і зовнішні сервіси автентифікації, що підвищує зручність роботи користувачів і спрощує адміністрування доступу.

Окремим завданням модуля авторизації є автоматичне створення облікових записів для вступників і студентів, а також надсилання користувачам повідомлень з необхідними даними для першого входу до системи. Це дозволяє зменшити кількість ручних операцій під час масового підключення користувачів до платформи та мінімізувати ймовірність помилок під час формування логінів і паролів.

Модуль логування має забезпечувати перехоплення винятків, критичних помилок і важливих системних подій, що виникають під час роботи користувачів із платформою. Збереження такої інформації у базі даних і файлах створює основу для подальшого аналізу помилок, швидкого пошуку причин збоїв та підвищення надійності програмного забезпечення. Завдяки цьому

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

адміністратори й розробники отримують можливість не лише реагувати на вже виявлені проблеми, а й запобігати їх повторному виникненню.

1.2. Аналіз аналогів програмного продукту

У відповідь на зазначені виклики в освітній сфері значна кількість компаній зосередила свої зусилля на створенні власних цифрових продуктів, орієнтованих на підвищення ефективності управління навчальним процесом. У результаті аналізу було ідентифіковано кілька подібних сервісів, що спеціалізуються на підтримці діяльності закладів вищої освіти.

Moodle є системою управління навчанням із відкритим вихідним кодом, розробленою для забезпечення дистанційної освіти. Платформа надає широкий набір інструментів, серед яких створення та адміністрування навчальних курсів, розробка тестових завдань, оцінювання академічних результатів студентів і забезпечення інтерактивної взаємодії між учасниками освітнього процесу. Ключовою перевагою Moodle є її висока адаптивність, що дозволяє налаштовувати систему відповідно до специфічних потреб конкретного навчального закладу завдяки використанню численних модулів і плагінів.

Платформа підтримує інтеграцію з поширеними сервісами, зокрема Zoom, Google Workspace та Microsoft Teams, що значно полегшує організацію онлайн-занять. Крім того, Moodle відповідає міжнародним стандартам електронного навчання, таким як SCORM та xAPI, забезпечуючи сумісність із іншими освітніми платформами й цифровими ресурсами.

Наявність великої спільноти користувачів сприяє формуванню розвиненої екосистеми підтримки, що спрощує процес упровадження, налаштування та супроводу системи. Важливою перевагою є також безкоштовне розповсюдження платформи, що робить її доступною для невеликих навчальних установ (рис. 1.1).

Офіційний сайт: <https://moodle.org/>

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

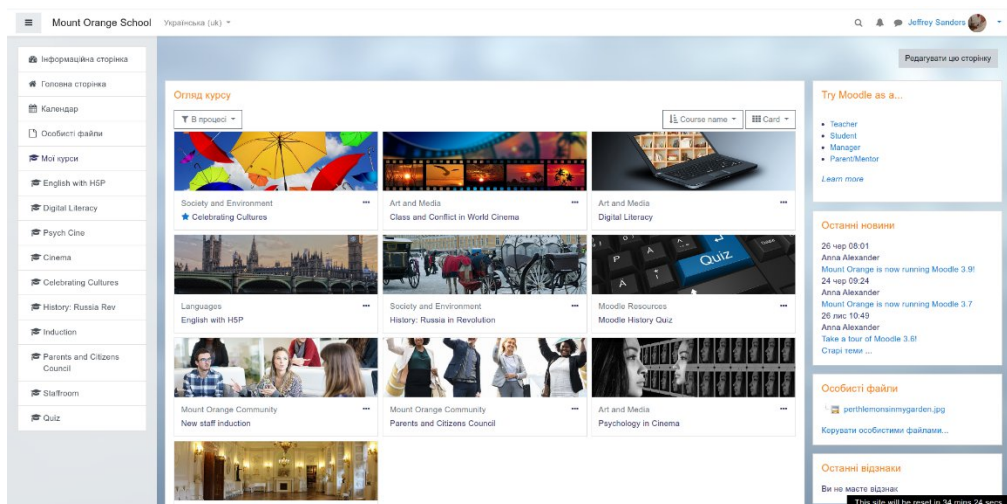


Рисунок 1.1. Платформа Moodle

До основних переваг Moodle належать відкритий програмний код, який дозволяє розробляти власні розширення й модулі, а також активна технічна спільнота, що забезпечує оперативне вирішення більшості технічних питань. Система характеризується високою гнучкістю та можливістю адаптації до індивідуальних вимог навчального закладу. Вона підтримує інтеграцію з популярними засобами відеозв'язку та колективної роботи, такими як Zoom, Google Workspace і Microsoft Teams. Окрім цього, Moodle є мультиплатформенною системою, доступною як на персональних комп'ютерах, так і на мобільних пристроях, що сприяє безперервності освітнього процесу.

Водночас платформа має низку недоліків. Зокрема, її налаштування та адміністрування вимагають достатньо високого рівня технічної підготовки. За відсутності оптимізації серверної інфраструктури складно забезпечити стабільну роботу системи при значній кількості користувачів або великих обсягах даних. Інтерфейс Moodle може бути складним для освоєння новими користувачами, а функціональні можливості управління кадровими та фінансовими процесами в системі обмежені, що змушує заклади освіти використовувати додаткові програмні рішення. Крім того, платформа потребує регулярного оновлення та технічного супроводу, що може бути проблематичним за умов обмежених ресурсів.

Отже, незважаючи на те, що Moodle є потужним засобом організації освітньої діяльності, її впровадження потребує значних часових і технічних

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

ресурсів. Доцільність використання цієї платформи повинна визначатися з урахуванням реальних потреб навчального закладу та наявних можливостей для забезпечення її стабільної підтримки.

Canvas LMS - сучасна система управління навчанням, розроблена компанією Instructure. Платформа застосовується в університетах, коледжах та інших навчальних закладах для організації онлайн-навчання. Вона побудована на хмарній архітектурі, що забезпечує високу продуктивність та доступність із будь-якого пристрою. Canvas LMS підтримує інтеграцію з численними сервісами, зокрема Google Workspace, Microsoft 365, Zoom, а також надає відкритий API для розширення функціональних можливостей.

Система характеризується інтуїтивно зрозумілим інтерфейсом, що полегшує користування як студентам, так і викладачам. Canvas LMS дозволяє створювати навчальні курси, тестові завдання, оцінювати успішність, вести форуми для обговорень та автоматично зберігати зміни в режимі реального часу. Для зручності користувачів доступний мобільний додаток, що забезпечує швидкий доступ до навчальних матеріалів (рис. 1.2).

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

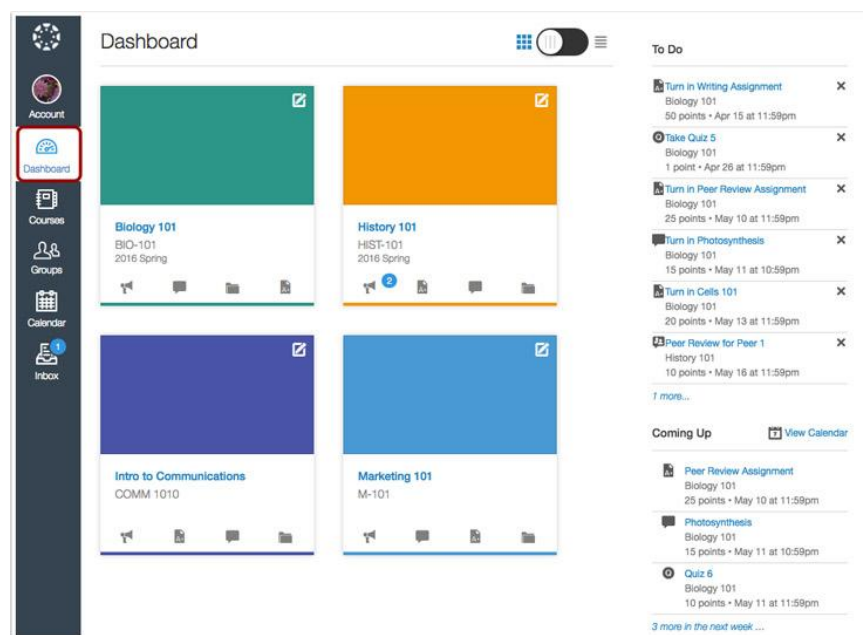


Рисунок 1.2. Система Canvas LMS

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

Хмарна структура Canvas LMS дає змогу навчальним закладам зменшити витрати на підтримку інфраструктури та отримати стабільну технічну підтримку. Платформа забезпечує широкі можливості інтеграції з іншими освітніми сервісами та сторонніми інструментами, а також дозволяє гнучко налаштовувати ролі користувачів і рівні доступу відповідно до потреб конкретного закладу або викладача.

До недоліків Canvas LMS належить висока вартість ліцензії, що може стати серйозною перешкодою для окремих навчальних закладів. Деякі розширені функції доступні лише у платних версіях, а частина інтеграцій потребує додаткового налаштування із залученням фахівців. Також відсутня повноцінна підтримка локального розгортання на серверах закладу, що може бути критично важливим у випадках підвищених вимог до захисту персональних даних або роботи без підключення до інтернету.

Таким чином, Canvas LMS є потужним інструментом для організації дистанційного навчання з багатьма сучасними функціями, але її впровадження слід ретельно оцінювати з урахуванням фінансових, технічних та безпекових аспектів діяльності конкретного освітнього закладу.

OpenEduCat - це модульна система управління навчанням із відкритим кодом, створена на основі ERP-платформи Odoo. Вона призначена для університетів, шкіл та інших навчальних закладів і надає комплексні інструменти для адміністрування студентів, курсів, фінансових операцій та інших аспектів освітнього процесу.

Система забезпечує ведення електронного журналу, формування розкладів занять, автоматизацію управління бібліотекою, фінансами та інфраструктурою закладу. OpenEduCat також підтримує аналітику та формування звітності, що сприяє обґрунтованому прийняттю рішень адміністрацією. Оскільки система інтегрована з платформою Odoo, вона легко поєднується з іншими модулями для бухгалтерського обліку, CRM та управління бізнес-процесами (рис. 1.3).

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

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				15
Змн.	Арк.	№ докум.	Підпис	Дата		

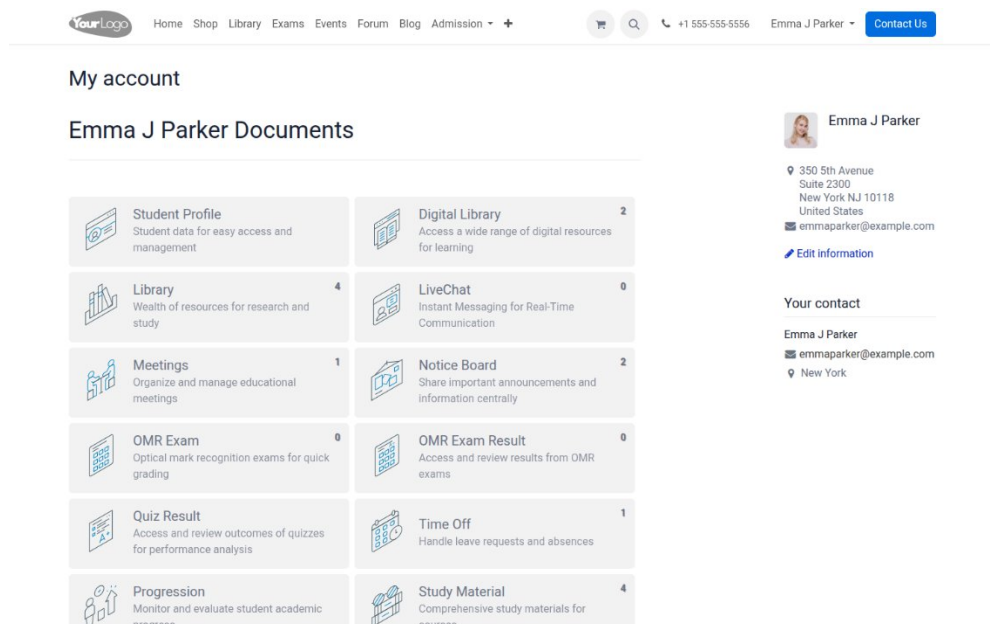


Рисунок 1.3. Система OpenEduCat

Серед основних переваг OpenEduCat - відкритий код та можливість адаптації системи під специфічні потреби навчального закладу, гнучке налаштування як у хмарному середовищі, так і на локальних серверах, а також автоматизація численних адміністративних процесів. Система також вирізняється потужними аналітичними інструментами, що дозволяють ефективно відстежувати динаміку освітнього процесу та оперативно реагувати на зміни.

Водночас використання OpenEduCat може бути складним для користувачів без достатньої технічної підготовки. Початкове налаштування системи вимагає спеціалізованих знань, а деякі функціональні модулі доступні лише у платній версії, що може обмежувати її застосування в закладах із обмеженим бюджетом. Додатковою складністю є недостатня кількість детальної документації та обмежена підтримка від розробників, що ускладнює інтеграцію з іншими цифровими системами.

Отже, OpenEduCat є потужним інструментом для цифровізації управління навчальним закладом, проте її ефективне впровадження потребує відповідної технічної підготовки, чіткого планування та належного ресурсного забезпечення.

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

Таблиця порівняння аналогів

Критерії порівняння	Moodle	Canvas LMS	OpenEduCat
Тип архітектури	Моноліт (PHP), on-premise + cloud, плагіни	SaaS (хмарна, сервіс-орієнтована, multi-tenant)	Модульна ERP (на базі Odoo), on-premise + cloud
Основне призначення	LMS для організації навчання	Cloud LMS з акцентом на UX та онлайн-навчання	LMS + ERP для повного управління закладом
Ліцензійна політика	Open Source (GPL), безкоштовна	Комерційна підписка (SaaS), частково open core	Open Source + платні модулі (open core)
LDAP / Active Directory	вбудовано	через SSO/SAML	через Odoo / модулі
Google (OAuth / Google Workspace)	плагіни	нативно	через OAuth модулі
Microsoft (Office 365 / Azure AD)	плагіни	нативно	через OAuth/SAML
Логи помилок (errors/exceptions)	+	+	+
Логи дій користувачів	+(детально)	+	+
Рівні логуювання (debug/info/warn/error)	+	+	+
Моніторинг у реальному часі	-	+	-
Експорт логів	+	+	+

За результатами аналізу аналогів було сформовано порівняльну таблицю, яка узагальнює основні характеристики розглянутих систем і дозволяє визначити їхні переваги та обмеження в контексті розробки модулів авторизації та логуювання.

1.3. Вибір архітектури системи

У сучасних інформаційних системах закладів вищої освіти особливе значення набуває правильне побудування архітектури програмного забезпечення, що дозволяє забезпечити стабільність, безпеку та масштабованість системи. Зростаюча кількість користувачів, включно з вступниками, студентами, викладачами та адміністративним персоналом, робить важливим ефективне управління кодовою базою та логікою взаємодії між компонентами системи, щоб спростити подальший розвиток та підтримку програмного продукту.

Традиційним підходом є монолітна архітектура, де всі компоненти системи - серверна логіка, база даних, модулі авторизації та логуювання - об'єднані в єдину структуру. Такий підхід забезпечує простоту розробки та швидке розгортання, особливо для невеликих проектів або систем із обмеженим функціоналом. Проте

із зростанням кількості функцій, способів авторизації та обсягу логів монолітна структура може стати громіздкою, ускладнюючи оновлення модулів, масштабування та впровадження нових функцій.

Монолітна архітектура має переваги завдяки простоті впровадження та розгортання, оскільки всі компоненти системи інтегровані в єдине середовище. Це робить процеси розробки, налагодження та тестування більш зрозумілими, а систему - зручною для моніторингу. Крім того, тісна взаємодія елементів забезпечує високу продуктивність, оскільки відсутні затримки, властиві розподіленим системам, а єдиний контроль над компонентами спрощує управління версіями та оновленнями. Водночас із ростом складності програмного продукту монолітна структура може стати громіздкою, що ускладнює її подальше оновлення та супровід. Зі збільшенням кількості функцій зростає ймовірність конфліктів між модулями, а зміни в одній частині системи можуть мати небажані наслідки для інших компонентів. Це робить розробку нових функцій складнішою, а масштабування системи є проблематичним. Крім того, навіть незначні зміни потребують ретельного тестування всього продукту, що може уповільнювати швидкість розробки.

На зміну монолітній архітектурі у сучасних інформаційних системах приходить мікросервісна архітектура, яка передбачає поділ програми на окремі незалежні сервіси, кожен з яких відповідає за конкретну функціональність. У системі «Цифровий університет» це дозволяє виділити окремі сервіси для авторизації, логування, управління користувачами та генерації облікових записів. Сервіси взаємодіють через чітко визначені API, що підвищує гнучкість системи та її масштабованість. Завдяки такому підходу розробники можуть працювати над різними модулями паралельно, скорочуючи час впровадження нових функцій і знижуючи ризик конфліктів у кодовій базі.

Однією з основних переваг мікросервісної архітектури є можливість незалежного масштабування кожного сервісу, що особливо важливо для модулів логування та авторизації, які обробляють значну кількість запитів одночасних користувачів. Крім того, окремі сервіси можна оновлювати або замінювати без

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				18
Змн.	Арк.	№ докум.	Підпис	Дата		

впливу на роботу всієї системи, що спрощує підтримку та модернізацію. Додатковою перевагою є можливість застосовувати різні технології та мови програмування для окремих мікросервісів, підбираючи оптимальні інструменти для конкретних завдань.

Водночас мікросервісна архітектура має і певні виклики. Вона потребує ретельного планування взаємодії між сервісами, налаштування мережових з'єднань і систем моніторингу. Розподіленість компонентів може призводити до затримок у комунікації, що впливає на продуктивність, а також потребує додаткових ресурсів і часу на реалізацію, що може бути обмеженням для невеликих проектів. Окрім цього, виникають складнощі з управлінням розподіленими транзакціями та забезпеченням безпеки обміну даними між сервісами.

Вибір архітектури системи залежить від її специфіки, масштабу та способу взаємодії компонентів. Для розроблюваної інформаційної системи «Цифровий університет», яка реалізує модулі авторизації та логування, оптимальним рішенням було обрано монолітну архітектуру. Основна причина полягає в тому, що більшість компонентів взаємодіє безпосередньо всередині одного середовища PHP/Yii2: логування здійснюється через обробку винятків і збереження у БД або файли, а модуль авторизації працює з зовнішніми сервісами (Google Cloud, Office 365, LDAP) через callback або API-запити. Монолітна архітектура дозволяє централізовано контролювати всі ці взаємодії, спрощує управління версіями та оновленнями, а також робить процес розробки та налагодження більш прозорим. В умовах використання хмарних сервісів та зовнішніх систем аутентифікації немає потреби розбивати систему на окремі мікросервіси, оскільки більшість інтеграцій відбувається через стандартні API і callback, а не через постійний обмін даними між внутрішніми сервісами.

Додатково слід зазначити, що використання контейнеризації на основі Docker підсилює переваги обраної архітектури. Розгортання монолітного застосунку в контейнері дозволяє стандартизувати середовище виконання, усунути проблему «працює на моїй машині» та забезпечити однакову поведінку

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				19
Змн.	Арк.	№ докум.	Підпис	Дата		

системи на різних етапах життєвого циклу програмного продукту — від розробки до продуктивної експлуатації. Це також спрощує масштабування системи шляхом запуску декількох екземплярів контейнера під навантаженням. Важливим аспектом є також можливість інтеграції системи з зовнішніми сервісами та API, що характерно для сучасних освітніх платформ. Завдяки використанню стандартизованих протоколів обміну даними (REST, OAuth2) система може легко взаємодіяти з хмарними сервісами автентифікації, електронною поштою та іншими корпоративними інструментами без необхідності зміни внутрішньої архітектури.

Крім того, обрана архітектура забезпечує поступову еволюцію системи. У разі зростання навантаження або появи нових функціональних вимог окремі компоненти, такі як аналітика, нотифікації або розширене логування, можуть бути винесені в окремі сервіси без повної перебудови основного застосунку. Таким чином, система зберігає гнучкість розвитку та адаптивність до майбутніх вимог без втрати стабільності.

1.4. Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення

Наступним етапом стало визначення інструментальних засобів, які забезпечують реалізацію функціональних вимог, стабільну роботу системи та можливість її подальшого супроводу. Вибір упав на мову PHP, яка зарекомендувала себе як надійне рішення у веб-розробці: велика кількість навчальних матеріалів, активна спільнота та широкий набір готових фреймворків роблять її доцільною для реалізації нашого проекту. Для написання логіки було обрано фреймворк Yii2, що забезпечує гнучкість розробки, зручність інтеграції з іншими компонентами системи та швидке створення функціоналу. В якості системи управління базами даних використовується MySQL, що гарантує стабільність і високу продуктивність при обробці великих обсягів інформації.

Застосування Yii2 як основного фреймворку дозволяє реалізувати архітектурний підхід MVC (Model-View-Controller), що сприяє чіткому

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

розділенню бізнес-логіки, представлення та обробки даних. Це підвищує читабельність коду, спрощує тестування окремих компонентів та забезпечує можливість подальшого масштабування системи. Крім того, Yii2 містить вбудовані механізми безпеки, такі як захист від SQL-ін'єкцій, CSRF-атак та XSS, що є критично важливим для інформаційних систем, які працюють з персональними даними користувачів.

Використання MySQL як реляційної системи управління базами даних обумовлено її високою надійністю, оптимізованою продуктивністю та широкою підтримкою з боку різних фреймворків і хмарних сервісів. Система дозволяє ефективно реалізувати складні запити, індексацію даних та механізми транзакцій, що особливо важливо для модулів авторизації, збереження журналів подій та управління користувацькими даними. Додатково MySQL забезпечує можливість реплікації та резервного копіювання, що підвищує відмовостійкість системи.

Для стабільної роботи розроблюваної системи необхідно передбачити серверне середовище, здатне обробляти одночасні запити користувачів, виконувати операції авторизації та забезпечувати збереження журналів подій. Рекомендовано використовувати сервер із процесором не менше ніж на чотири ядра з тактовою частотою від 3.00 GHz, підтримкою багатопоточності та архітектурою x86_64. Такі характеристики дозволять забезпечити достатній рівень продуктивності під час обробки запитів до модулів авторизації, логування та взаємодії з базою даних.

Оперативна пам'ять сервера має становити не менше 32 GB, оскільки система працює з великою кількістю користувацьких запитів, даними авторизації та журналами помилок. За можливості доцільно використовувати пам'ять із підтримкою ECC, що підвищує надійність роботи системи в умовах інтенсивного навантаження. Для збереження даних рекомендовано застосовувати SSD-накопичувач обсягом не менше 200 GB із додатково зарезервованим простором для логів, оскільки журнали помилок і подій можуть швидко збільшуватися під час активної експлуатації платформи.

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				21
Змн.	Арк.	№ докум.	Підпис	Дата		

Як операційне середовище доцільно використовувати Linux-систему на основі Debian або Ubuntu. Сервер повинен підтримувати механізми автоматичного запуску процесів через systemd або supervisord, а також можливість розгортання застосунку у Docker-контейнерах або на віртуальній машині. Для коректної взаємодії з іншими компонентами інфраструктури необхідно забезпечити стабільне мережеве підключення з пропускнуою здатністю не менше 100 Mbps та мінімальною затримкою у внутрішній мережі.

Додатково важливим аспектом є використання контейнеризації на основі Docker, що дозволяє стандартизувати середовище виконання застосунку та уникнути залежностей від конкретної конфігурації серверної операційної системи. Це спрощує процес розгортання, оновлення та масштабування системи, а також забезпечує передбачувану поведінку програмного забезпечення в різних середовищах — від локальної розробки до продуктивного сервера.

Окрему увагу слід приділити організації резервного копіювання та відновлення даних. У межах системи доцільно реалізувати регулярне створення резервних копій бази даних та журналів подій із можливістю їх автоматичного відновлення у разі збоїв або втрати даних. Це є критичним елементом забезпечення надійності інформаційної системи закладу вищої освіти.

Крім того, для підвищення безпеки системи рекомендується використання HTTPS із SSL/TLS сертифікатами, а також впровадження механізмів обмеження доступу за IP-адресами або ролями користувачів. Це дозволяє мінімізувати ризики несанкціонованого доступу до системи та забезпечити захист персональних даних студентів і співробітників.

Важливим елементом реалізації системи є впровадження механізмів централізованого логування та моніторингу стану застосунку. Це дозволяє не лише фіксувати критичні помилки, але й аналізувати поведінку користувачів та навантаження на систему в реальному часі. Такий підхід забезпечує своєчасне виявлення аномалій у роботі програмного забезпечення та підвищує загальну стабільність платформи.

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				22
Змн.	Арк.	№ докум.	Підпис	Дата		

Для оптимізації продуктивності системи доцільно застосовувати кешування часто використовуваних даних, зокрема інформації про користувачів, ролі та права доступу. Використання кешування дозволяє зменшити кількість звернень до бази даних, що особливо важливо при великій кількості одночасних користувачів у періоди пікового навантаження, наприклад під час вступної кампанії або сесій.

Окремим аспектом є реалізація ролей та прав доступу користувачів у системі. Розмежування доступу між адміністраторами, викладачами, студентами та вступниками дозволяє забезпечити контроль над функціоналом системи та запобігти несанкціонованим діям. Такий підхід реалізується через ролеву модель авторизації (RBAC), яка є стандартом для сучасних інформаційних систем.

Для підвищення надійності та масштабованості системи передбачено можливість горизонтального масштабування шляхом запуску декількох екземплярів застосунку у контейнерах Docker із використанням балансувальника навантаження. Це дозволяє рівномірно розподіляти запити між серверами та забезпечувати стабільну роботу системи навіть при значному зростанні кількості користувачів.

Висновки до першого розділу

У першому розділі було проведено всебічний аналіз поставленого завдання, що дозволило чітко окреслити мету та ключові цілі кваліфікаційної роботи. Особлива увага приділялася формуванню коректної постановки проблеми, оскільки саме це є відправною точкою для побудови ефективної та стабільної інформаційної системи «Цифровий університет». Визначення основних функціональних вимог до розроблюваного продукту, зокрема модулів авторизації та логування, стало важливим етапом планування, дозволивши встановити пріоритети та розробити стратегічний підхід до реалізації.

На основі проведеного аналізу обрана монолітна архітектура, що забезпечує централізоване управління всіма компонентами системи, високий рівень продуктивності та зручність налагодження. Такий підхід дозволяє

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

ефективно реалізувати функції авторизації через LDAP, Office 365, Google OAuth2 та електронну пошту, а також здійснювати централізоване логування винятків і критичних помилок із збереженням у базі даних та файлах. Монолітна структура спрощує управління версіями, оновлення модулів і підтримку системи, забезпечуючи стабільну роботу при одночасному обробленні значної кількості запитів користувачів.

Також обґрунтовано вибір інструментальних засобів: мова PHP із фреймворком Yii2 для бекенду забезпечує гнучкість розробки, інтеграцію з іншими компонентами та швидке створення функціоналу, а MySQL гарантує стабільність і високу продуктивність при роботі з великими обсягами даних. Визначені вимоги до апаратного забезпечення, включно з процесором, оперативною пам'яттю, дисковою підсистемою та стабільним підключенням до мережі, дозволяють забезпечити безперебійну роботу системи та належний рівень продуктивності навіть при пікових навантаженнях.

Таким чином, виконаний аналіз і прийняті рішення формують надійну концептуальну основу для подальшої розробки програмного продукту. Визначені функціональні напрямки, обрані інструменти та архітектурні рішення забезпечують ефективний старт реалізації системи, гарантують її стабільність, безпеку та зручність для користувачів, відповідаючи сучасним технологічним вимогам і очікуванням освітніх закладів.

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		24

РОЗДІЛ 2. ПРОЕКТУВАННЯ СИСТЕМИ ОНЛАЙН-МЕНЕДЖМЕНТУ ОСВІТНІХ ПРОЦЕСІВ

2.1. Визначення варіантів використання та структури системи

Для забезпечення стабільної та безпечної роботи модулів «Авторизація» та «Логування» в рамках інформаційної системи «Цифровий університет» сформульовано комплекс вимог до їх розробки. Вони включають як функціональні, так і нефункціональні аспекти, що спрямовані на гарантування високої продуктивності, надійності, безпеки, масштабованості та сумісності з іншими компонентами системи.

З позиції бізнес-вимог модулі «Авторизація» та «Логування» призначені для підвищення безпеки, керованості та прозорості роботи інформаційної системи «Цифровий університет». Їх реалізація спрямована на централізоване управління доступом користувачів, підтримку інтеграції з корпоративними та хмарними сервісами, автоматичне створення облікових записів для окремих категорій користувачів і фіксацію важливих подій, які впливають на стабільність роботи програмного продукту.

Система реалізується як веб-застосунок із монолітною архітектурою, у межах якої модулі авторизації та логування інтегровані безпосередньо в загальну структуру платформи. Серверна частина побудована на основі РНР-фреймворку Yii2, що забезпечує обробку запитів, взаємодію з базою даних, рендеринг сторінок і виконання бізнес-логіки. Централізоване зберігання даних у базі дозволяє уникати дублювання інформації, підтримувати її актуальність і забезпечувати контроль доступу відповідно до ролей користувачів.

Для адміністрації університету важливою є можливість керувати обліковими записами користувачів, переглядати інформацію про їхній статус, роль, групову належність та історію авторизації. Система має забезпечувати фільтрацію й сортування користувачів, що спрощує роботу з великою кількістю акаунтів. Окреме значення має автоматизоване створення корпоративних облікових записів та підключення до LDAP і Office 365, оскільки ці механізми

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

дозволяють уніфікувати роботу з користувачами та зменшити кількість ручних операцій.

Для власників системи важливою є можливість гнучкого налаштування параметрів авторизації, перегляду логів, контролю дій користувачів і зміни облікових даних, що використовуються для інтеграції із зовнішніми сервісами. Такий функціонал дозволяє швидко реагувати на зміни в інфраструктурі, коригувати параметри підключення до сервісів автентифікації та підтримувати стабільну роботу системи без безпосереднього втручання у програмний код.

Характеристика об'єкта комп'ютеризації.

Об'єктом комп'ютеризації є процес доступу користувачів до інформаційної системи закладу вищої освіти через авторизацію. Основними користувачами системи будуть робітники університету, студенти та вступники, які взаємодіють із системою через веб-інтерфейс. Модулі повинні бути зручними та інтуїтивно зрозумілими як для досвідчених користувачів, так і для новачків, забезпечувати безпечний вхід до системи, контроль дій користувачів та надійне збереження інформації без необхідності встановлення додаткового програмного забезпечення на пристрої користувачів. Логування буде доступно лише команді розробки та адміністрації.

Функціональні вимоги до системи визначають, що система повинна забезпечувати надійну авторизацію користувачів. Передбачено централізоване керування обліковими записами користувачів, включаючи автоматичне створення акаунтів для студентів та персоналу.

Модуль логування має фіксувати всі помилки які виникають у користувачів і системні події, також має зберігати записи у базі даних та файлах для подальшого аналізу. Система повинна підтримувати гнучкий пошук та фільтрацію подій за користувачами, типом події, датою та іншими параметрами. Крім того, передбачено інтеграцію з корпоративною електронною поштою, Office 365 і LDAP, включно з налаштуванням SSL-сертифікатів для безпечного підключення до LDAP-сервера та створенням облікових записів із різними рівнями доступу. Система повинна забезпечувати надійний контроль доступу до

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

ресурсів і повну прозорість дій користувачів для підвищення безпеки та стабільності роботи.

Нефункціональні вимоги до системи визначають якісні характеристики її роботи, зокрема зручність використання, надійність, продуктивність і придатність до подальшої експлуатації. Інтерфейс повинен бути інтуїтивно зрозумілим для основних категорій користувачів, щоб працівники університету могли опанувати базові сценарії роботи протягом одного навчального заняття тривалістю до 60 хвилин. Форми введення даних мають містити зрозумілі підказки, повідомлення про помилки та логічну структуру полів, що зменшує ймовірність некоректного введення інформації.

Система повинна забезпечувати швидкий відгук інтерфейсу під час стандартних дій користувача. Для типових операцій, таких як вхід до системи, відкриття сторінок або перегляд основної інформації, час відповіді не повинен перевищувати 2 секунд. Для складних пошукових або аналітичних запитів допустимим є більший час виконання, однак він має залишатися в межах, які не створюють відчутного дискомфорту для користувача.

Надійність системи передбачає її доступність не менше ніж 99,5% часу протягом доби та можливість відновлення після збою протягом 15 хвилин. Перед виконанням критичних операцій, зокрема зміни прав доступу, створення облікових записів або експорту логів, має здійснюватися перевірка коректності й цілісності даних. Це дозволяє знизити ризик пошкодження інформації та забезпечити стабільність роботи платформи.

Продуктивність системи має бути достатньою для одночасної роботи понад 1000 користувачів без істотного зниження швидкодії. Особливу увагу необхідно приділити обробці великих обсягів логів, фільтрації подій, пошуку користувачів і взаємодії з зовнішніми сервісами авторизації. Крім того, інтерфейс повинен бути адаптивним і коректно відображатися на персональних комп'ютерах, планшетах і смартфонах.

Експлуатаційна придатність системи забезпечується монолітною структурою, яка спрощує супровід, оновлення та додавання нових функцій.

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				27
Змн.	Арк.	№ докум.	Підпис	Дата		

Система має містити журнал подій, механізми перегляду й аналізу помилок, а також засоби сповіщення про критичні ситуації. Інтеграція з корпоративною поштою, Office 365 та LDAP повинна виконуватися із застосуванням безпечних параметрів підключення, зокрема SSL-сертифікатів і контрольованих прав доступу.

Головними користувачами системи є адміністрація - група викладачів та працівників університету, які мають різні ролі та повноваження, що можуть змінюватися залежно від потреб. Веб-портал забезпечує адміністраторам можливість самостійно створювати нові ролі та надавати їм необхідні права доступу, що дозволяє гнучко масштабувати функціональні зони відповідальності кожної ролі. Модулі Авторизація та Логування забезпечують керування доступом користувачів, облік їхніх дій у системі, що відбуваються у програмному середовищі.

Актор «Власник», що представляє компанію-власника системи, здатен оперативно реагувати на запити користувачів, тимчасово обмежувати доступ під час проведення технічних робіт, контролювати дії персоналу через системні журнали та виконувати необхідні налаштування порталу в режимі реального часу без прямого втручання у код.

2.2. Обґрунтування вибору системи управління базами даних

Під час проєктування інформаційної системи онлайн-менеджменту освітніх процесів важливим етапом є вибір системи керування базами даних, оскільки саме база даних забезпечує збереження основної інформації про користувачів, параметри авторизації, ролі, події, помилки та результати взаємодії з зовнішніми сервісами. Для модулів «Авторизація» та «Логування» база даних має особливе значення, адже вона використовується не лише як сховище довідкової інформації, а й як один із ключових елементів забезпечення безпеки, стабільності та контрольованості системи.

У межах розроблюваної системи база даних повинна підтримувати роботу з різними категоріями користувачів. До них належать студенти, вступники,

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

викладачі, адміністративний персонал і технічні користувачі, які забезпечують інтеграцію з зовнішніми сервісами. Для кожного з них необхідно зберігати ідентифікаційні дані, зв'язок із відповідною роллю, статус активності, параметри доступу та інформацію, необхідну для подальшої взаємодії з системою. Крім того, база даних має містити конфігураційні параметри для роботи з LDAP, Google OAuth2 та Office 365, що дозволяє змінювати налаштування авторизації без прямого редагування програмного коду.

Для реалізації системи було обрано MySQL, оскільки ця система керування базами даних добре підходить для веб-орієнтованих застосунків, розроблених із використанням PHP та фреймворку Yii2. MySQL забезпечує достатній рівень продуктивності для обробки великої кількості запитів, підтримує транзакції, індексацію, зовнішні ключі та механізми розмежування доступу. Такі можливості є важливими для інформаційної системи, у якій паралельно виконуються операції входу користувачів, перевірки ролей, створення акаунтів, запису помилок і формування журналів подій.

Важливою перевагою MySQL є її сумісність із фреймворком Yii2. Yii2 містить зручний механізм ActiveRecord, який дозволяє працювати з таблицями бази даних як із моделями предметної області. Завдяки цьому зменшується кількість дублювання коду, спрощується реалізація CRUD-операцій і підвищується читабельність програмної логіки. Для модулів авторизації та логування це дає можливість швидко отримувати дані користувача, перевіряти наявність ролей, зберігати інформацію про помилки та будувати запити для фільтрації журналів.

Під час вибору системи керування базами даних також враховувалися вимоги до надійності. MySQL підтримує механізми транзакційності, що дозволяє забезпечити коректне виконання операцій, пов'язаних зі створенням або оновленням облікових записів. Наприклад, під час автоматичного створення користувача важливо, щоб збереження основної інформації, призначення ролі та додавання пов'язаних даних виконувалися узгоджено. У разі помилки система повинна мати можливість не залишати базу даних у частково зміненому стані.

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

Окрему роль у роботі системи відіграє логування. Записи про помилки можуть створюватися часто, особливо під час активної експлуатації платформи або тестування нового функціоналу. Тому база даних повинна ефективно зберігати велику кількість однотипних записів і забезпечувати швидкий пошук за датою, користувачем, URL-адресою, типом помилки або іншими параметрами. MySQL дозволяє оптимізувати такі операції за допомогою індексів, що є важливим для сторінок перегляду логів і статистики.

Для адміністрування бази даних використовується DataGrip, підключення до якого виконується через SSH. Такий підхід підвищує безпеку роботи з базою, оскільки прямий доступ до серверного середовища обмежується, а всі операції виконуються через захищений канал. DataGrip дає можливість переглядати структуру таблиць, виконувати SQL-запити, контролювати зв'язки між сутностями та перевіряти коректність даних під час розробки й супроводу системи.

У контексті безпеки MySQL дозволяє створювати окремих користувачів бази даних із різними рівнями прав. Це важливо для поділу доступу між застосунком, адміністраторами та технічними спеціалістами. Для робочого середовища доцільно надавати застосунку лише ті права, які необхідні для виконання його функцій, а адміністративні операції виконувати через окремий обліковий запис. Такий підхід знижує ризик несанкціонованої зміни або видалення даних.

Окремо слід відзначити можливості MySQL щодо масштабування та роботи з високими навантаженнями. Система підтримує реплікацію даних, що дозволяє створювати резервні копії бази в реальному часі та розподіляти навантаження між декількома серверами. Це є важливим для інформаційної системи освітнього закладу, оскільки кількість одночасних користувачів може суттєво зростати в періоди реєстрації, складання іспитів або масового входу до системи.

Важливим аспектом є також підтримка механізмів резервного копіювання та відновлення даних. MySQL дозволяє реалізувати як повне, так і

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				30
Змн.	Арк.	№ докум.	Підпис	Дата		

інкрементальне резервне копіювання, що забезпечує мінімізацію втрат даних у разі технічних збоїв або помилок користувачів. Це особливо критично для модулів логуювання, де накопичується велика кількість інформації про події та помилки системи.

Для підвищення продуктивності роботи з базою даних у системі передбачається використання індексації ключових полів, таких як ідентифікатори користувачів, ролі, часові мітки та типи подій. Використання індексів дозволяє значно пришвидшити виконання запитів, особливо при формуванні звітів, фільтрації логів та перевірці прав доступу користувачів.

Також слід враховувати можливість розширення структури бази даних у майбутньому. MySQL дозволяє гнучко змінювати схему даних без суттєвого впливу на роботу системи, що є важливим у контексті розвитку інформаційної платформи. Це дає змогу додавати нові модулі, такі як аналітика навчальної діяльності або розширені сервіси сповіщень, без необхідності повної перебудови існуючої бази даних.

Крім того, MySQL має широку підтримку інструментів моніторингу та оптимізації продуктивності. Адміністратори можуть аналізувати виконання запитів, виявляти “вузькі місця” у роботі бази даних та оптимізувати структуру таблиць або індексів. Це дозволяє підтримувати стабільну роботу системи навіть при зростанні обсягів даних та збільшенні кількості активних користувачів.

Таким чином, вибір MySQL є обґрунтованим з огляду на вимоги розроблюваної системи до продуктивності, стабільності, сумісності з PHP/Yii2, підтримки транзакцій і зручності адміністрування. У поєднанні з DataGrid та безпечним підключенням через SSH ця система керування базами даних забезпечує достатню технічну основу для реалізації модулів авторизації та логуювання.

2.3. Розробка бази даних системи

Розробка бази даних для модулів «Авторизація» та «Логуювання» здійснювалася з урахуванням функціонального призначення системи,

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

особливостей взаємодії з користувачами та необхідності інтеграції з зовнішніми сервісами. База даних має забезпечувати збереження інформації про облікові записи, ролі, параметри авторизації, системні події, помилки та конфігураційні дані, що використовуються під час роботи з LDAP, Google OAuth2 та Office 365.

Основною логічною сутністю системи є користувач. Для нього зберігаються дані, необхідні для ідентифікації в системі, зв'язку з роллю, перевірки статусу доступу та відображення інформації в інтерфейсі. Оскільки користувачі можуть належати до різних категорій, структура бази даних повинна бути достатньо гнучкою. Один користувач може бути студентом, викладачем, вступником або представником адміністрації, а його доступ до функціоналу визначається призначеними ролями та правами.

Для модуля авторизації важливим є збереження параметрів, які визначають спосіб входу користувача до системи. Авторизація може виконуватися через LDAP, Office 365, Google OAuth2 або за допомогою даних, надісланих користувачеві електронною поштою. Тому в базі даних необхідно передбачити таблиці або конфігураційні сутності, які містять параметри підключення до зовнішніх сервісів, адреси callback-обробників, ідентифікатори клієнтів, дозволені області доступу та інші технічні налаштування.

У разі використання Office 365 база даних зберігає конфігураційні параметри, необхідні для формування посилання авторизації, отримання access token та подальшого звернення до Microsoft Graph API. Такі параметри не повинні бути жорстко прописані в коді, оскільки це ускладнює супровід системи та створює ризики для безпеки. Збереження конфігурації у базі даних або захищених змінних середовища дозволяє змінювати параметри підключення без повторного розгортання застосунку.

Для Google OAuth2 база даних також повинна підтримувати збереження параметрів клієнта, redirect URI та областей доступу. Після успішної автентифікації система отримує базову інформацію про користувача, зокрема електронну пошту та ім'я, після чого перевіряє наявність відповідного облікового запису в системі. Такий підхід дозволяє використовувати Google-

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				32
Змн.	Арк.	№ докум.	Підпис	Дата		

авторизацію як зручний спосіб входу для користувачів, які вже існують у базі даних.

Для LDAP-авторизації база даних використовується разом із зовнішнім каталогом користувачів. Після введення логіна та пароля система перевіряє облікові дані через LDAP-сервер, а потім зіставляє отриману інформацію з локальним користувачем. Якщо відповідного запису ще немає, система може створити його автоматично на основі даних, отриманих із каталогу. Це дозволяє підтримувати єдину логіку входу для працівників університету та зменшує потребу в ручному адмініструванні облікових записів.

Окрема частина бази даних пов'язана з обробкою вступників. Для цієї категорії користувачів система повинна зберігати інформацію про електронну пошту, статус заяви, наявність сформованого облікового запису, згенерований пароль або інші дані, необхідні для першого входу. Така структура дозволяє автоматизувати створення кабінетів вступників, повторне надсилання листів, регенерацію паролів і зміну адреси отримувача у випадку помилки.

Модуль логування потребує окремої структури збереження даних. Для кожної помилки або винятку доцільно зберігати тип помилки, повідомлення, URL-адресу, на якій вона виникла, ідентифікатор користувача, метод запиту, IP-адресу, трасування та додаткові службові дані. Таке структурування дозволяє не лише переглядати окремі помилки, а й аналізувати їх повторюваність, групувати за типами та визначати найбільш проблемні ділянки системи.

Для уникнення дублювання однакових помилок може використовуватися хеш, сформований на основі ключових параметрів винятку. Якщо помилка з таким хешем уже існує, система може не створювати повторний запис або оновлювати пов'язану статистику. Це дозволяє зменшити надлишковість даних і зробити журнал помилок більш інформативним для розробників та адміністраторів.

Зв'язки між таблицями повинні забезпечувати цілісність даних. Наприклад, запис про помилку може бути пов'язаний із користувачем, який виконував дію в момент її виникнення, а користувач, у свою чергу, пов'язується

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		33

з роллю або профілем. Така структура дає можливість швидко визначити, у якого користувача виникла проблема, який тип доступу він мав і за яких умов було зафіксовано помилку.

Таким чином, база даних системи виконує не лише функцію сховища, а й забезпечує логічну узгодженість між модулями авторизації, логування та адміністрування. Її структура дозволяє підтримувати різні способи входу, автоматизувати створення облікових записів, зберігати журнали помилок і забезпечувати подальший аналіз стабільності роботи додатку.

Окрему увагу при проєктуванні бази даних приділено нормалізації даних, що дозволяє зменшити надлишковість інформації та забезпечити цілісність збережених записів. Розбиття даних на логічно пов'язані таблиці дає змогу уникнути дублювання інформації про користувачів, ролі та події, а також спрощує супровід і масштабування структури бази даних у майбутньому.

Для підвищення продуктивності роботи системи передбачено використання індексів на ключових полях, які часто використовуються у запитах. Зокрема, індекси доцільно застосовувати для ідентифікаторів користувачів, електронної пошти, типів ролей, часових міток та статусів подій. Це дозволяє суттєво пришвидшити виконання операцій вибірки даних, особливо при формуванні звітів та аналізі логів.

Важливим елементом структури бази даних є механізм аудиту змін, який дозволяє відстежувати всі критичні операції з даними користувачів. До таких операцій належать створення, оновлення та видалення облікових записів, зміна ролей та оновлення параметрів доступу. Збереження історії змін підвищує рівень безпеки системи та забезпечує можливість відновлення попереднього стану даних у разі помилкових дій.

Також передбачено використання механізмів архівації застарілих або неактуальних записів, зокрема логів та історичних даних авторизації. Архівація дозволяє зменшити навантаження на основні таблиці бази даних, підвищити швидкість виконання запитів та оптимізувати використання дискового простору без втрати важливої інформації.

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				34
Змн.	Арк.	№ докум.	Підпис	Дата		

Додатково слід зазначити можливість розширення схеми бази даних для підтримки аналітичних модулів у майбутньому. Це може включати зберігання агрегованих даних про активність користувачів, частоту входів у систему, статистику помилок та інші метрики. Така інформація може бути використана для подальшого вдосконалення системи та прийняття управлінських рішень щодо її розвитку.

2.4. Реалізація функціоналу системи управління навчальним процесом

Перед розглядом програмної реалізації доцільно описати загальну логіку роботи основних способів авторизації, які підтримуються в системі. Діаграми демонструють взаємодію між користувачем, веб-застосунком, зовнішніми сервісами автентифікації та внутрішніми механізмами створення або перевірки облікового запису. Такий підхід дозволяє зрозуміти послідовність дій ще до аналізу окремих фрагментів коду.

У цьому сценарії користувач вводить логін і пароль у формі входу, після чого застосунок передає ці дані на LDAP-сервер через захищене з'єднання. LDAP-сервер виконує перевірку облікових даних і повертає результат автентифікації. Якщо введені дані є коректними, система отримує інформацію про користувача, зіставляє її з локальним записом у базі даних або створює новий обліковий запис. Якщо перевірка завершується помилкою, користувач отримує повідомлення про неможливість входу.



Рисунок 2.1. Діаграма авторизації LDAP

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		35

Таким чином, LDAP-авторизація використовується як механізм інтеграції з корпоративним каталогом користувачів. Її перевага полягає в тому, що облікові дані користувача не дублюються в локальній системі, а перевіряються через централізований сервіс. Це підвищує рівень безпеки та спрощує адміністрування, оскільки зміна пароля або блокування акаунта може виконуватися на рівні LDAP-каталогу.

Рисунок 2.2 відображає логіку авторизації через Office 365 та Google OAuth2. На відміну від LDAP, у цьому випадку користувач не передає пароль безпосередньо до системи. Замість цього застосунок перенаправляє його на сторінку зовнішнього провайдера автентифікації. Після успішного входу провайдер повертає користувача до системи разом із кодом авторизації, який надалі використовується для отримання access token. За допомогою цього токена система отримує базову інформацію про користувача та виконує вхід або створення відповідного акаунта.

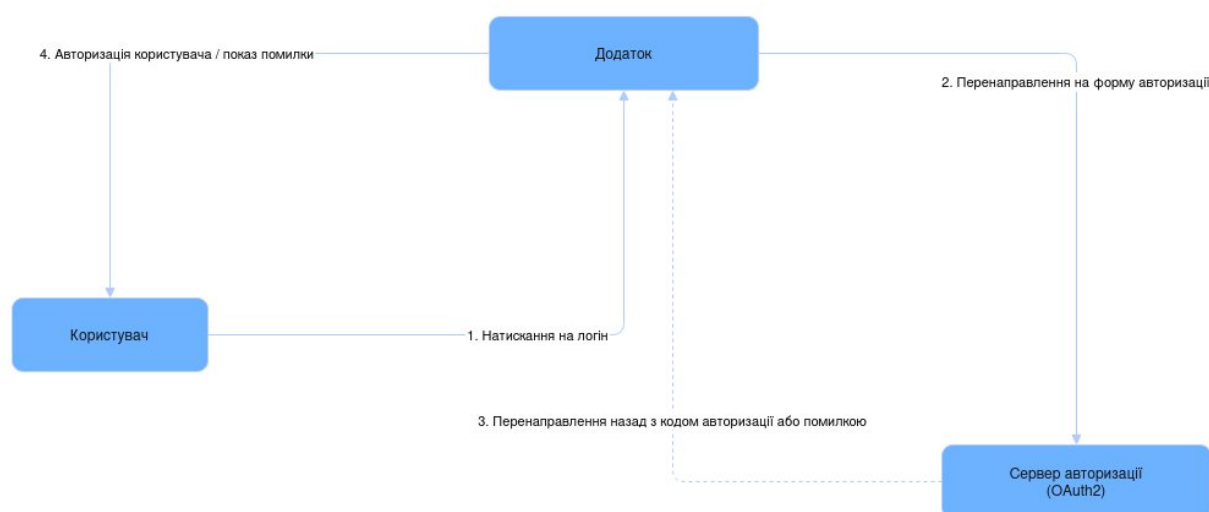


Рисунок 2.2. Діаграма авторизації Office 365 та Google Cloud

Такий підхід відповідає сучасній практиці інтеграції з хмарними сервісами, оскільки користувач проходить автентифікацію на стороні довіреного провайдера. Система не зберігає пароль від Google або Office 365, а працює лише з результатом успішної авторизації. Це зменшує ризики витоку облікових даних

і дозволяє використовувати додаткові механізми захисту, які надаються зовнішніми сервісами, зокрема двофакторну автентифікацію.

Рисунок 2.3 описує авторизацію вступника. У цьому сценарії система отримує інформацію про вступників із відповідного джерела даних, перевіряє статус заяви та визначає, чи потрібно створювати обліковий запис. Якщо користувач відповідає умовам створення кабінету, система генерує пароль, формує обліковий запис і надсилає дані для входу на електронну пошту вступника. У разі необхідності відповідальна особа може повторно надіслати лист або згенерувати новий пароль.

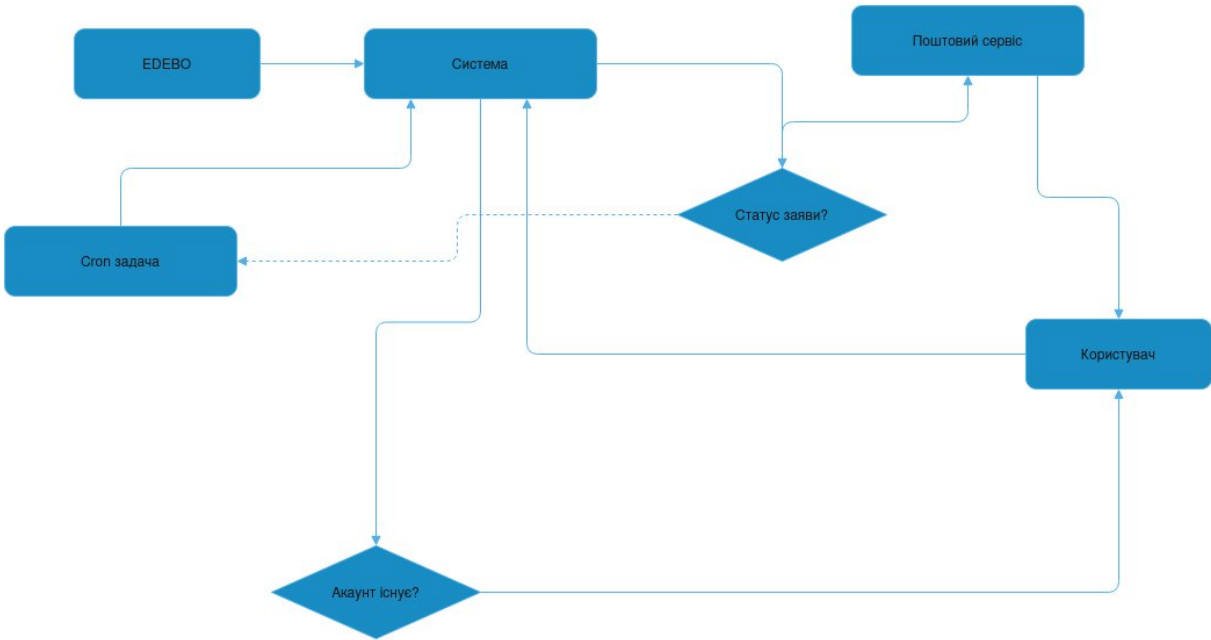


Рисунок 2.3. Діаграма авторизації для абітурієнта

Запропонована схема дозволяє автоматизувати роботу з великою кількістю вступників під час вступної кампанії. Вона зменшує навантаження на працівників університету, мінімізує ризик ручних помилок і забезпечує централізований контроль за створенням облікових записів. Окремою перевагою є можливість повторного надсилання облікових даних, що є важливим у випадку втрати листа або помилково вказаної електронної адреси.

Після опису загальної логіки авторизації доцільно перейти до аналізу програмної реалізації. Фрагменти коду в цьому підрозділі подано не як повний

вміст класів, а як окремі частини, що демонструють ключові етапи роботи модулів. Такий підхід дозволяє послідовно розглянути формування посилання для авторизації, ініціалізацію провайдера, обробку callback-запиту, створення або пошук користувача, зміну пароля та механізм логування помилок.

```
public function actionOffice365Login(): Response
{
    return $this->redirect(AuthService::getOffice365AuthUrl());
}

public static function getOffice365AuthUrl(string $context = 'portal'): string{
    $office365 = new Office365Service($context);
    $provider = $office365->getProvider();
    Yii::$app->session->set('oauth2state' . $context, $provider->getState());
    return $provider->getAuthorizationUrl();
}
```

На першому етапі Office 365-авторизації система формує URL, на який буде перенаправлено користувача. Для цього створюється екземпляр сервісу Office365Service, після чого з нього отримується провайдер автентифікації. Окремо в сесію записується значення state, яке використовується для перевірки коректності відповіді від зовнішнього сервісу та захисту від підміни запиту. Після цього система повертає URL авторизації, за яким користувач переходить на сторінку входу Office 365.

```
public function __construct(string $name = 'portal')
{
    $this->initializeProvider($name);
}

private function initializeProvider(string $name): void{
    $config = Office365Config::find()->where(['name' => $name])->asArray()->one();
    $this->provider = new Azure([
        'clientId' => $config['clientId'], 'clientSecret' => $config['clientSecret'],
        'redirectUri' => $config['redirectUri'], 'tenant' => $config['tenant'],
        'defaultEndPointVersion' => $config['defaultEndPointVersion'],
        'scopes' => $config['scopes'] ]);
}
```

У цьому фрагменті виконується ініціалізація провайдера Azure, який використовується для взаємодії з Office 365. Конфігураційні параметри отримуються з бази даних за назвою контексту авторизації. До таких параметрів належать clientId, clientSecret, redirectUri, tenant, версія кінцевої точки та перелік дозволених scopes. Після отримання цих даних створюється об'єкт провайдера,

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				38
Змн.	Арк.	№ докум.	Підпис	Дата		

який надалі використовується для формування URL авторизації та отримання access token. Важливо, що такі параметри мають зберігатися у захищеному вигляді, оскільки вони забезпечують доступ до зовнішнього сервісу автентифікації.

```
$office365 = new Office365Service($context);
$provider = $office365->getProvider();
$token = $provider->getAccessToken('authorization_code', [
    'code' => $_GET['code']]);
$url = Office365ApiEndpoints::findOne(['name' => 'users_me']);
$me = $provider->get($url->url ?? 'https://graph.microsoft.com/v1.0/me', $token);
return $isStudent
    ? self::makeStudentByButton($me)
    : self::makeUserByButton($me);
```

Після цього як ми отримали AccessToken ми отримуємо URL для отримання інформації про себе та створюємо користувача в системі і після цього ми маємо доступ до акаунту.

Є ще можливість входу через форму в office365:

```
public function actionLogin(): Response|string{
    $model = new LoginForm();
    if ($model->load(Yii::$app->request->post()) && $model->login()) {return
        $this->goBack();}
```

Ми передаємо пароль та логін в LoginForm

```
if (AuthService::isOffice365()) {
    $user = AuthService::makeUser(['password' => $this->password, 'email' =>
        $this->username]);
    if ($user) {
        AuthService::addUserRoleForNewUser($user);
        AuthService::login($user, $this->rememberMe ? \
            3600 * 24 * 30 : 0);
        return true;}
    return false;}
```

У цьому випадку використовується сценарій входу через форму, де користувач вводить логін та пароль безпосередньо в інтерфейсі системи. Введені

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		39

дані передаються до LoginForm, після чого система визначає активний тип авторизації. Якщо увімкнено Office 365, облікові дані передаються до відповідного сервісу для перевірки. У разі успішної автентифікації система отримує дані користувача, створює або оновлює локальний акаунт, призначає необхідну роль і виконує вхід до системи.

```
$office365Service = new Office365Service();
    $authResult = $office365Service->authenticateWithCredentials(
        $email,$userData['password']);
    $user = $authResult['user'];
```

Якщо ми авторизували користувача в office365 правильно, то нам поверне дані про користувача і ми створимо йому акаунт в системі і його автоматично автентифікує після цього.

Наступним способом входу є авторизація через Google OAuth2. Цей механізм використовується для того, щоб користувач міг підтвердити свою особу за допомогою облікового запису Google. На відміну від локального входу, система не обробляє пароль користувача, а перенаправляє його на сторінку Google, де відбувається автентифікація. Після успішного входу Google повертає до системи код авторизації, який використовується для отримання токена доступу.

```
public function actionGoogleLogin(): Response{
    $service = new GoogleAuthService();
    return $this->redirect($service->getAuthUrl());}
```

Після того як ми налаштуємо Google Cloud project ми зможемо отримати посилання для авторизації, яке перекине нас на форму логіну Google, де ми виберемо відповідну пошту та увійдемо в систему. Ми також отримаємо код в Callback:

```
public function actionGoogleCallback(): Response{
    return AuthService::handleGoogleCallback();}
public static function handleGoogleCallback(): Response{
    $googleService = new GoogleAuthService();
    $userData = $googleService->handleCallback($code);
```

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				40
Змн.	Арк.	№ докум.	Підпис	Дата		

```

public function handleCallback(string $code): array{
    $client = $this->getClient();
    $token = $client->fetchAccessTokenWithAuthCode($code);
    $client->setAccessToken($token['access_token']);
    $oauth2 = new OAuth2($client);
    $userInfo = $oauth2->userinfo->get();

```

Після отримання коду авторизації сервіс GoogleAuthService обмінює його на access token і за допомогою OAuth2 API отримує базову інформацію про користувача. Насамперед система використовує електронну пошту, оскільки саме вона дозволяє зіставити Google-акаунт із локальним обліковим записом у базі даних. У поточній реалізації Google-авторизація застосовується для користувачів, які вже існують у системі. Такий підхід є доцільним для студентів або працівників, які мають активний акаунт, але не можуть скористатися стандартним способом входу. Завдяки цьому зменшується кількість звернень до адміністратора щодо відновлення доступу, а користувач отримує альтернативний безпечний спосіб автентифікації.

Окремим способом входу є авторизація через LDAP, яка використовується для інтеграції з корпоративним каталогом користувачів. У цьому сценарії користувач вводить логін і пароль у форму входу, після чого система передає ці дані до LDAP-сервісу. Якщо автентифікація проходить успішно, система отримує інформацію про користувача з каталогу та перевіряє наявність відповідного локального акаунта.

```

public function actionLogin(): Response|string{
    $model = new LoginForm();
    if ($model->load(Yii::$app->request->post()) && $model->login()) {return
    $this->goBack();}

```

Ми як і в минулій авторизації використовуємо цей же контролер для входу через форму. Тут ми також передаємо пароль та логін для створення акаунту.

```

return LdapAuthService::loginTeacher(
    $this->username, $this->password,$this->rememberMe);

```

Першим етапом LDAP-авторизації є перевірка коректності облікових даних користувача. Якщо LDAP-сервер підтверджує логін і пароль, система

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

отримує додаткову інформацію про користувача, яка надалі використовується для створення або оновлення локального запису. Такий підхід дозволяє підтримувати актуальність даних і не дублювати процес адміністрування користувачів у різних системах.

```
if (!LdapService::authenticate($username, $password)) {
    return false;}

```

```
$ldap = LdapService::getLdapUserInfo($username);

```

Якщо знайшли його в LDAP ми намагаємося знайти його в системі, якщо ні створюємо акаунт:

```
$user = User::findByUsername($username);

```

```
if (!$user) {

```

```
$user = self::createUserFromLdap($username, $password, $ldap);

```

Після створення ми додаємо роль викладача для вчителя перевіряючи чи `person_id`:

```
if ($user->person_id) {

```

```
$staff = Staff::findOne(['person_id' => $user->person_id]);

```

```
if ($staff) {

```

```
AuthRepository::assignTeacherRoleIfStaff($user->id);}

```

Після цього ми логінімо користувача в систему.

Переглянемо авторизацію для абітурієнта:

Спочатку ми маємо отримати акаунти для вступників, які здобули відповідний статус:

```
$query = (new Query())->select([

```

```
'person_id', 'email', 'person_request_status_type_name',

```

```
'is_signed_decision', 'is_signed_decision_by_file'])

```

```
->from('edebo_person_requests_infos_2025') ->where(['not', ['email' => null]])

```

```
->andWhere(['<>', 'email', ""]);

```

```
$query->andWhere(['in', 'person_request_status_type_name', [

```

```
'Рекомендовано (контракт)', 'Рекомендовано (бюджет)', 'Допущено', 'Зареєстровано']]);

```

```
$query->andWhere(['or', ['is_signed_decision' => true],

```

```
['is_signed_decision_by_file' => true] ]]);

```

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				42
Змн.	Арк.	№ докум.	Підпис	Дата		

Потім ми перевіряємо чи не створювали ми уже акаунти їм:

```
foreach ($eligibleRequests as $request) {  
    $accountExists = AdmissionCampaign2025Cabinet::find()  
        ->where(['person_id' => $request['person_id']]) ->exists();  
    if (!$accountExists) {try {  
        self::processApplication($request, $credentialsPrefix, $tokenPrefix);  
        $results[] = ['person_id' => $request['person_id'], 'status' => 'success'];  
    } catch (Exception $e) {  
        $results[] = [  
            'person_id' => $request['person_id'], 'status' => 'error', 'message' => $e->getMessage()  
        ]  
    }  
}
```

Після цього ми надсилаємо повідомлення на пошту відповідного типу:

```
$password = (new AbiturientLoginForm())->generatePassword();  
$gmail = new GoogleGmailService($credentialsPrefix, $tokenPrefix);  
$gmail->sendRegistrationMessage($email, $password);
```

Після цього абітурієнт матиме можливість зайти на акаунт.

Якщо абітурієнт забуде пароль ми можемо згенерувати чи втратить лист ми маємо можливість надіслати йому новий згенерований пароль:

```
$password = $this->generatePassword();  
$result = $this->sendPasswordEmail($cabinetId, $password);  
$this->createUserAccount($cabinet->person_id,  
    $cabinet->email, $password);
```

Або надіслати старий:

```
$result = $this->sendPasswordEmail($cabinetId, $cabinet->password);
```

Абітурієнту на пошту прийде повідомлення, що дозволить йому зайти на сторінку абітурієнта.

Також важлива можливість зміни паролю, якщо ми поглянемо на office365:

```
AuthService::changePassword($email, $currentPassword, $newPassword, 'cabinet'))  
office365Service = new Office365Service($name);  
$result = $office365Service->changePasswordSelf($email, $oldPassword, $newPassword);  
public function changePasswordSelf(string $email, string $oldPassword, string  
    $newPassword): array{try {  
    $accessToken = self::getPasswordAccessToken($email, $oldPassword);  
    $this->apiRequest(
```

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				43
Змн.	Арк.	№ докум.	Підпис	Дата		

```
'POST','change_password',$accessToken, [
  'headers' => ['Content-Type' => 'application/json'],
  'body' => ['currentPassword' => $oldPassword,'newPassword' => $newPassword]]];
```

Ми відправляємо запит на зміну паролю від самого користувача, щоб таке виконувати потрібно надавати доступи в Azure.

Якщо ми хочемо змінити пароль в LDAP:

```
if (LdapService::changeStudentPassword($username, $currentPassword, $newPassword))
public static function changePassword(
  string $login,string $password,
  string $newPassword,string|null $searchField = self::USER_SEARCH_FIELD){
  $provider = self::getLdapEditProvider()->connect();
  if (LdapService::authenticate($login, $password)) {
    return self::setPasswordTo($login, $newPassword, $provider, $searchField);
  }
  $searchProvider = self::getLdapSearchProvider()->connect();
  $user = self::getUserByUserName($searchProvider, $searchField,
  $username)??self::getUserByUserName($searchProvider,
  $searchField,$userFromDatabase->fio)??
  self::getUserByEmail($searchProvider,$userFromDatabase->email);
```

Спочатку нам потрібно перевірити чи існує користувач, а уже потім ми змінюємо йому пароль.

```
if ($user) {
  $dn = $user->getDn();
  $encodedPassword = self::encodePassword($newPassword);
  $entry = ['unicodePwd' => [$encodedPassword]];
  $ldapConnection = $provider->getConnection()-
  >getConnection();
  if (ldap_mod_replace($ldapConnection, $dn, $entry))
```

Після того як ми знайшли користувача нам потрібно закодувати пароль та замінити його в LDAP.

Після розгляду механізмів авторизації необхідно описати реалізацію логування помилок. Цей модуль відповідає за автоматичне перехоплення винятків і критичних помилок, які виникають під час роботи застосунку. Його завдання полягає не лише у фіксації факту помилки, а й у збереженні контексту,

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

необхідного для подальшого аналізу: URL-адреси, повідомлення, трасування, користувача, методу запиту та додаткових параметрів середовища.

```
'bootstrap' => [  
    'queryLogger',  
    function () {  
        ErrorLogger::register();},],
```

Для запуску механізму логування під час ініціалізації застосунку реєструється спеціальний обробник помилок. Він підключається на етапі bootstrap, завдяки чому система починає перехоплювати винятки ще до виконання основної бізнес-логіки. Це важливо, оскільки помилки можуть виникати не лише під час роботи контролерів, а й на ранніх етапах обробки запиту.

```
public static function register(): void{  
    set_exception_handler([self::class, 'handleException']);  
    register_shutdown_function([self::class, 'handleShutdown']);}
```

Для роботи слухача потрібна функція register і відповідна логіка.

```
private static function processException($exception): void{  
    if (self::shouldIgnore($exception)) {return;}  
    $statusCode = self::getStatusCode($exception);  
    $type = self::getOrCreateErrorType($exception, $statusCode);  
    $url = self::getCurrentUrl();  
    $hash = self::generateHash($exception, $statusCode, $url);  
    if (ErrorLog::find()->where(['hash' => $hash])->exists()) { return;}  
    self::saveError($type->id, $url, $exception, $hash);}
```

Під час обробки винятку система спочатку перевіряє, чи не належить він до переліку помилок, які не потрібно зберігати. Після цього визначається HTTP-статус, тип помилки, поточна URL-адреса та формується унікальний хеш. Хеш використовується для виявлення повторюваних помилок, щоб журнал не дублював однакові записи й залишався зручним для аналізу. Якщо аналогічна помилка вже була зафіксована, система може не створювати повторний запис. Якщо помилка є новою, вона зберігається у базі даних разом із контекстною інформацією.

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

```

private static function saveError(int $errorTypeId, string $url, $exception, string $hash):
void{
    $userId = Yii::$app->user->id ?? null;
    $error = new ErrorLog([
        'error_type_id' => $errorTypeId, 'user_id' => $userId,
        'url' => $url, 'message' => $exception->getMessage(),
        'trace' => $exception->getTraceAsString(), 'hash' => $hash,
        'request_method' => $_SERVER['REQUEST_METHOD'] ??
        null, 'referrer_url' => $_SERVER['HTTP_REFERER'] ?? null,
        'extra_data' => self::getExtraData($exception),]); $error->save(false); }

```

Під час збереження помилки до бази даних записуються її основні характеристики: тип, повідомлення, трасування, URL-адреса, метод HTTP-запиту, сторінка переходу та додаткові службові дані. Якщо користувач був авторизований, до запису також додається його ідентифікатор. Така деталізація дозволяє відтворити контекст виникнення помилки та швидше визначити її причину. Наприклад, адміністратор може перевірити, на якій сторінці виникла проблема, який запит її спричинив і чи була вона пов'язана з діями конкретного користувача.

Переглянемо отримання логів з файлів:

```

public function actionGetLogFiles(): array{
    Yii::$app->response->format = Response::FORMAT_JSON;
    $backendLogs=glob(Yii::getAlias('@backend/runtime/logs/*.log'))
    $frontendLogs=glob(Yii::getAlias('@frontend/runtime/logs/*.log'))
    return [
        'backend' => array_map('basename', $backendLogs),
        'frontend' => array_map('basename', $frontendLogs),
    ];
}

```

Для перегляду великих лог-файлів використовується поетапне завантаження вмісту. Система не передає весь файл одразу, оскільки це може створити значне навантаження на сервер і браузер користувача. Замість цього файл розбивається на частини за кількістю рядків, після чого інтерфейс поступово підвантажує наступні фрагменти. Такий механізм є доцільним для

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				46
Змн.	Арк.	№ докум.	Підпис	Дата		

великих журналів, оскільки забезпечує швидке відкриття сторінки, зменшує обсяг переданих даних і робить перегляд логів більш зручним.

```
public function actionGetLogContent(): Response{
    $content = file_get_contents($filePath); $lines = explode("\n", $content);
    $totalLines = count($lines); $remainingLines = max(0, $totalLines - $offset - $limit);
    $chunkLines = array_slice($lines, $offset, $limit);
    $newOffset = $offset + count($chunkLines);

    $chunkContent = ""; $startLineNumber = $offset + 1;
    foreach ($chunkLines as $index => $line) {
        $lineNumber = $startLineNumber + $index;
        $chunkContent .= str_pad($lineNumber, 6, ' ', STR_PAD_LEFT) . ': ' . $line . "\n";
    }
    return $this->asJson([
        'success' => true, 'filename' => $filename,
        'content' => $chunkContent, 'lastModified' => date('Y-m-d H:i:s'),
        filemtime($filePath), 'size' => filesize($filePath),
        'totalLines' => $totalLines, 'offset' => $newOffset,
        'remainingLines' => $remainingLines, 'hasMore' => $remainingLines > 0]);
}
```

Тут ми отримуємо дані з файлу, зроблено це як lazy loading і кожні 15 тисяч підвантажує додаткові дані.

Висновки до другого розділу

У другому розділі було здійснено детальний аналіз вимог та архітектурних рішень, необхідних для реалізації модулів «Авторизація» та «Логування» в межах інформаційної системи «Цифровий університет». У процесі дослідження визначено бізнес-вимоги, функціональні та нефункціональні характеристики системи, що забезпечують надійне управління доступом користувачів, високий рівень безпеки та стабільність роботи веб-порталу.

На основі аналізу предметної області сформовано модель доступу користувачів до системи з урахуванням ролей, типів облікових записів та різних сценаріїв автентифікації. Реалізовано підтримку кількох механізмів авторизації, зокрема через Office 365, Google OAuth2 та LDAP, що дозволяє інтегрувати систему з корпоративною інфраструктурою закладу вищої освіти та зовнішніми хмарними сервісами. Запропонований підхід забезпечує автоматичне створення

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				47
Змн.	Арк.	№ докум.	Підпис	Дата		

облікових записів студентів, вступників і персоналу, а також гнучке керування правами доступу відповідно до призначених ролей.

Окрему увагу приділено реалізації модуля логування, який забезпечує централізовану фіксацію системних подій, помилок і винятків. Реалізований механізм дозволяє зберігати інформацію як у базі даних, так і у файловій системі, що спрощує подальший аналіз, діагностику збоїв та підвищує керованість програмного продукту. Використання унікальних хешів помилок запобігає дублюванню записів і дозволяє зосередитися на аналізі унікальних проблем, що виникають під час експлуатації системи.

Вибір системи керування базами даних MySQL обґрунтовано її продуктивністю, надійністю та зручністю інтеграції з PHP-фреймворком Yii2. Це забезпечило ефективну реалізацію механізмів зберігання облікових записів, ролей користувачів та логів, а також створило умови для подальшого масштабування системи.

У результаті виконаної роботи розроблено надійну та гнучку підсистему авторизації й логування, яка відповідає сучасним вимогам до безпеки, прозорості та керованості інформаційних систем закладів вищої освіти. Запропоновані рішення забезпечують централізоване управління доступом, контроль дій користувачів і стабільну роботу платформи, що є важливою складовою цифрової трансформації освітнього середовища.

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				48
Змн.	Арк.	№ докум.	Підпис	Дата		

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

3.1. Порядок встановлення та налаштування параметрів додатку

Перед запуском проекту на реальних машинах потрібно виконати налаштування відповідних додатків. Розпочнемо з Google Cloud Gmail для надсилання повідомлень:

Спочатку потрібно створити акаунт в Google або використати уже наявний акаунт для авторизації в Google Cloud.

Потім нам потрібно створити проект в GoogleCloud (рис. 3.1):

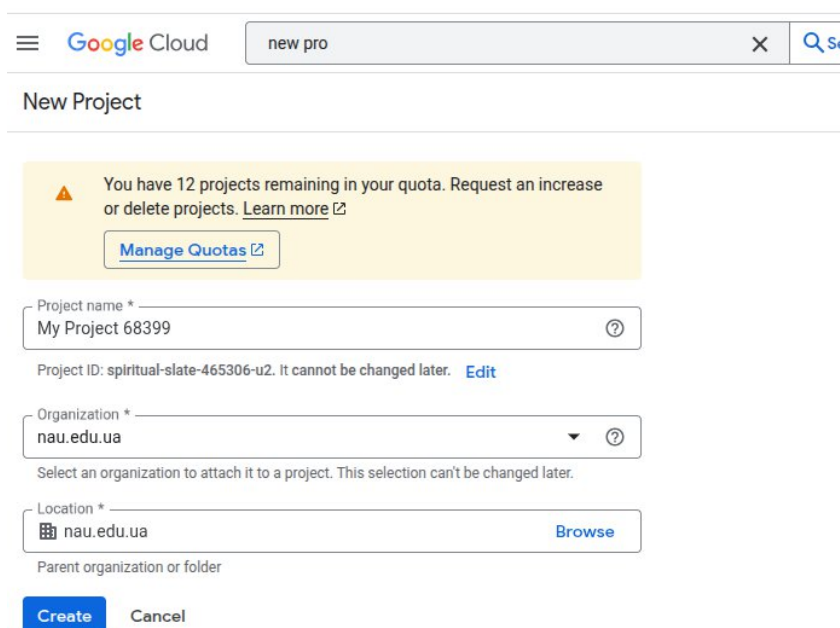


Рисунок 3.1. Створення проекту в Google Cloud

Далі нам потрібно перейти в проект та додати в APIs & Services плагін Gmail Api(рис. 3.2):

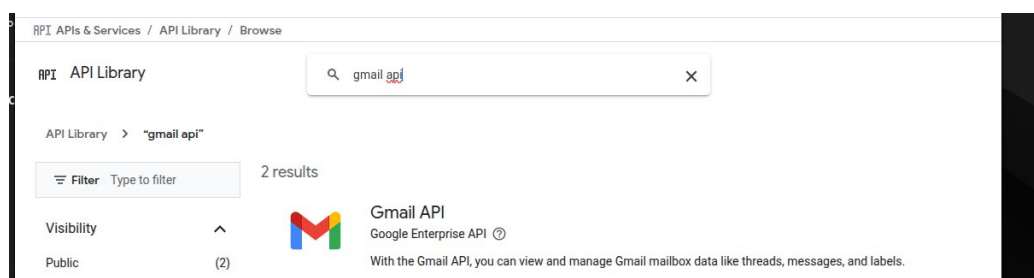


Рисунок 3.2. Додавання плагіну Gmail Api

Далі нам потрібно перейти в OAuth consent screen та розпочати налаштування(рис. 3.3):

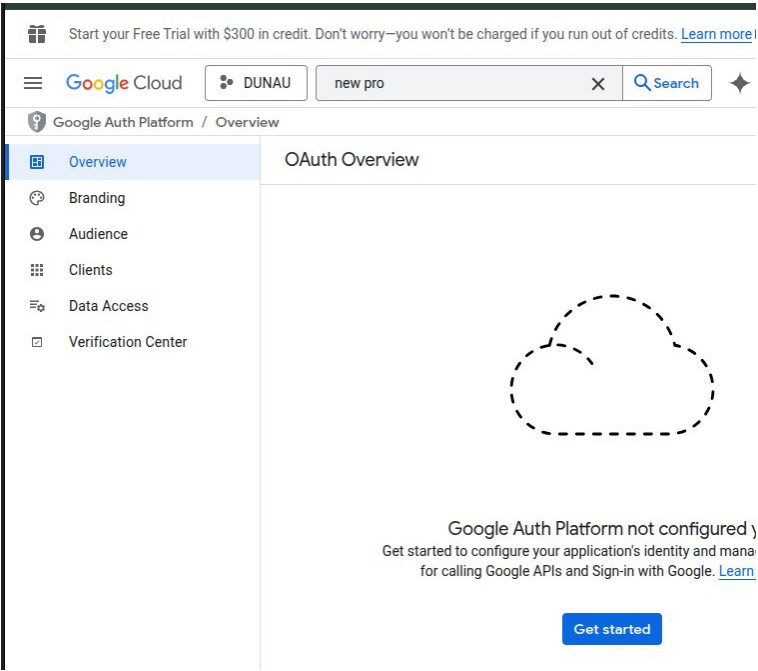


Рисунок 3.3. Сторінка для початку налаштування

Тепер нам потрібно пройти усі етапи створення та закінчити створення(рис. 3.4):

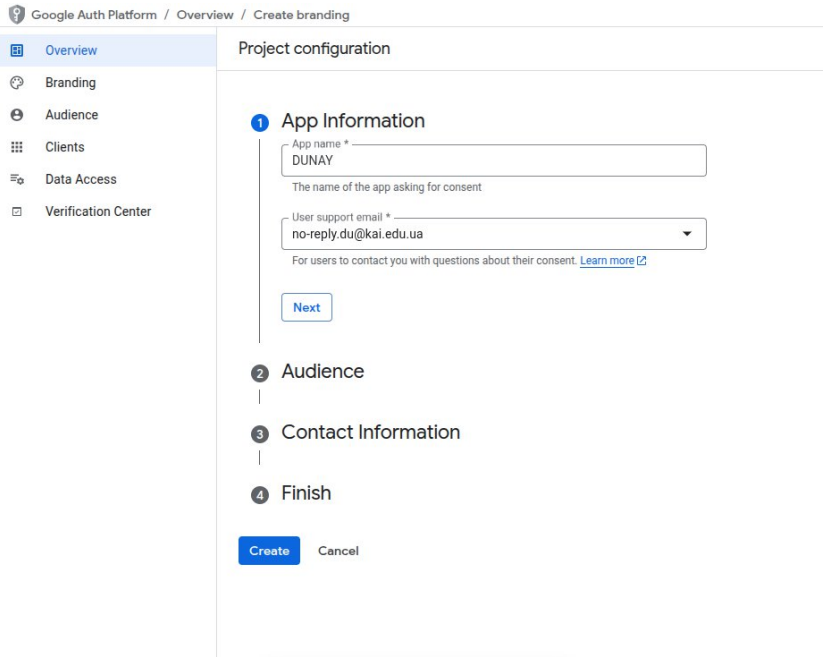


Рисунок 3.4. Форма для створення

Після того як ми створимо нам потрібно скопіювати Client ID та Client Secret (рис. 3.5):

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

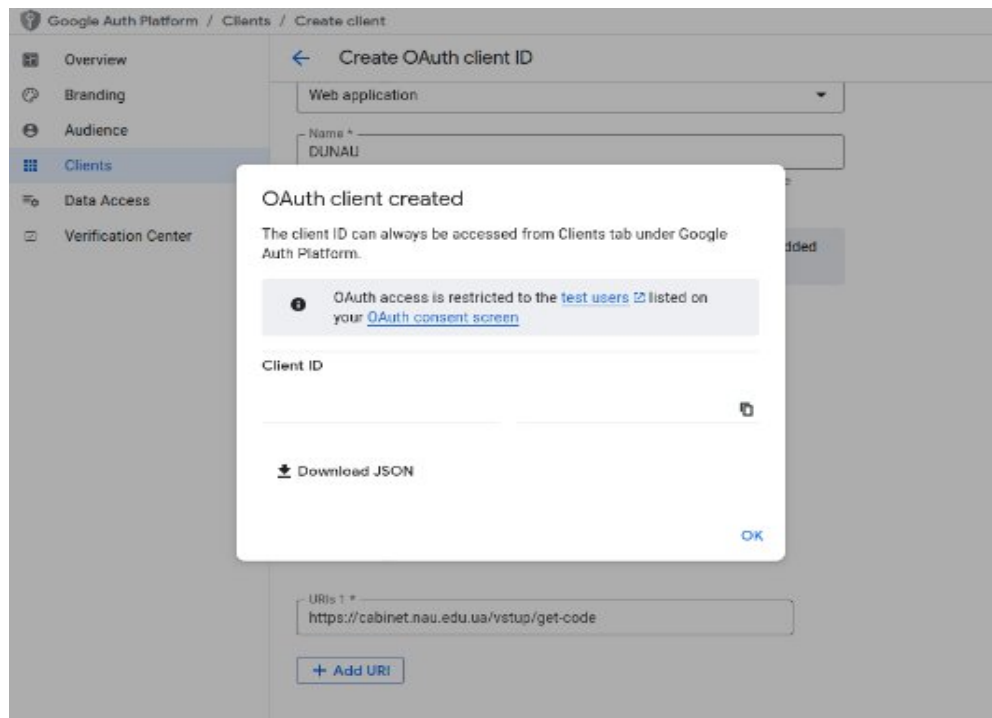


Рисунок 3.5. Форма для копіювання

Далі нам потрібно завантажити файл та вставити на сторінку(рис. 3.6):

Оновлення налаштування пошти: du.knuba.noreply@gmail.com

Почта

du.knuba.noreply@gmail.com

Файл з правами

Оберіть файл ... Обрати ...

Оновити Відмінити

Рисунок 3.6. Форма для збереження даних для входу

Тепер нам потрібно натиснути на кнопку згенерувати токен і після цього перейти на сторінку з кодом, скопіювати його та вставити і після цього пошта почне працювати(рис. 3.7):

Налаштування пошти (+)					
Показані 1-1 із 1 запису.					
#	Почта	чи є credential	чи є token	Credentials Path	Token Path
1	du.knuba.noreply@gmail.com	✓	✓	/var/www/diguni/common/config/json/credentials_1.json	/var/www/diguni/common/config/json/token_1.json

Рисунок 3.7. Сторінка з налаштуваннями пошти

Продовжимо налаштування системи і на черзі у нас авторизація через Google, як і в минулому ми авторизуємося в акаунт та заходимо в Google Cloud, створюємо проект додаємо плагін Google People API(рис. 3.8):

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

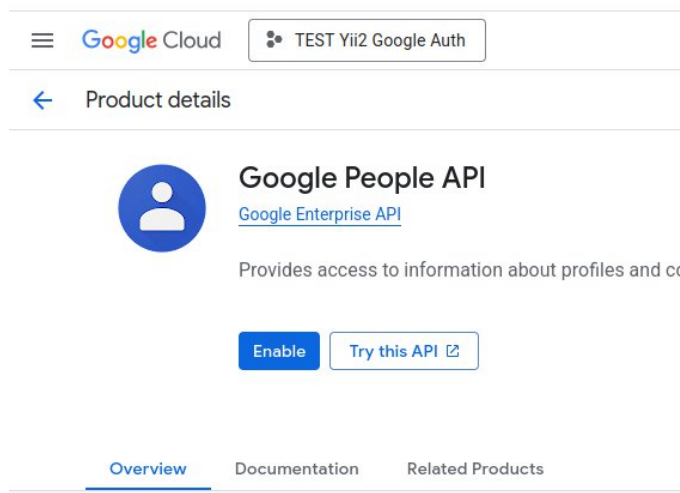


Рисунок 3.8. Плагін для авторизації

Далі нам потрібно скопіювати Client ID та Client Secret та створити запис на сторінці при створенні також додати URL перенаправлення та сферу доступу (рис. 3.9):

Рисунок 3.9. Форма для створення

Після цих дій користувач, який обере метод входу через Google буде попадати на сторінку авторизації і після назад на сайт(рис. 3.10):

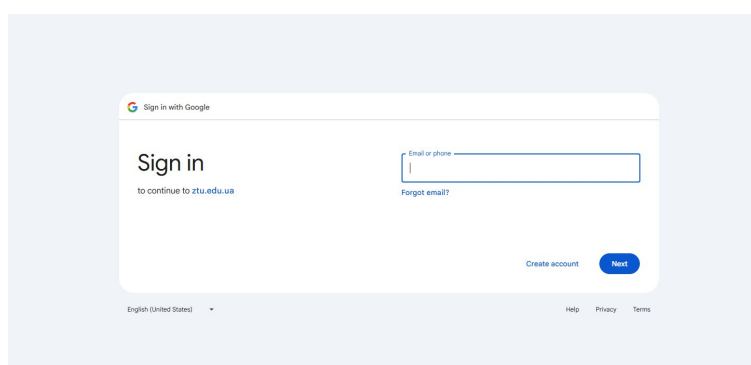


Рисунок 3.10. Google форма для авторизації

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

Продовжимо налаштування системи і на черзі у нас авторизація через Office 365. Перше перевіримо чи наявні URL для взаємодії з Office 365, якщо якийсь URL змінився чи відсутній потрібно додати, щоб авторизація коректно працювала(рис. 3.11):

#	Назва	URL	Опис
1	users	https://graph.microsoft.com/v1.0/users	Створення користувача, перевірка існування
2	users_me	https://graph.microsoft.com/v1.0/me	Отримання інформації про поточного користувача
3	change_password	https://graph.microsoft.com/v1.0/me/changePassword	Зміна пароля користувачем
4	assign_license	https://graph.microsoft.com/v1.0/users/{userId}/assignLicense	Призначення ліцензій користувачу
5	license_details	https://graph.microsoft.com/v1.0/users/{userId}/licenseDetails	Отримання ліцензій користувача
6	subscribed_skus	https://graph.microsoft.com/v1.0/subscribedSkus	Отримання доступних підписок/ліцензій
7	filter_office_location	https://graph.microsoft.com/v1.0/users?\$filter=officeLocation eq '{officeLocation}'&\$select=id,officeLocation&\$count=true	Перевірка існування користувача за officeLocation

Рисунок 3.11. Сторінка з кінцевими точками

Далі нам потрібно отримати Client ID, Client Secret, URL перенаправлення, Tenant ID. Також налаштувати права доступу, а саме дати такі можливості 'User.ReadWrite.all', 'Directory.ReadWrite.All'. Ось ці поля ви можете отримати від адміністраторів університету, також потрібно, щоб вони додали URL перенаправлення та дали права. Після усіх дій створює запис і використовуєте(рис. 3.12):

#	Допом.	ID клієнта	Секрет клієнта	URL перенаправлення	Орендар	Версія ендпоінту за замовчуванням	Області доступу
1							

Рисунок 3.12. Сторінка з редагуванням конфігурацій

Перейдемо до налаштування авторизації за допомогою LDAP. Нам потрібно, щоб Системний адміністратор підняв LDAP надав нам хости з якими взаємодіяти, відповідні порти, надав SSL сертифікати та створив 2 акаунти один із яких адміністраторський подібно до таких:

'username' => 'CN=digunidev,OU=People,DC=ztu,DC=local';

'username'=> "CN=diguniadmin,OU=People,DC=ztu,DC=local";.

І після цього ми можемо додати в common/config/main.php відповідні записи:

```
'ldap' => ['class' => LdapService::class,  
'config' => ['hosts' => ['ldap01.example.net'],  
'port' => 389, 'base_dn' => 'OU=People,DC=example,DC=local',  
'username' => 'CN=ldapreader,OU=ServiceAccounts,DC=example,DC=local',  
'password' => '', 'follow_referrals' => false,  
'use_ssl' => false, 'use_tls' => false,  
'account_prefix' => 'example\\',  
],]  
],]
```

3.2. Структура інтерфейсу додатку

Розроблена система онлайн-керування освітніми процесами має зручний та інтуїтивно зрозумілий інтерфейс користувача, а також забезпечує коректне відображення і повну адаптивність на різних типах пристроїв. Для забезпечення персоналізованого доступу до ресурсу реалізовано механізм автентифікації.

Перед отриманням доступу до головної сторінки порталу користувач повинен пройти процедуру авторизації шляхом введення власних облікових даних у спеціальну форму (рис. 3.13-3.18), що дозволяє гарантувати безпеку та надання функціональних можливостей відповідно до призначеної ролі в системі.

На рисунку 3.13 показано форму входу до порталу, яка містить поля для введення логіна й пароля, прапорець запам'ятовування користувача та кнопку входу через Google. Така форма призначена для користувачів, які мають облікові дані в системі або можуть підтвердити свою особу за допомогою Google-акаунта. Наявність двох способів входу підвищує зручність використання системи та дає змогу обрати найбільш придатний варіант автентифікації.

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		54

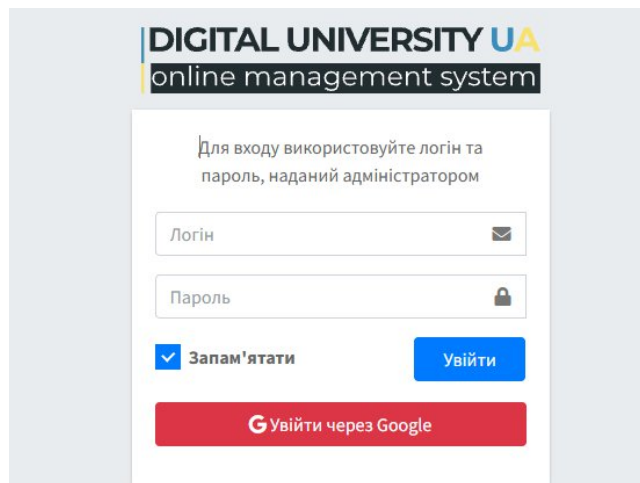


Рисунок 3.13. Вхід на портал через LDAP та Google

Після введення логіна й пароля система перевіряє облікові дані та визначає роль користувача. Якщо користувач обирає вхід через Google, його буде перенаправлено на сторінку зовнішнього провайдера, після чого система отримає підтвердження автентифікації та виконає вхід за наявності відповідного локального акаунта.

На рисунку 3.14 наведено інший варіант форми входу, адаптований для кабінету здобувача. Інтерфейс містить поля для логіна й пароля, кнопку входу та можливість автентифікації через Google. Візуальна структура форми є більш орієнтованою на користувача, оскільки поєднує функціональний блок авторизації з графічним оформленням кабінету.

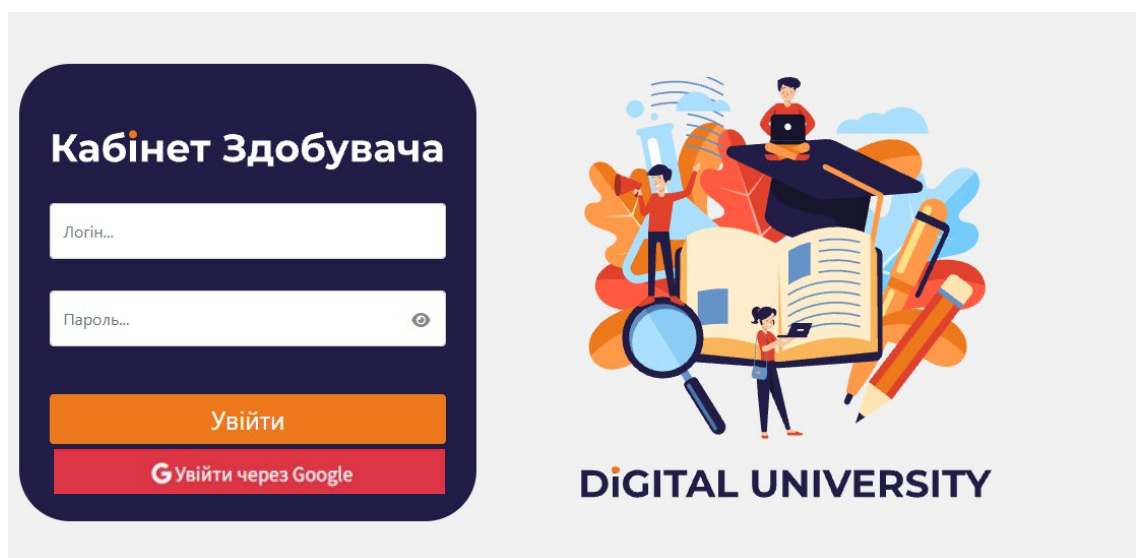


Рисунок 3.14. Вхід на портал через LDAP та Google

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				55
Змн.	Арк.	№ докум.	Підпис	Дата		

Такий варіант інтерфейсу дозволяє зберегти єдину логіку входу, але водночас адаптує зовнішній вигляд сторінки до окремої категорії користувачів. Це важливо для підвищення зрозумілості системи, оскільки здобувач одразу бачить, що перебуває саме на сторінці свого кабінету.

На рисунку 3.15 представлено форму входу до порталу через Office 365. У цьому випадку користувач може ввести логін і пароль або скористатися кнопкою переходу до авторизації через Office 365. Такий спосіб входу є доцільним для користувачів, які працюють із корпоративними обліковими записами університету.

Рисунок 3.15. Вхід на портал через Office 365

Після натискання кнопки Office 365 система перенаправляє користувача до зовнішнього сервісу автентифікації. У разі успішного входу користувач повертається до платформи, а система отримує необхідні дані для перевірки або створення локального облікового запису.

Рисунок 3.16 демонструє спрощений варіант входу до кабінету здобувача через Office 365. На відміну від попередніх форм, тут основний акцент зроблено саме на зовнішній авторизації через корпоративний сервіс. Такий інтерфейс може використовуватися в ситуаціях, коли для певної групи користувачів передбачено єдиний спосіб входу.

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				56
Змн.	Арк.	№ докум.	Підпис	Дата		

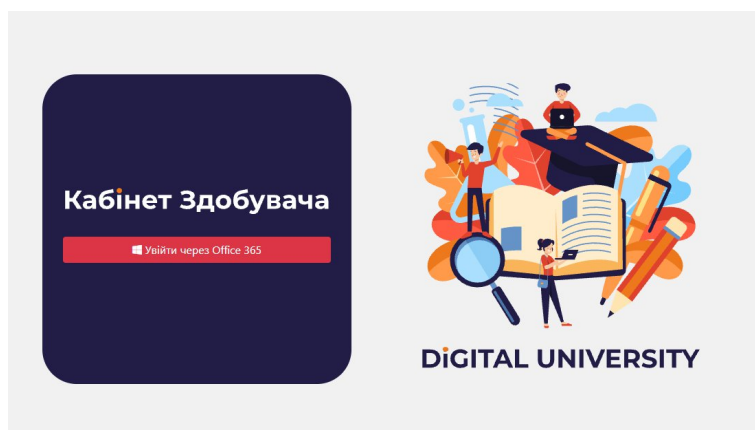


Рисунок 3.16. Вхід в cabinet через Office 365

Використання окремої кнопки Office 365 спрощує сценарій взаємодії для користувача, оскільки не потребує введення додаткових даних у локальну форму. Уся перевірка особи виконується на стороні зовнішнього сервісу, після чого система надає доступ відповідно до ролі користувача.

На рисунку 3.17 показано перший етап входу до кабінету вступника. Користувач вводить електронну пошту, яка була зазначена під час вступної кампанії або отримана з відповідного джерела даних. Система перевіряє наявність такої адреси серед сформованих облікових записів вступників.

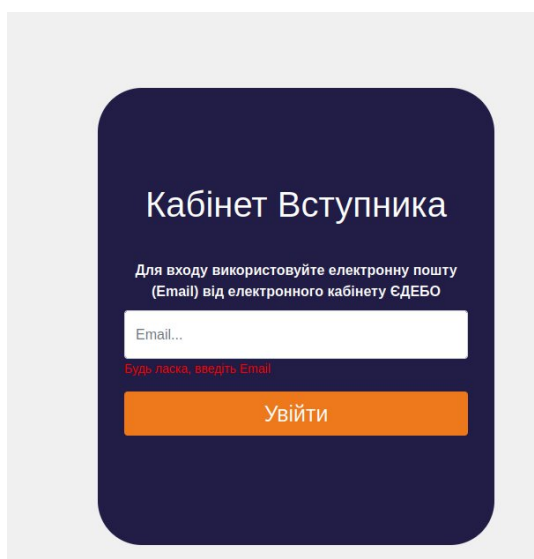


Рисунок 3.17. Кабинет Вступника введення пошти

Якщо електронну пошту знайдено, користувач переходить до наступного етапу автентифікації. Якщо адреса відсутня або введена некоректно, система повідомляє про помилку. Такий поділ процесу входу на етапи дозволяє зробити інтерфейс зрозумілішим і зменшити кількість помилок під час введення даних.

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				57
Змн.	Арк.	№ докум.	Підпис	Дата		

На рисунку 3.18 наведено другий етап входу до кабінету вступника, на якому користувач вводить пароль, надісланий йому електронною поштою. У формі також відображається введена адреса, що дозволяє користувачеві переконатися, що він проходить авторизацію саме для потрібного акаунта.

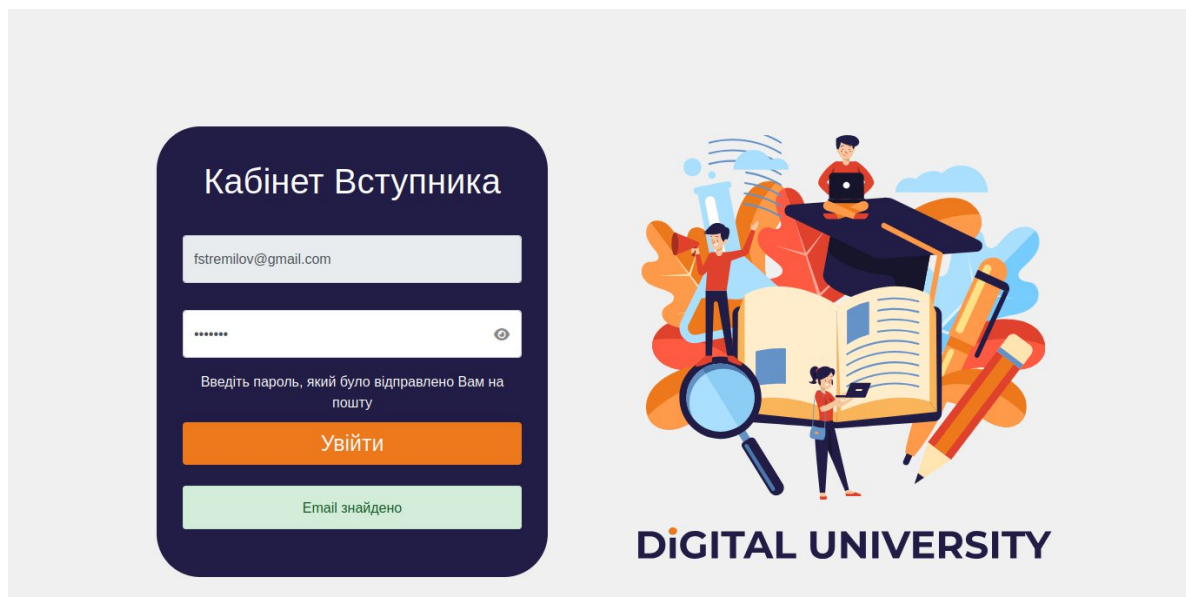


Рисунок 3.18. Кабінет Вступника введення пароля

Після введення пароля система перевіряє його відповідність збереженим обліковим даним. У разі успішної перевірки вступник отримує доступ до власного кабінету. Такий механізм дозволяє централізовано керувати доступом вступників без відкритої самостійної реєстрації, що знижує ризик створення сторонніх або некоректних акаунтів.

З міркувань інформаційної безпеки в системі не реалізовано механізм самостійної реєстрації користувачів через веб-інтерфейс. Створення облікових записів здійснюється відповідним структурним підрозділом університету, який має повноваження на формування та розповсюдження облікових даних (логінів і паролів) для доступу до платформи. Такий централізований підхід до керування доступом дає змогу суттєво знизити ризики несанкціонованого використання системи сторонніми особами, що не пов'язані з університетом.

Після успішного проходження процедури автентифікації користувач автоматично перенаправляється на головну сторінку платформи.

Користувацький інтерфейс спроектовано з урахуванням принципів зручності та

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

швидкого доступу до ключових функціональних можливостей. Для цього значна кількість модулів і підмодулів структурована та логічно згрупована у вигляді навігаційної панелі, розташованої в лівій частині екрана (рис. 3.19-3.21), що спрощує орієнтацію в системі та підвищує ефективність роботи користувачів.

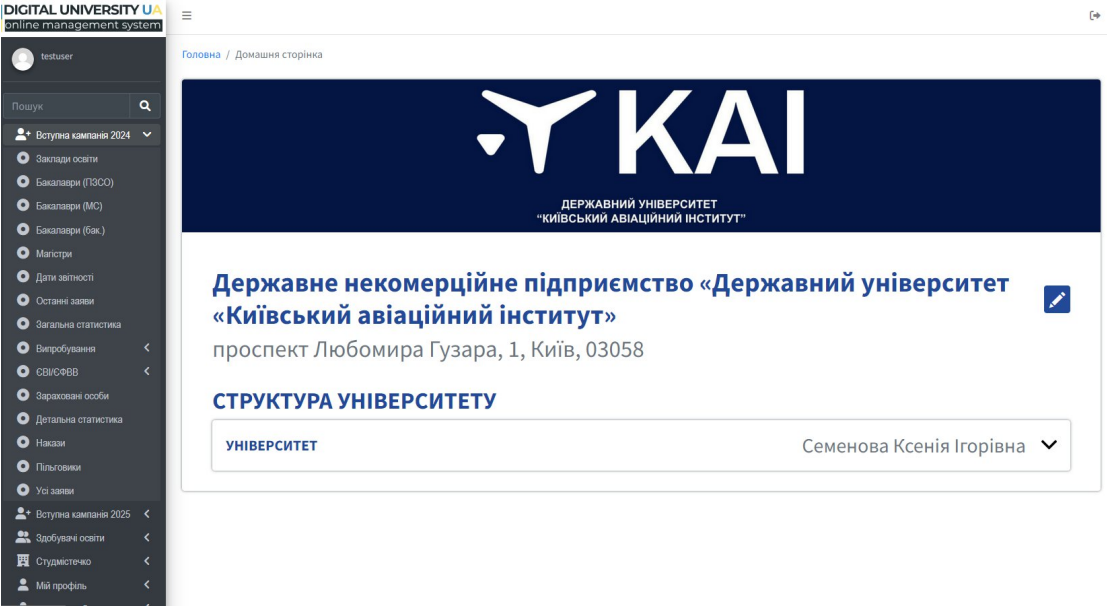


Рисунок 3.19. Головна сторінка викладача

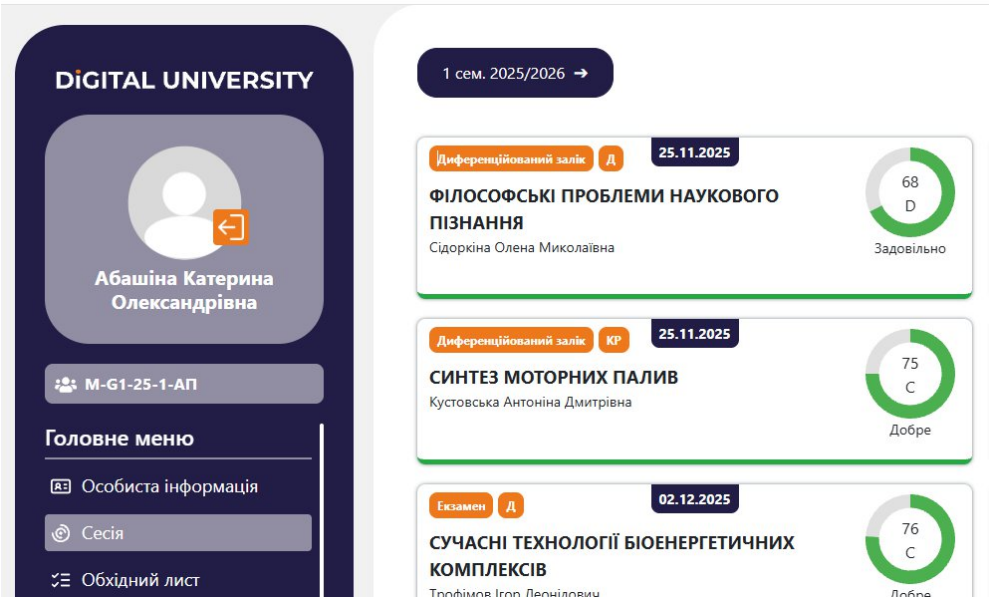


Рисунок 3.20. Головна сторінка студента

Головна сторінка студента - це стартова панель системи, на яку користувач автоматично потрапляє після успішного входу, студент отримує доступ до всього функціоналу, який йому необхідний.

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

DIGITAL UNIVERSITY



Касянчик Роман
Олександрович

Головне меню

Особиста інформація

Завантаження документів

Формування документів

Особиста інформація

Для неповнолітніх є обов'язковим внесення інформації про одного з батьків або законного представника!

Вступник

Мати

Батько

Інший законний представник

ПІБ

Тип документу

Виберіть тип документу...

Телефон

+380 XX-XXX-XX-XX

Серія паспорту (за наявності)

XX

Ел. пошта

example@gmail.com

Номер паспорту

XXXXXX

Місце роботи

Ким виданий

Адреса проживання

Коли виданий

mm/dd/yyyy

Рисунок 3.21. Головна сторінка вступника

Важливою складовою забезпечення інформаційної безпеки та зручності користування системою є можливість зміни пароля. Наявність такого функціоналу дозволяє користувачам своєчасно оновлювати облікові дані у разі підозри на їх компрометацію, після первинного входу або з міркувань підвищення рівня захисту персональної інформації. Крім того, регулярна зміна пароля відповідає загальноприйнятим рекомендаціям з кібербезпеки та сприяє запобіганню несанкціонованому доступу до облікових записів, забезпечуючи надійний захист даних і стабільну роботу системи в цілому (рис. 3.22-3.23).



Рисунок 3.22. Кнопка для зміни паролю

При зміні пароля нове значення автоматично синхронізується із відповідною системою автентифікації, тобто зберігається та оновлюється не лише в межах самої платформи, але й у зовнішніх сервісах, а саме в LDAP або Office 365, що залежно від того, який саме тип авторизації був обраний для конкретного користувача або організації, що забезпечує єдине управління обліковими даними та узгоджений доступ до всіх інтегрованих систем.

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				60
Змн.	Арк.	№ докум.	Підпис	Дата		

Рисунок 3.23. Форма для зміни паролю

Не менш важливою є можливість створення облікових записів викладачів безпосередньо через веб-інтерфейс системи. Такий функціонал значно спрощує та прискорює процес підключення нових працівників до цифрового освітнього середовища, зменшуючи навантаження на технічний персонал і усуваючи потребу у ручному втручанні на рівні серверних налаштувань. Завдяки цьому відповідальні особи можуть оперативно створювати акаунти викладачів, призначати їм ролі та права доступу відповідно до посадових обов'язків, що підвищує гнучкість адміністрування системи та забезпечує її актуальність у процесі кадрових змін(рис. 3.24-3.26).

Рисунок 3.24. Форма для створення викладача в LDAP

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				61
Змн.	Арк.	№ докум.	Підпис	Дата		

СТВОРЕННЯ КОРИСТУВАЧА

ПІБ...

Пароль...

Повторіть пароль...

ID фіз. особи... (необов'язкове)

Зберегти

Рисунок 3.25. Форма для створення викладача в Office 365

На відміну від користувачів, створених через LDAP, у випадку Office 365 для кожного облікового запису додатково автоматично генерується нова електронна пошта відповідно до визначеного системного патерну, що забезпечує стандартизоване створення поштових адрес та їх узгодженість у межах організації.

Працівники Показані 1-15 із 903 записів.									
#	ID фізичної особи	ПІБ	Записи про зайнятість	Стать	Дата народження	Чи активний	ІПН	Логін користувача	
1	4469881	Галак Катерина Тимофіївна		Жіноча	10.06.1970	✓		galak.kt	
2	3906290	Сафронова Олена Олександрівна		Жіноча	17.09.1953	✓	1961803705	safronova.oo	
3	4475535	Алостолова Сосна Любов Олександрівна	✗	Жіноча	10.01.1977	✓	2813406367		
4	4930713	Цалко Олександрій Юрійович		Чоловіча	02.08.1986	✓	3162521713	tsapko.oy	
5	3385948	Кінь Данило Олександрович		Чоловіча	24.02.1997	✓	3548410193	kin.do	
6	5082480	Уряднікова Інга Вікторівна		Жіноча	14.11.1967	✓	2478910847	uryadnikova.iv	
7	4531280	Іваннікова Ганна Василівна		Жіноча	28.10.1981	✓	2988614140	ivannikova.hv	
8	6654454	Бірук Яна Ігорівна		Жіноча	03.07.1991	✓	3342121007	biruk.lai	
9	4453136	Згалаг-Лозинська Любов Олександрівна		Жіноча	05.07.1976	✓	2794508680		
10	1277197	Лазаренко Владислав Ігорович		Чоловіча	29.07.1993	✓	3417803374	lazarenko.vi	
11	4457884	Кобзар Алла Юріївна		Жіноча	09.12.1974	✓	2737117806	kobzar.alu	
12	4475350	Басараб Володимир Олександрович		Чоловіча	28.07.1973	✓	2687215475		
13	6698014	Галдєцька Ірина Григорівна		Жіноча	04.08.1970	✓			
14	4475864	Цалко Юрій Володимирович		Чоловіча	20.08.1961	✓	2551203358		
15	5051072	Ніколаєв Всеволод Петрович		Чоловіча	30.07.1953	✓	1956903275		

Рисунок 3.26. Створення викладача в Office 365 за допомогою натискання на + біля користувача

Важливою складовою функціональності системи є наявність окремої сторінки для редагування шаблону листа, який надсилається абітурієнтам з обліковими даними для входу до системи. Така сторінка дозволяє відповідальним особам оперативно змінювати текст повідомлення, додавати

актуальну інформацію, інструкції щодо першого входу або контакти служби підтримки без необхідності втручання у програмний код. Гнучке налаштування змісту листів підвищує інформативність комунікації з абітурієнтами, зменшує кількість помилок і запитань з їх боку, а також забезпечує відповідність повідомлень поточним вимогам вступної кампанії та внутрішнім регламентам університету(рис. 3.27-3.29).

Шаблон для email		
Тема	Ім'я відправника	Дії
Реєстрація електронного кабінету абітурієнта KAI	Приймальна комісія KAI	Редагувати
Статус вашої заяви	Приймальна комісія	Редагувати

Рисунок 3.27. Сторінка для перегляду та редагування шаблонів

Шаблон для email

Вітаємо!

Для вас створено кабінет студента KAI.

Для входу в систему скористайтеся наступними даними:

Логін (email): example@email.com

Пароль: *****

Посилання на кабінет вступника: <https://cabinet.nau.edu.ua/vstup>

Виконати вимоги до зарахування в кабінеті необхідно не пізніше:

Рисунок 3.28. Шаблон для вступника

Шаблон для вступника надходить у форматі HTML, що дозволяє гнучко керувати його структурою та зовнішнім виглядом, застосовувати власні стилі оформлення, а також компонувати та розміщувати різні інформаційні блоки у довільному порядку відповідно до вимог інтерфейсу та логіки відображення даних у системі.

Шаблон відмови вступнику

Шановний вступнику! / Шановна вступнице!

При поданні заяви на спеціальність **Назва спеціальності** Ви відмітили, що претендуєте на бюджет та контракт. Але Ваш конкурсний бал менше 130, що, відповідно до Порядку прийому, не дозволяє Вам приймати участь у конкурсі на бюджетні місця.

Тому на даному етапі Ви отримали статус «Відмова», але після 11 серпня Ваша заява все-одно зможе приймати участь у конкурсі на контракт та наш університет зможе надати відповідну рекомендацію.

З повагою, приймальна комісія

Рисунок 3.29. Шаблон для відмови студенту у наданні бюджетного місця

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				63
Змн.	Арк.	№ докум.	Підпис	Дата		

Окремої уваги потребує реалізація сторінки керування обліковими даними абітурієнтів, яка надає можливість регенерації пароля, повторного надсилання інформаційного повідомлення та, за необхідності, зміни електронної адреси отримувача. Такий функціонал є критично важливим у випадках втрати листа, помилково вказаної адреси або компрометації облікових даних. Наявність централізованого інструменту для виконання цих дій дозволяє швидко відновлювати доступ абітурієнтів до системи без залучення технічних спеціалістів, зменшує навантаження на службу підтримки та підвищує загальний рівень зручності й безпеки використання платформи під час вступної кампанії(рис. 3.30).

Унікальні особи, що подавали заявки

Показані 1-20 із 7 776 записів.

#		Прізвище	Ім'я	По батькові	Освітній ступінь	Дата народження	Логін (Email)
1		Стремільов	Федір	Володимирович	Бакалавр	1986-07-05	fstremilov@gmail.com
2		Безпалый	Олександр	Русланович	Доктор філософії	2001-07-10	bezpaluy2001@gmail.com
3		Мещеряков	Ян	Ярославович	Магістр	2000-12-31	dwho5136@gmail.com

Рисунок 3.30. Сторінка з списком абітурієнтів

Додатково важливою складовою системи є наявність окремої сторінки для автоматизованого генерування облікових записів Office 365 для користувачів платформи. Така сторінка забезпечує централізоване створення корпоративних акаунтів із заданими параметрами доступу, що значно спрощує процес підключення студентів і працівників до хмарних сервісів університету. Передбачена можливість експорту сформованих облікових даних у форматі Excel, що дозволяє зручно передавати інформацію відповідальним підрозділам, зберігати резервні копії або використовувати дані для подальшої звітності та адміністрування. Реалізація цього функціоналу підвищує ефективність управління користувачами, зменшує ймовірність помилок під час масового створення акаунтів і забезпечує узгодженість роботи інформаційної системи з екосистемою Office 365(рис. 3.31).

Головна / Генерація нових акаунтів для студентів

2167
Створено акаунтів

457
Потребуєть генерації

Дата початку навчання
22. 12. 2025

Генерувати акаунти

Експортувати з дати
дд - мм - рррр

Експорт

Генерація нових акаунтів для студентів
Показані 1-20 із 2 167 записів.

Рисунок 3.31. Сторінка з генерацією та експортом

В системі передбачена окрема сторінка для перегляду логів, що забезпечує детальний моніторинг роботи платформи та дозволяє адміністраторам ефективно аналізувати події й помилки. На цій сторінці можна переглядати інформацію про перехід користувачів між URL-адресами, трасування помилок (trace), повідомлення про помилки, а також додаткові дані, такі як IP користувача, тип запиту та джерело переходу. Крім того, передбачено можливість перегляду логів, збережених у файлах, з підтримкою посторінкового або поетапного завантаження великих обсягів даних(рис. 3.32-3.40).

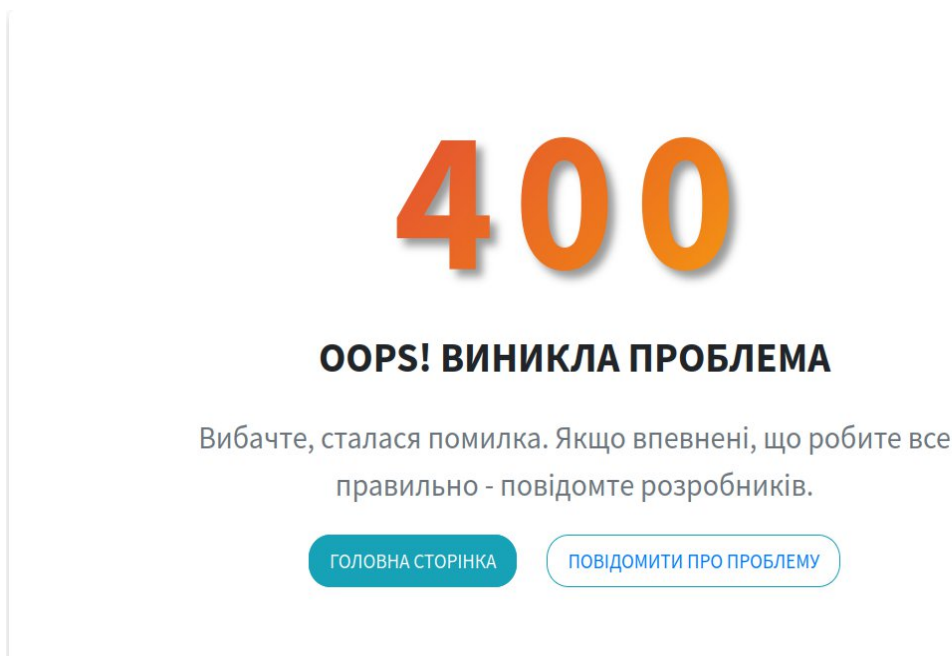


Рисунок 3.32. Сторінка з помилкою

Для контролю стабільності роботи системи адміністраторам надається доступ до журналу помилок. У журналі відображаються всі виняткові ситуації, що виникли під час роботи користувачів або виконання системних процесів. Інформація групується за типом помилки, часом виникнення та іншими параметрами, що дозволяє оперативно знаходити причини несправностей. На

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				65
Змн.	Арк.	№ докум.	Підпис	Дата		

рисунку 3.33 показано інтерфейс перегляду журналу помилок. Адміністратор може переглядати перелік зафіксованих подій, аналізувати їхню повторюваність та визначати пріоритетність усунення виявлених проблем.

Журнали помилок Показані 1-50 із 51 запису.									
#	Розгорнути	Деталі	Тип помилки	Код типу помилки	Користувачі	Метод запиту	Сторінка до	URL	Повідомлення
1	-		TypeError (500)	500	anatoli.kotsiur	GET	https://portal.nau.edu.ua/qualification-work/g...	https://portal.nau.edu.ua/qualification-work/g...	common/services/qualificationWork(QualificationWorkService:downloadFile)... : Argument #1 (\$filePath) ...
2	-		ErrorException (2)	2	8402244	POST	https://cabinet.nau.edu.ua/student/qualificatio...	https://cabinet.nau.edu.ua/student/qualificatio...	move_uploaded_file(/var/www/diguni/files/qualification-works/3018/... file_work.pdf.p7s): Failed to open...
3	-		ErrorException (2)	2	7037453	POST	https://cabinet.nau.edu.ua/student/qualificatio...	https://cabinet.nau.edu.ua/student/qualificatio...	move_uploaded_file(/var/www/diguni/files/qualification-works/2729/... file_work.pdf.p7s): Failed to open...
4	-		MethodNotAllowedHttpException (405)	405	larysa.voliarska	GET	-	https://portal.nau.edu.ua/site/logout	Method Not Allowed. This URL can only handle the following request metho... POST.
5	-		ErrorException (2)	2	6943460	POST	https://cabinet.nau.edu.ua/student/qualificatio...	https://cabinet.nau.edu.ua/student/qualificatio...	move_uploaded_file(/var/www/diguni/files/qualification-works/3382/... file_work.pdf.p7s): Failed to open...
6	-		TypeError (500)	500	4021789	POST	https://cabinet.nau.edu.ua/student/qualificatio...	https://cabinet.nau.edu.ua/student/qualificatio...	Unhandled match case 600
7	-		ErrorException (2)	2	3593754	POST	https://cabinet.nau.edu.ua/student/qualificatio...	https://cabinet.nau.edu.ua/student/qualificatio...	move_uploaded_file(/var/www/diguni/files/qualification-works/2814/... file_work.pdf.p7s): Failed to open...
8			MethodNotAllowedHttpException (405)	405	6870515	HEAD	-	https://cabinet.nau.edu.ua/student/qualificatio...	Method Not Allowed. This URL can only handle the following request metho... GET.
Дії ID Користувач Метод запиту Сторінка до URL									
2817 6870515 HEAD - https://cabinet.nau.edu.ua/student/qualification-work/download-document?qualificationW...									
2824 6870515 HEAD - https://cabinet.nau.edu.ua/student/qualification-work/download-document?qualificationW...									

Рисунок 3.33. Сторінка логування

Детальна інформація про помилку містить технічні відомості, необхідні для аналізу причин її виникнення. Зокрема відображаються повідомлення винятку, маршрут запиту, користувач, який виконував дію, та інші службові параметри.

Статистика по URL Всього 60 записів.	
URL	Кількість помилок
https://cabinet.nau.edu.ua/student/qualification-work/upload-pdf-file	12
https://portal.nau.edu.ua/session/exam-sheet-sign/get-exam-sheet-files-by-req	11
https://portal.nau.edu.ua/qualificationWork/qualification-work-view/view	10
https://portal.nau.edu.ua/educationalUnitDocuments/studyload-teachers/teacher	7
https://portal.kai.edu.ua/session/exam-sheet-sign/get-exam-sheet-files-by-req	6
https://cabinet.nau.edu.ua/student/qualification-work/upload-sign-file	5
https://portal.nau.edu.ua/session/generate/elective-exam-sheet/generate?canCreateEmpty=0	4
https://portal.nau.edu.ua/qualificationWork/qualification-work-view/view?id=2966	3
https://portal.nau.edu.ua/qualificationWork/qualification-work-view/view?id=3294	2
https://portal.nau.edu.ua/qualificationWork/qualification-work-view/view?id=2966&_pjax=%23kv-pjax-container-comission	2
https://portal.nau.edu.ua/qualificationWork/qualification-work-view/view?id=3294&_pjax=%23kv-pjax-container-comission	2
https://portal.nau.edu.ua/qualificationWork/qualification-work-view/view?id=3292&_pjax=%23kv-pjax-container-comission	2

Рисунок 3.34. Статистика по URL

Статистика за URL-адресами дозволяє визначати сторінки та функціональні модулі системи, на яких найчастіше виникають помилки або інші проблемні ситуації. Аналіз цієї інформації дає можливість виявляти найбільш критичні ділянки програмного забезпечення, що негативно впливають на стабільність роботи системи та користувацький досвід. На основі отриманих

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		
						66

статистичних даних розробники можуть формувати пріоритетний план усунення недоліків, спрямовуючи ресурси насамперед на виправлення найбільш проблемних компонентів. Крім того, така статистика дозволяє оцінювати ефективність уже виконаних виправлень і контролювати загальну динаміку покращення якості програмного забезпечення.

Трасування	Вирішено	Додаткові дані	Дата створення TF	Дата оновлення
	Hi			
#0 /var/www/diguni/common/services/qualificationWork/QualificationWorkService.php(1320): common/serv...	✗ Hi	GET params: ["qualificationWorldId": "3021", "fileNam...	2025-12-21 14:19:32	2025-12-21 14:19:32
#0 [internal function]: yii\base\ErrorHandler->handleError(2, 'move_uploaded_f...', /var/www/diguni...	✗ Hi	GET params: [] POST params: ["qualificationWorldId"...	2025-12-21 10:56:03	2025-12-21 10:56:03
#0 [internal function]: yii\base\ErrorHandler->handleError(2, 'move_uploaded_f...', /var/www/diguni...	✗ Hi	GET params: [] POST params: ["qualificationWorldId"...	2025-12-21 10:33:19	2025-12-21 10:33:19
#0 [internal function]: yii\filters\VerbFilter->beforeAction(Object(yii\base\ActionEvent)) #1 /var/w...	✗ Hi	GET params: [] POST params: [] IP: 46.96.73.178 Us...	2025-12-21 10:05:46	2025-12-21 10:05:46
#0 [internal function]: yii\base\ErrorHandler->handleError(2, 'move_uploaded_f...', /var/www/diguni...	✗ Hi	GET params: [] POST params: ["qualificationWorldId"...	2025-12-20 18:55:29	2025-12-20 18:55:29
#0 [internal function]: frontend\controllers\student\QualificationWorkController->actionNavigator(2...	✗ Hi	GET params: ["qualificationWorldId": "2308"] POST pa...	2025-12-20 16:36:23	2025-12-20 16:36:23
#0 [internal function]: yii\base\ErrorHandler->handleError(2, 'move_uploaded_f...', /var/www/diguni...	✗ Hi	GET params: [] POST params: ["qualificationWorldId"...	2025-12-20 15:59:00	2025-12-20 15:59:00
#0 [internal function]: yii\filters\VerbFilter->beforeAction(Object(yii\base\ActionEvent)) #1 /var/w...	✗ Hi	GET params: ["qualificationWorldId": "2535", "fileNam...	2025-12-20 13:51:01	2025-12-20 17:26:53

Рисунок 3.35. Сторінка логування інші поля

Таблиця також містить додаткові поля, які надають детальну інформацію про кожну зафіксовану помилку. Зокрема, зберігаються дані про параметри запиту, що були передані до системи в момент виникнення помилки, дата та час її виникнення, а також інформація про трасування викликів (stack trace). Наявність таких відомостей значно спрощує процес діагностики та пошуку причин несправностей, оскільки дозволяє відтворити умови, за яких виникла помилка. Особливу цінність має трасування викликів, яке дає змогу визначити послідовність виконання методів та встановити конкретне місце в програмному коді, де виникла виняткова ситуація. Завдяки цьому розробники можуть оперативно локалізувати проблему та скоротити час, необхідний для її усунення.

Статистика помилок за типами Всього 8 записів.	
Тип помилки	Кількість виникнень
TypeError (500)	50
ErrorException (2)	35
BadRequestHttpException (400)	11
Exception (42000)	8
IntegrityException (23000)	7
MethodNotAllowedHttpException (405)	4
ErrorException (8192)	1
RangeNotSatisfiableHttpException (416)	1

Рисунок 3.36. Статистика помилок за типами

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				67
Змн.	Арк.	№ докум.	Підпис	Дата		

Також у системі реалізовано таблицю статистики помилок, яка відображає типи зафіксованих винятків та кількість їх виникнень за обраний період часу. Така інформація дозволяє отримати узагальнене уявлення про стабільність роботи програмного забезпечення та визначити найбільш поширені проблеми, що виникають під час експлуатації системи. Аналіз кількості помилок кожного типу дає змогу виявити компоненти, які потребують першочергового вдосконалення або додаткового тестування. Крім того, накопичення статистичних даних у динаміці дозволяє оцінювати ефективність внесених змін та контролювати процес підвищення якості програмного забезпечення. На основі отриманої статистики розробники можуть обґрунтовано визначати пріоритетність виправлення помилок і планувати подальші роботи з оптимізації та підтримки системи.

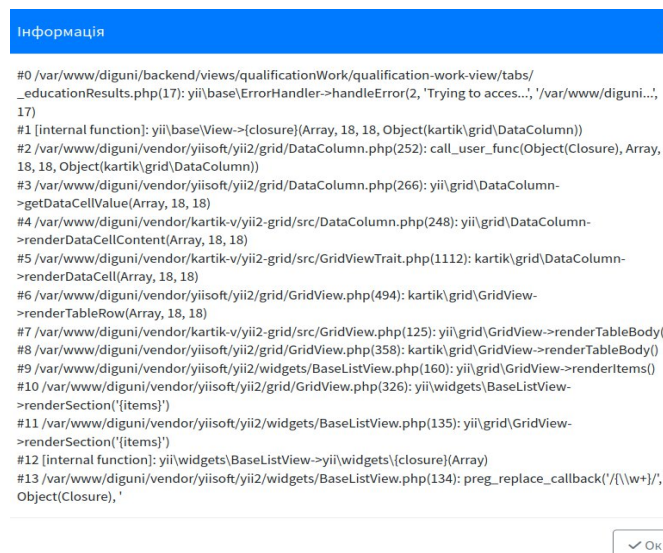


Рисунок 3.37. Модальне вікно при натисканні на поле трасування

При натисканні на будь-яке поле таблиці відкривається модальне вікно, яке містить детальну інформацію щодо вибраного значення. У ньому відображається розширений текст або додаткові дані, які не поміщаються у межах таблиці через обмеження ширини колонки, що дозволяє повністю переглянути вміст без скорочень або обрізання інформації.

Окрім збереження записів у базі даних, система підтримує роботу з файловими журналами. Це дозволяє отримувати інформацію навіть у випадках, коли помилка виникла до моменту збереження даних у базі.

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				68
Змн.	Арк.	№ докум.	Підпис	Дата		

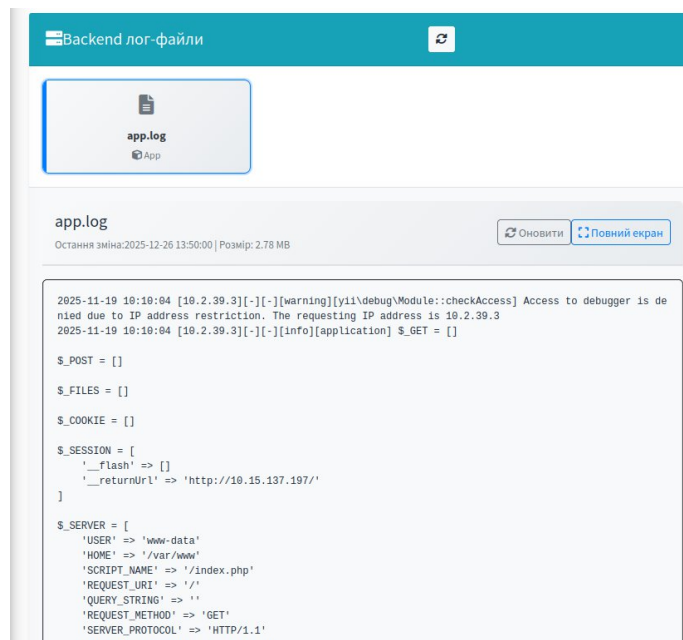


Рисунок 3.38. Логи з файлу

На рисунку 3.38 представлено сторінку перегляду доступних лог-файлів. Користувач може обрати необхідний журнал та перейти до його детального аналізу.

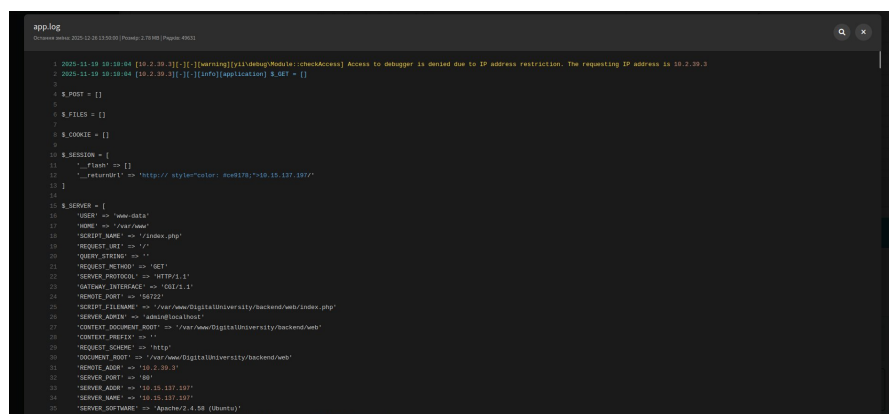


Рисунок 3.39. Модальне вікно для читання логів у повноекранному режимі

При відкритті у повноекранному режимі користувач може детально ознайомитися з вмістом лог-файлу, а також скористатися функцією пошуку для швидкого знаходження необхідного запису в межах усього файлу. Також було реалізовано механізм lazy-loading, який дозволяє завантажувати дані поступово, у міру необхідності, що зменшує початкове навантаження на систему та підвищує продуктивність під час роботи з великими файлами.

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				69
Змн.	Арк.	№ докум.	Підпис	Дата		

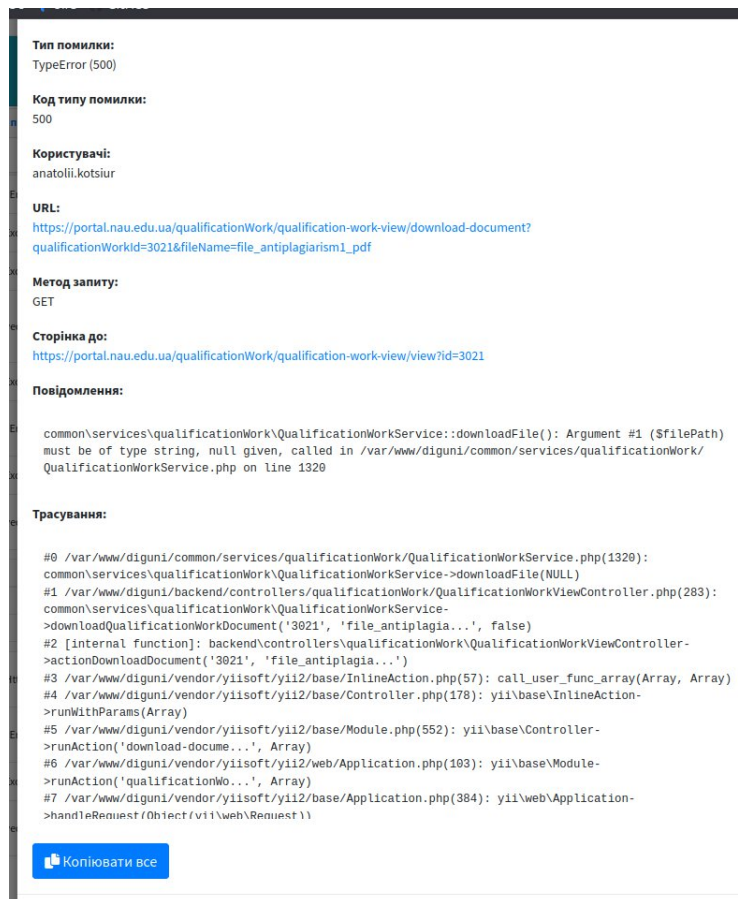


Рисунок 3.40. Модальне вікно для копіювання усієї інформації про помилку

Також реалізовано функціонал швидкого копіювання даних для створення квитків у Jira, що дозволяє оперативно переносити необхідну інформацію з логів або таблиці в задачу без ручного переписування. Це суттєво спрощує процес постановки задачі, зменшує ризик помилок при копіюванні та прискорює комунікацію між учасниками команди розробки. Завдяки цьому механізму користувач може швидко сформулювати структурований опис інциденту або задачі та одразу передати його в систему відстеження. Після створення задачі вона потрапляє до робочого процесу команди, що дозволяє негайно розпочати її виконання. Після завершення роботи із задачею передбачено можливість її подальшої обробки: користувач може позначити її як виконану для фіксації результату або видалити зі списку, якщо вона більше не є актуальною. Це забезпечує актуальність даних та допомагає підтримувати порядок у переліку задач.

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

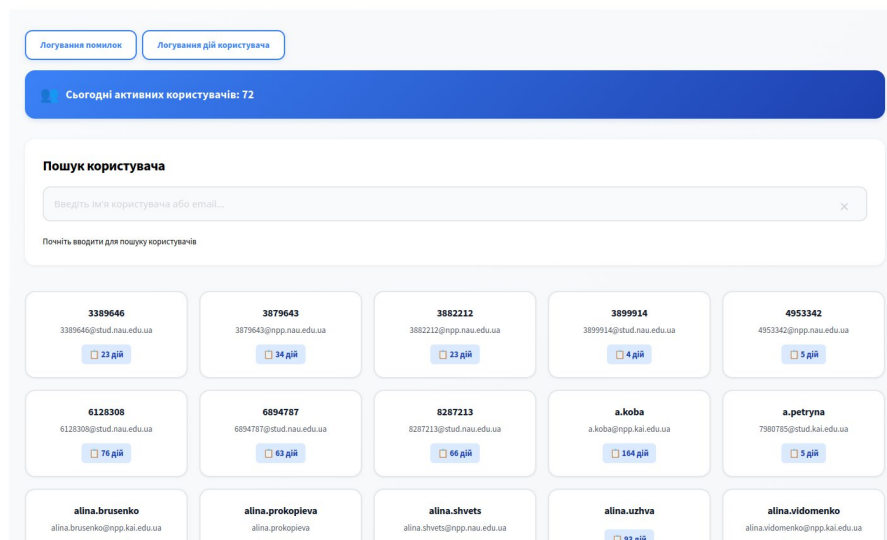


Рисунок 3.41. Пошук користувачів

Реалізовано пошук по користувачах, який дозволяє швидко знаходити потрібного користувача в системі за іменем або поштою. Після вибору користувача стає доступним перегляд його активності, зокрема дій, які він виконував на різних сторінках застосунку. Це включає історію взаємодій, переглядів та змін, що дає змогу аналізувати поведінку користувача та відстежувати послідовність його операцій у системі.

Логуювання дій користувачів										
Перегляд всіх дій всіх користувачів системи										
всього дій										
35191										
#	КОРИСТУВАЧ	USER ID	ДАТА ТА ЧАС ТР	МЕТОД	КОНТРОЛЕР	ДІЯ	МАРШРУТ	URL	IP АДРЕСА	USER AGENT
1	yurk.synhub	11367	24.02.2026 11:02:46	POST	session/exam-sheet	get-point-name-all	session/exam-sheet/get-point-name-all	https://portal.kai.edu.ua/session/exam-sheet/get-point-name-all	19.8.128.51	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:104.0) [Frodo Baggins]
2	o.tamashin	6621	24.02.2026 11:02:38	POST	publications/publication	create	publications/publication/create	https://portal.kai.edu.ua/publications/publication/create	81.144.217.104	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:104.0) [Frodo Baggins]
3	lyenka.v	627	24.02.2026 11:02:34	POST	qualification-work-view	navigator	qualification-work-view/navigator	https://portal.kai.edu.ua/qualification-work-view/navigator	5.248.244.81	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:104.0) [Frodo Baggins]

Рисунок 3.42. Перегляд дій усіх користувачів

Також наявний перегляд активності всіх користувачів системи, що дозволяє відстежувати дії на сайті, а саме виконаних POST-запитів, що були зафіксовані. Завдяки цьому забезпечується можливість аналізу дій користувачів, що дозволяє виявляти закономірності у використанні функціоналу системи, відстежувати ключові події.

Повернутися до списку

3389646

ID КОРИСТУВАЧА

12117

EMAIL

3389646@stud.nau.edu.ua

ПІБ

Пустовойт Руслан Олександрович

PERSON ID

3389646

Логи дій користувача

всього дій

23

дій за сьогодні

0

#	ДАТА ТА ЧАС ?	МЕТОД	КОНТРОЛЕР	ДІЯ	МАРШРУТ	URL	IP АДРЕСА	USER AGENT	ДАНІ ЗАПИТУ
1	16.02.2026 14:44:12	POST	session/exam-sheet	get-point-name-all	session/exam-sheet/get-point-name-all	https://portal.nau.edu.ua/session/exam-sheet/get-p...	10.0.128.11	Mozilla/5.0 (Windows NT 10.0; Win64; x64...	["get*":["post*"]]
2	16.02.2026 14:43:38	POST	session/exam-sheet	get-point-name-all	session/exam-sheet/get-point-name-all	https://portal.nau.edu.ua/session/exam-sheet/get-p...	10.0.128.11	Mozilla/5.0 (Windows NT 10.0; Win64; x64...	["get*":["post*"]]
3	16.02.2026 14:43:26	POST	session/exam-sheet	get-point-name-all	session/exam-sheet/get-point-name-all	https://portal.nau.edu.ua/session/exam-sheet/get-p...	10.0.128.11	Mozilla/5.0 (Windows NT 10.0; Win64; x64...	["get*":["post*"]]

Рисунок 3.43. Перегляд дій користувача

Також реалізовано можливість перегляду дій конкретного користувача, що дозволяє відстежувати його активність на сайті в динаміці. У цьому розділі відображаються виконані користувачем операції, які базуються на POST-запитах, що фіксують зміни стану системи та взаємодію з її основними функціями. Такий підхід дає змогу детально аналізувати послідовність дій користувача, виявляти характер його роботи з інтерфейсом та відстежувати ключові події. Це забезпечує більш цілісне уявлення про поведінку користувача в системі, а також може використовуватися для діагностики помилок, аудиту активності або аналізу інцидентів у межах роботи застосунку.

```

graph TD
    subgraph Docker_Conteiner [Docker Контейнер]
        Docker_Logs[Логи docker контейнера]
        subgraph Riven PHP [Рівень PHP]
            subgraph Logi Yii2 [Логи Yii2]
                User_Logs[Логи дій користувача]
            end
            Error_Logs[Логи помилок]
        end
    end

```

Рисунок 3.44. Структура та рівні логування

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				72
Змн.	Арк.	№ докум.	Підпис	Дата		

Діаграма відображає багаторівневу структуру системи логування, що поділена на три основні рівні: Docker-контейнер, PHP-застосунок та фреймворк Yii2. На рівні Docker-контейнера фіксуються системні логи роботи середовища виконання, які містять інформацію про стан контейнера та його процеси. На рівні PHP здійснюється обробка та збереження логів помилок, що виникають у процесі виконання застосунку. У межах рівня Yii2 реалізовано логування прикладних подій, зокрема дій користувачів у системі. Така багаторівнева архітектура дозволяє розділити системні, прикладні та користувацькі події, що спрощує діагностику помилок, аналіз роботи застосунку та підвищує ефективність моніторингу всієї системи.

Рисунок 3.45 відображає життєвий цикл логів у системі - від моменту ініціації додатку до фінального запису даних у базу. Архітектура побудована таким чином, щоб мінімізувати навантаження на базу даних та забезпечити стабільну роботу системи навіть при великій кількості користувацьких подій. Для досягнення цієї мети було обрано підхід із двоетапним збереженням даних: спочатку до файлу, а вже потім у базу даних через механізм фонових завдань.

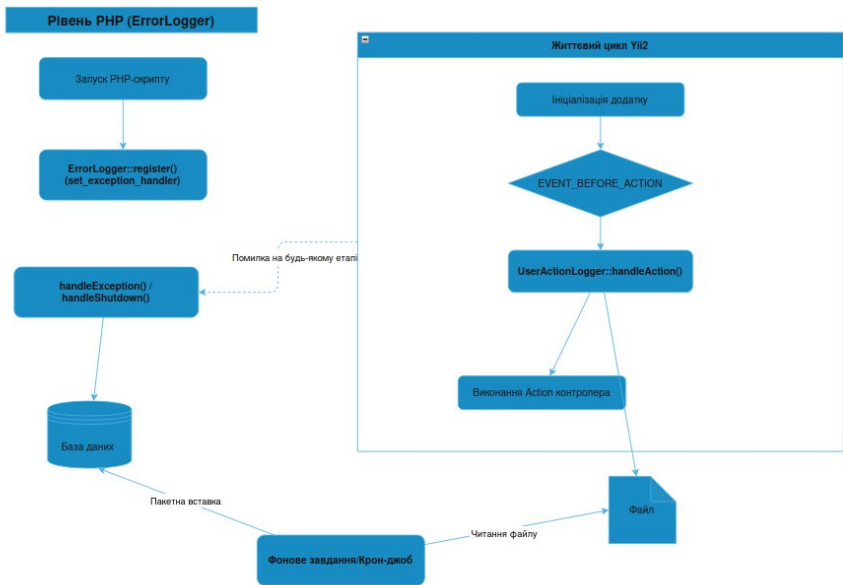


Рисунок 3.45. Діаграма логування

Як видно з діаграми, процес логування розпочинається на рівні PHP зі запуску скрипту, після чого виконується реєстрація обробника виключень. У

разі виникнення будь-якої помилки на будь-якому етапі роботи додатку спрацьовують відповідні обробники, які передають дані безпосередньо до бази даних.

Паралельно функціонує блок життєвого циклу фреймворку Yii2. Після ініціалізації додатку система очікує на відповідну подію, після настання якої викликається метод фіксації дій користувача. Після виконання контролера дані записуються у файл - це перший і швидкий етап збереження інформації.

Другий етап - перенесення накопичених даних до бази, що відбувається асинхронно через механізм фонових завдань. Така архітектура є ключовою з точки зору оптимізації: база даних не отримує прямого навантаження під час кожного запиту, а оновлюється пакетно у визначені проміжки часу, що підвищує продуктивність системи при пікових навантаженнях.

3.3. Тестування роботи розробленого додатку

У межах розробки системи для автоматизації управління освітнім процесом у закладі вищої освіти було проведено ручне функціональне тестування, яке передбачало перевірку основного функціоналу без використання автоматизованих тестових скриптів або фреймворків, що дозволило здійснити більш гнучку та сценарно-орієнтовану перевірку роботи системи в умовах, максимально наближених до реальної експлуатації. Такий підхід забезпечив можливість не лише виявити формальні помилки в роботі окремих модулів, але й оцінити поведінку інтерфейсу, логіку переходів між сторінками, узгодженість даних між різними компонентами системи, а також загальну зручність використання для різних категорій користувачів, включаючи студентів, абітурієнтів, викладачів та адміністративний персонал університету.

Основною метою тестування було підтвердження стабільності та коректності роботи системи в цілому, перевірка правильності реалізації ключових бізнес-процесів, а також відповідність функціональних модулів, зокрема «Авторизація» та «Логування», встановленим вимогам технічного

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				74
Змн.	Арк.	№ докум.	Підпис	Дата		

завдання та очікуванням замовника щодо безперебійної роботи платформи в умовах багатокористувацького навантаження.

У межах тестування було детально перевірено роботу основних механізмів автентифікації користувачів, зокрема входу через Office 365, Google OAuth2 та LDAP, при цьому особлива увага приділялася коректності обробки валідних та некоректних облікових даних, поведінці системи при повторних спробах входу, а також правильності перенаправлення користувача після успішної авторизації відповідно до призначеної ролі в системі. Під час перевірки було встановлено, що користувачі з коректними обліковими даними успішно проходять процедуру автентифікації, після чого система надає доступ до функціональних модулів відповідно до ролі користувача, забезпечуючи розмежування прав доступу та захист інформації. Для категорії вступників додатково було перевірено механізм роботи попередньо створених акаунтів, включаючи коректність ініціалізації профілю, а також автоматичне надсилання повідомлень з обліковими даними на зазначену електронну адресу.

Окремим етапом тестування стало перевірення процесів створення та адміністрування облікових записів викладачів через адміністративний інтерфейс системи, де відповідальна особа має можливість створювати нові акаунти, призначати ролі доступу, змінювати паролі користувачів, а також виконувати базову верифікацію наявності користувача в системі. Результати тестування підтвердили, що створені облікові записи коректно зберігаються в базі даних, правильно відображаються в інтерфейсі адміністратора, а також забезпечують доступ до функціоналу відповідно до визначених ролей і прав доступу без порушення логіки системи авторизації.

Також було перевірено функціональність редагування повідомлень для вступників, зокрема можливість зміни HTML-шаблонів листів, які використовуються для надсилання інструкцій, облікових даних та супровідної інформації. Система забезпечує коректну обробку змін у шаблонах, збереження оновленого контенту та подальше надсилання повідомлень із актуалізованим текстом, що підтверджує правильну роботу механізму динамічного формування

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				75
Змн.	Арк.	№ докум.	Підпис	Дата		

повідомлень та підтримку структурованих HTML-шаблонів із можливістю гнучкого налаштування блоків і стилів відображення.

Під час тестування функціоналу регенерації паролів було встановлено, що адміністратор має можливість ініціювати створення нового пароля для вступника, після чого система коректно оновлює відповідні дані, повторно формує повідомлення для користувача та забезпечує можливість зміни електронної адреси отримувача у разі необхідності. Усі виконані операції належним чином фіксуються в системі, що забезпечує прозорість дій адміністратора, а користувач отримує актуалізовані дані для подальшого входу в особистий кабінет без втрати доступу до системи.

Додатково було протестовано механізм масового створення облікових записів у середовищі Office 365, а також функціонал експорту даних користувачів у форматі Excel для подальшої обробки або передачі до інших підрозділів. Система коректно формує необхідні записи, забезпечує генерацію структурованого файлу з даними користувачів та гарантує його сумісність із табличними редакторами, що підтверджує придатність реалізованого функціоналу для масштабованого адміністрування великої кількості облікових записів.

Завершальним етапом ручного функціонального тестування стала перевірка модуля логування, який відповідає за фіксацію подій, помилок та дій користувачів у системі. Було встановлено, що система коректно записує інформацію про запити користувачів, включаючи URL-адреси, типи HTTP-запитів, повідомлення про помилки, стек трасування, IP-адреси та часові мітки подій. Логування реалізовано як на рівні бази даних, так і у файловому представленні, що забезпечує додаткову надійність зберігання даних. Також доступні узагальнені статистичні дані, які дозволяють аналізувати частоту виникнення помилок і характер навантаження на систему, що в цілому підтверджує стабільність роботи ключових модулів та готовність системи до експлуатації в реальному навчальному середовищі.

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				76
Змн.	Арк.	№ докум.	Підпис	Дата		

Висновки до третього розділу

У розділі було детально розглянуто інтерфейс та порядок роботи з системою онлайн-менеджменту освітніх процесів. Проведене тестування показало, що розроблений інтерфейс є інтуїтивно зрозумілим, адаптивним для різних типів пристроїв та забезпечує зручний доступ до всіх ключових функціональних модулів системи.

Реалізовані механізми автентифікації через LDAP, Google OAuth2 та Office 365 гарантують безпечний персоналізований доступ до платформи відповідно до ролі користувача. Наявність функціоналу для створення та управління акаунтами викладачів, редагування шаблонів повідомлень для абітурієнтів, регенерації паролів, зміни електронної адреси, а також централізованого генерування акаунтів Office 365 з експортом у Excel підвищує ефективність адміністрування та зручність роботи з системою.

Система логування забезпечує детальний контроль за діями користувачів, трасування помилок і перегляд статистики, що дозволяє своєчасно виявляти та усувати проблеми, підвищуючи надійність та стабільність роботи платформи.

Таким чином, модуль інтерфейсу та функціональних налаштувань успішно реалізує поставлені завдання: забезпечує безпечний доступ користувачів, ефективне адміністрування облікових записів та контроль за роботою системи, що підтверджується результатами ручного функціонального тестування. Впроваджений інтерфейс є зручним, надійним та готовим до використання в реальних умовах освітнього середовища.

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

ВИСНОВКИ

У межах виконання кваліфікаційної роботи було розроблено комплексну інформаційну систему онлайн-менеджменту освітніх процесів, яка забезпечує автоматизацію ключових функцій управління вступною кампанією та діяльністю закладу вищої освіти. Основна мета системи полягала у підвищенні ефективності адміністрування облікових записів користувачів, безпечної авторизації, централізованого логування та контролю за подіями, а також оптимізації взаємодії абітурієнтів, студентів і персоналу з цифровою платформою.

На етапі аналізу було визначено бізнес-вимоги та сформульовано функціональні й нефункціональні характеристики системи, що забезпечують надійний контроль доступу, безпеку персональних даних і стабільність роботи веб-порталу. Проведено оцінку архітектурних рішень і обрано монолітну структуру системи, що дозволяє централізовано керувати усіма компонентами, ефективно реалізовувати модулі автентифікації через LDAP, Google OAuth2 та Office 365, а також організувати логування винятків, помилок та дій користувачів.

Вибір інструментальних засобів, а саме PHP із фреймворком Yii2 для бекенду та СУБД MySQL для зберігання даних, забезпечив високу продуктивність, гнучкість розробки та інтеграцію з зовнішніми сервісами. Розроблено зручний інтерфейс користувача, адаптивний для різних пристроїв, з централізованою навігаційною панеллю та логічно згрупованими модулями, що підвищує ефективність роботи користувачів.

Система надає широкі можливості адміністрування: створення та управління акаунтами викладачів і абітурієнтів, редагування шаблонів повідомлень, регенерація паролів та зміна адрес отримувачів, пакетне генерування акаунтів Office 365 з експортом у Excel, а також перегляд детальних логів і статистики. Розроблений функціонал підвищує оперативність, зручність та безпеку управління обліковими записами, знижує ризики несанкціонованого доступу та сприяє прозорості освітніх процесів.

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				78
Змн.	Арк.	№ докум.	Підпис	Дата		

Ручне функціональне тестування підтвердило стабільність роботи системи, коректність виконання всіх модулів та відповідність реалізованого функціоналу встановленим вимогам. Реалізовані рішення забезпечують безпечний, ефективний і керований цифровий простір для користувачів різних категорій і створюють надійну основу для подальшого масштабування та інтеграції системи в освітнє середовище.

Таким чином, виконана робота охопила повний життєвий цикл розробки інформаційної системи: від аналізу вимог та проектування архітектури до впровадження, налаштування інтерфейсу та проведення тестування, що забезпечує готовність системи до використання у реальних умовах закладу вищої освіти.

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				79
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Цифровізація у вищій освіті: інституційні підходи до викладання та навчання / Семеняко Ю.Б., Брюховецька І.В., Бохонько О.Є. – Класичний приватний університет, 2023. – 186 с.
2. Платформа Moodle [Електронний ресурс] – Режим доступу до ресурсу: <https://moodle.org/>
3. Canvas [Електронний ресурс] – Режим доступу до ресурсу: <https://www.instructure.com/canvas>
4. OpenEduCat [Електронний ресурс] – Режим доступу до ресурсу: <https://www.openeducat.org/>
5. The Definitive Guide to Yii 2.0 [Електронний ресурс] – Режим доступу до ресурсу: <https://www.yiiframework.com/doc/guide/2.0/en>
6. Data Structures and Algorithm Analysis in PHP / Michael A. Redmond – Morgan Kaufmann, 2014. – 400 с.
7. PHP Cookbook: Modern Code Solutions for Professional Developers / Eric Mann - O'Reilly Media, 2023 – 420 с.
8. Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript and Web Graphics / Jennifer Niederst Robbins – O'Reilly, 2021. – 640 с.
9. Introduction to Algorithms / Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein – MIT Press, 2022. – 1312 с.
10. Algorithms in a Nutshell / George T. Heineman, Gary Pollice, Stanley Selkow – O'Reilly, 2017. – 800 с.
11. MySQL Crash Course / Ben Forta – Addison-Wesley, 2024. – 400 с.
12. PHP, MySQL and Apache / Julie C. Meloni – Peachpit Press, 2022. – 512 с.
13. High Performance MySQL: Optimization, Backups, Replication & Load Balancing / Baron Schwartz, Peter Zaitsev, Vadim Tkachenko, Jeremy Zawodny – O'Reilly, 2023. – 1024 с.
14. The Art of Unit Testing / Roy Osherove – Manning Publications, 2013. – 320 с.

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				80
Змн.	Арк.	№ докум.	Підпис	Дата		

15. Pro PHP Application Performance: Tuning PHP, MySQL and Apache / Armando Padilla – Apress, 2011. – 240 с.
16. Ajax: The Complete Reference / Thomas Powell – McGraw-Hill (Complete Reference), 2007. – 560 с.
17. Creating Software with Modern Diagramming Techniques / Matthew Skelton, Alexander Paisley – Pragmatic Bookshelf, 2021. – 280 с.
18. PHP Excel Programming with OpenTBS / Daniel Opitz – Leanpub, 2020. – 154 с.
19. UML Distilled: A Brief Guide to the Standard Object Modeling Language / Martin Fowler – Addison-Wesley, 2018. – 208 с.
20. Software Requirements / Karl Wieggers, Joy Beatty – Microsoft Press, 2014. – 648 с.
21. How Browsers Work: Behind the scenes of modern web browsers [Электронный ресурс] – Режим доступа до ресурсу: <https://www.html5rocks.com/en/tutorials/internals/howbrowserswork/>
22. Introducing JSON [Электронный ресурс] – Режим доступа до ресурсу: <https://www.json.org/json-en.html>
23. OpenLDAP Software [Электронный ресурс] – Режим доступа: <https://www.openldap.org/>
24. PHP LDAP Documentation [Электронный ресурс] – Режим доступа: <https://www.php.net/manual/en/book.ldap.php>
25. LDAP: Technical Specification [Электронный ресурс] – Режим доступа: <https://ldap.com/>
26. Microsoft Office 365 Developer Documentation [Электронный ресурс] – Режим доступа: <https://learn.microsoft.com/en-us/office/developer/>
27. Microsoft Graph API [Электронный ресурс] – Режим доступа: <https://learn.microsoft.com/en-us/graph/overview>
28. Microsoft Identity Platform: OAuth 2.0 and OpenID Connect [Электронный ресурс] – Режим доступа: <https://learn.microsoft.com/en-us/azure/active-directory/develop/v2-protocols-oidc>

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				81
Змн.	Арк.	№ докум.	Підпис	Дата		

29. Google People API [Електронний ресурс] – Режим доступу:
<https://developers.google.com/people>
30. Google Gmail API [Електронний ресурс] – Режим доступу:
<https://developers.google.com/gmail/api>
31. Google Identity Platform: OAuth 2.0 [Електронний ресурс] – Режим доступу: <https://developers.google.com/identity/protocols/oauth2>

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				82
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		83

```

class Controller extends Controller
{
    /**
     * @return array[]
     */
    public function behaviors(): array
    {
        return [
            'access' => [
                'class' => AccessControl::class,
                'rules' => [
                    [
                        'allow' => true,
                        'actions' => [
                            'index',
                            'delete',
                            'resolve',
                            'delete-group',
                            'get-log-content',
                            'get-log-files',
                            'get-system-log',
                        ],
                        'roles' => ['@']
                    ],
                    [
                        'allow' => false
                    ]
                ]
            ]
        ];
    }

    /**
     * @return string
     */
    public function actionIndex(): string
    {
        $searchModel = new ErrorLogSearch();
        $dataProvider = $searchModel->search(Yii::$app->request->queryParams);

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

    /**
     * @param $id
     * @return Response
     * @throws Exception
     * @throws NotFoundHttpException

```

```

*/
public function actionDelete($id): Response
{
    $this->findModel($id)->deleteWithRelated();
    return $this->redirect(['index']);
}

/**
 * @return Response
 */
public function actionDeleteGroup(): Response
{
    $ids = Yii::$app->request->get('ids');

    if ($ids) {
        $idArray = is_array($ids) ? $ids : explode(',', $ids);
        ErrorLog::deleteAll(['id' => $idArray]);
    }

    return $this->redirect(['index']);
}

/**
 * @return Response
 */
public function actionResolve(): Response
{
    $ids = Yii::$app->request->get('ids');
    $current = Yii::$app->request->get('current');

    if ($ids !== null && $current !== null) {
        $idArray = is_array($ids) ? $ids : explode(',', $ids);

        $newValue = $current ? 0 : 1;

        ErrorLog::updateAll(
            ['resolved' => $newValue],
            ['id' => $idArray]
        );
    }

    return $this->redirect(['index']);
}

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

    $backendLogs = glob(Yii::getAlias('@backend/runtime/logs/*.log'));

```

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

```

$frontendLogs = glob(Yii::getAlias('@frontend/runtime/logs/*.log'));
return [
    'backend' => array_map('basename', $backendLogs),
    'frontend' => array_map('basename', $frontendLogs),
];
}

/**
 * @return Response
 */
public function actionGetLogContent(): Response
{
    $type = Yii::$app->request->get('type');
    $filename = Yii::$app->request->get('filename');
    $limit = Yii::$app->request->get('limit', 15000);
    $offset = Yii::$app->request->get('offset', 0);

    if (!in_array($type, ['backend', 'frontend']) || empty($filename)) {
        return $this->asJson(['error' => 'Invalid parameters']);
    }

    $alias = $type === 'backend' ? '@backend/runtime/logs/' : '@frontend/runtime/logs/';
    $filePath = Yii::getAlias($alias . $filename);

    if (!file_exists($filePath) || !is_readable($filePath)) {
        return $this->asJson(['error' => 'File not found or not readable']);
    }

    if (pathinfo($filePath, PATHINFO_EXTENSION) !== 'log') {
        return $this->asJson(['error' => 'Invalid file type']);
    }

    $content = file_get_contents($filePath);
    $lines = explode("\n", $content);
    $totalLines = count($lines);

    $remainingLines = max(0, $totalLines - $offset - $limit);

    $chunkLines = array_slice($lines, $offset, $limit);
    $newOffset = $offset + count($chunkLines);

    $chunkContent = "";
    $startLineNumber = $offset + 1;

    foreach ($chunkLines as $index => $line) {
        $lineNumber = $startLineNumber + $index;
        $chunkContent .= str_pad($lineNumber, 6, ' ', STR_PAD_LEFT) . ': ' . $line . "\n";
    }

    return $this->asJson([
        'success' => true,
        'filename' => $filename,
    ]);
}

```

		Зубик М.Р.			ІІІ3.КР.Б – 121 – 26 – ІІ3	Арк.
		Морозов А.В.				86
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        'content' => $chunkContent,
        'lastModified' => date('Y-m-d H:i:s', filemtime($filePath)),
        'size' => filesize($filePath),
        'totalLines' => $totalLines,
        'offset' => $newOffset,
        'remainingLines' => $remainingLines,
        'hasMore' => $remainingLines > 0
    });
}

/**
 * Finds the ErrorLog model based on its primary key value.
 * If the model is not found, a 404 HTTP exception will be thrown.
 * @param integer|string $id
 * @return ErrorLog the loaded model
 * @throws NotFoundHttpException if the model cannot be found
 */
protected function findModel(int|string $id): ErrorLog
{
    if (($model = ErrorLog::findOne($id)) !== null) {
        return $model;
    } else {
        throw new NotFoundHttpException('The requested page does not exist.');
```

```

    }
}

class GoogleController extends Controller
{
    /**
     * @return array[]
     */
    public function behaviors(): array
    {
        return [
            'verbs' => [
                'class' => VerbFilter::class,
                'actions' => [
                    'delete' => ['post'],
                ],
            ],
            'access' => [
                'class' => AccessControl::class,
                'rules' => [
                    [
                        'allow' => true,
                        'actions' => ['index', 'view', 'create', 'update', 'delete'],
                        'roles' => ['@']
                    ],
                    [
                        'allow' => false
                    ]
                ]
            ]
        ];
    }
}

```

```

        ]
    ]
];
}

/**
 * @return string
 */
public function actionIndex(): string
{
    $searchModel = new GoogleAuthConfigSearch();
    $dataProvider = $searchModel->search(Yii::$app->request->queryParams);

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

/**
 * @return Response|string
 * @throws Exception
 */
public function actionCreate(): Response|string
{
    $model = new GoogleAuthConfig();

    if ($model->load(Yii::$app->request->post()) && $model->save(false)) {
        return $this->redirect(['index']);
    } else {
        return $this->render('create', [
            'model' => $model,
        ]);
    }
}

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

    if ($model->load(Yii::$app->request->post()) && $model->save(false)) {
        return $this->redirect(['index']);
    } else {
        return $this->render('update', [
            'model' => $model,
        ]);
    }
}

```

```

    }
}

/**
 * @param $id
 * @return Response
 * @throws NotFoundException
 * @throws Throwable
 * @throws StaleObjectException
 */
public function actionDelete($id): Response
{
    $this->findModel($id)->delete();

    return $this->redirect(['index']);
}

/**
 * @param $id
 * @return GoogleAuthConfig|null
 * @throws NotFoundException
 */
protected function findModel($id): ?GoogleAuthConfig
{
    if (($model = GoogleAuthConfig::findOne($id)) !== null) {
        return $model;
    } else {
        throw new NotFoundException('The requested page does not exist.');
```

```

    }
}
class GoogleSearch extends GoogleAuthConfig
{
    /**
     * @inheritdoc
     */
    public function rules(): array
    {
        return [
            [['id'], 'integer'],
            [['client_id', 'client_secret', 'redirect_uri', 'scope'], 'safe'],
        ];
    }

    /**
     * @inheritdoc
     */
    public function scenarios(): array
    {
        return Model::scenarios();
    }
}

```

```

/**
 * Base query for search
 *
 * @return ActiveQuery
 */
protected function baseQuery(): ActiveQuery
{
    return GoogleAuthConfig::find()->orderBy(['id' => SORT_ASC]);
}

/**
 * @param ActiveQuery $query
 * @return ActiveDataProvider
 */
protected function createDataProvider(ActiveQuery $query): ActiveDataProvider
{
    return new ActiveDataProvider([
        'query' => $query,
    ]);
}

/**
 * @param ActiveQuery $query
 * @return void
 */
protected function applyFilters(ActiveQuery $query): void
{
    $query->andWhere([
        'id' => $this->id,
    ]);

    $query->andWhere(['like', 'client_id', $this->client_id])
        ->andWhere(['like', 'client_secret', $this->client_secret])
        ->andWhere(['like', 'redirect_uri', $this->redirect_uri])
        ->andWhere(['like', 'scope', $this->scope]);
}

/**
 * @param array $params
 * @return ActiveDataProvider
 */
public function search(array $params): ActiveDataProvider
{
    $query = $this->baseQuery();
    $dataProvider = $this->createDataProvider($query);

    $this->load($params);

    if (!$this->validate()) {
        return $dataProvider;
    }
}

```

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Морозов А.В.				90
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        $this->applyFilters($query);

        return $dataProvider;
    }
}

class AdmissionCampaign2025RejectedPropositionController extends Controller
{
    /**
     * @return array[]
     */
    public function behaviors(): array
    {
        return [
            'access' => [
                'class' => AccessControl::class,
                'rules' => [
                    [
                        'allow' => true,
                        'actions' => ['index'],
                        'roles' => ['@']
                    ],
                    [
                        'allow' => false
                    ]
                ]
            ]
        ];
    }

    /**
     * @return mixed
     */
    public function actionIndex(): mixed
    {
        $searchModel = new AdmissionCampaign2025RejectedPropositionExclusionSearch();
        $dataProvider = $searchModel->search(Yii::$app->request->queryParams);

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

    /**
     * @param $id
     * @return AdmissionCampaign2025RejectedPropositionExclusion|null
     * @throws NotFoundHttpException
     */
    protected function
    findModel($id): ?AdmissionCampaign2025RejectedPropositionExclusion
    {

```

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Морозов А.В.				91
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        if (($model =
AdmissionCampaign2025RejectedPropositionExclusion::findOne($id)) !== null) {
            return $model;
        } else {
            throw new NotFoundException('The requested page does not exist.');
```

```

    }
}

class AdmissionCampaign2025RejectedRequestController extends Controller
{

```

```

    /**

```

```

     * @return array[]

```

```

     */

```

```

    public function behaviors(): array
    {

```

```

        {

```

```

            return [

```

```

                'access' => [

```

```

                    'class' => AccessControl::class,

```

```

                    'rules' => [

```

```

                        [

```

```

                            'allow' => true,

```

```

                            'actions' => ['index'],

```

```

                            'roles' => ['@']

```

```

                        ],

```

```

                        [

```

```

                            'allow' => false

```

```

                        ]

```

```

                    ]

```

```

                ];

```

```

            }

```

```

        }

```

```

    /**

```

```

     * @return string

```

```

     */

```

```

    public function actionIndex(): string
    {

```

```

        {

```

```

            $searchModel = new AdmissionCampaign2025RejectedRequestSearch();

```

```

            $dataProvider = $searchModel->search(Yii::$app->request->queryParams);

```

```


```

```


```

```

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

```

```

                'searchModel' => $searchModel,

```

```

                'dataProvider' => $dataProvider,

```

```

            ]);

```

```

        }

```

```

    /**

```

```

     * @param $id

```

```

     * @return AdmissionCampaign2025RejectedRequest|null

```

```

     */

```

		Зубик М.Р.			ІІІЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Морозов А.В.				92
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    * @throws NotFoundException
    */
protected function findModel($id): ?AdmissionCampaign2025RejectedRequest
{
    if (($model = AdmissionCampaign2025RejectedRequest::findOne($id)) !== null) {
        return $model;
    } else {
        throw new NotFoundException('The requested page does not exist.');
```

```

    }
}

class Office365ApiEndpointsController extends Controller
{
    /**
     * @return array[]
     */
    public function behaviors(): array
    {
        return [
            'verbs' => [
                'class' => VerbFilter::class,
                'actions' => [
                    'delete' => ['post'],
                ],
            ],
            'access' => [
                'class' => AccessControl::class,
                'rules' => [
                    [
                        'allow' => true,
                        'actions' => ['index', 'create', 'update', 'delete'],
                        'roles' => ['@']
                    ],
                    [
                        'allow' => false
                    ]
                ]
            ]
        ];
    }

    /**
     * @return string
     */
    public function actionIndex(): string
    {
        $searchModel = new Office365ApiEndpointsSearch();
        $dataProvider = $searchModel->search(Yii::$app->request->queryParams);

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

```

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ІІЗ	Арк.
		Морозов А.В.				93
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        'dataProvider' => $dataProvider,
    );
}

/**
 * @return Response|string
 * @throws Exception
 */
public function actionCreate(): Response|string
{
    $model = new Office365ApiEndpoints();

    if ($model->loadAll(Yii::$app->request->post()) && $model->save()) {
        return $this->redirect(['index']);
    } else {
        return $this->render('create', [
            'model' => $model,
        ]);
    }
}

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

    if ($model->loadAll(Yii::$app->request->post()) && $model->save()) {
        return $this->redirect(['index']);
    } else {
        return $this->render('update', [
            'model' => $model,
        ]);
    }
}

/**
 * @param $id
 * @return Response
 * @throws NotFoundHttpException
 * @throws Exception
 */
public function actionDelete($id): Response
{
    $this->findModel($id)->deleteWithRelated();

    return $this->redirect(['index']);
}

```

```

/**
 * @param $id
 * @return Office365ApiEndpoints|null
 * @throws NotFoundException
 */
protected function findModel($id): ?Office365ApiEndpoints
{
    if (($model = Office365ApiEndpoints::findOne($id)) !== null) {
        return $model;
    } else {
        throw new NotFoundException('The requested page does not exist.');
```

		Зубик М.Р.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		95