

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:

«Розробка інформаційної системи онлайн-менеджменту освітніх процесів "Digital University UA": модуль "Здобувачі освіти" та DevOps-інфраструктура»

Виконав студент групи ІПЗ-21-2
ЦАЛКО Ярослав Володимирович

Керівник роботи:
МОРОЗОВ Андрій Васильович

Рецензент:
ЄФРЕМОВ Юрій Миколайович

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

«ЗАТВЕРДЖУЮ»

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

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **ЦАЛКО Ярослав Володимирович**

Керівник роботи: **МОРОЗОВ Андрій Васильович**

Тема роботи: **«Розробка інформаційної системи онлайн-менеджменту освітніх процесів "Digital University UA": модуль "Здобувачі освіти" та DevOps-інфраструктура»** затверджена Наказом закладу вищої освіти від **«14» лютого 2025 р., №61/с**

Вихідні дані для роботи: **Розробити систему взаємодії зі здобувачами освіти та впровадити DevOps-інфраструктуру для підвищення ефективності розробки інформаційної системи онлайн-менеджменту освітніх процесів "Digital University UA".**

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

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

РЕФЕРАТ

Кваліфікаційна робота включає модуль «Здобувачі освіти» та впровадження DevOps-інфраструктури в рамках розробки інформаційної системи онлайн-менеджменту освітніх процесів «Digital University UA». Пояснювальна записка до кваліфікаційної роботи містить 78 сторінок, 24 ілюстрації та 1 таблицю.

Метою роботи є створення модулю, що надасть повний функціонал для адміністрації університету, який забезпечуватиме роботу із великими обсягами даних студентів, здійсненням контролю груп, синхронізацію даних із Єдиною державною електронною базою освіти (ЄДЕБО) проведення випускових операцій.

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

КЛЮЧОВІ СЛОВА: DEVOPS, YII2, СИСТЕМА, АДМІНІСТРАЦІЯ, БАЗА ДАНИХ, ОСВІТА

					ІПЗ.КР.Б – 121 – 25 – ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	Розробка інформаційної системи онлайн-менеджменту освітніх процесів Digital University UA: модуль "Здобувачі освіти" та DevOps-інфраструктура Пояснювальна записка	Літ.	Арк.	Аркушів
Розроб.		Цалко Я.В.						
Керівник		Морозов А. В.					3	78
Н. контр.						Житомирська політехніка, група ІПЗ-21-2		
Зав. каф.		Вакалюк Т.А.						

ABSTRACT

The qualification work includes the module “Education Applicants” and the implementation of DevOps infrastructure as part of the development of the information system for online management of educational processes “Digital University UA”. The explanatory note to the qualification work contains 78 pages, 24 illustrations and 1 table.

The aim of the work is to create a module that will provide full functionality for the university administration, which will ensure work with large volumes of student data, group control, data synchronization with the Unified State Electronic Education Database (UEDB) and graduation operations.

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

KEYWORDS: DEVOPS, YII2, SYSTEM, ADMINISTRATION, DATABASE, EDUCATION

		Цалко Я.В.			ПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

ЗМІСТ

РЕФЕРАТ.....	3
ABSTRACT	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	6
ВСТУП.....	7
РОЗДІЛ 1. АНАЛІЗ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ	9
1.1 Постановка задачі	9
1.2 Аналіз аналогів програмного продукту	10
1.3 Вибір архітектури системи.....	15
1.4 Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення	19
Висновки до першого розділу	20
РОЗДІЛ 2. ПРОЕКТУВАННЯ СИСТЕМИ ОНЛАЙН-МЕНЕДЖМЕНТУ ОСВІТНІХ ПРОЦЕСІВ.....	22
2.1 Визначення варіантів використання та структури системи.....	22
2.2 Обґрунтування вибору DevOps-інфраструктури	27
2.3 Розробка бази даних системи.....	30
2.4 Реалізація функціоналу системи управління навчальним процесом	32
Висновки до другого розділу	49
РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З СИСТЕМОЮ ОНЛАЙН-МЕНЕДЖМЕНТУ ОСВІТНІХ ПРОЦЕСІВ.....	50
3.1 Порядок встановлення та налаштування параметрів додатку	50
3.2 Структура інтерфейсу додатку.....	51
Висновки до третього розділу	62
ВИСНОВКИ	64
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	65
ДОДАТКИ.....	68
Додаток А	69

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД – база даних

ЗВО – заклад вищої освіти

CRUD – Create-Read-Update-Delete

СУБД – Система управління базою даних

ЄДЕБО – Єдина державна електронна база з питань освіти

CI/CD - Continuous Integration/Continuous Delivery

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

ВСТУП

У нинішньому світі оцифрування стало важливою частиною росту кожної області життя, і навчання — не є винятком. Оцифрування вузів — це не тільки введення нових технік, а плановий крок до створення дійсного доступного та новаторського середовища для освіти.

Спочатку, цифрові техніки дуже полегшують керування освітнім процесом. Автоматизація звичайних справ, таких, як облік відвідувань, складання розкладу, реєстрація на курси чи іспити — дає можливість адміністрації зосередитися на стратегічному розвитку університету.

По-друге, цифрові платформи дають студентам легкий доступ до навчальних матеріалів, результатів оцінок, онлайн-курсів і консультацій незалежно від місця. Це підвищить рівень академічної рухливості та персоналізації вчення.

Саме тому об'єктом розгляду стає інформаційна система онлайн-менеджменту освітніх процесів.

Метою даної роботи є розробка модуля «Здобувачі освіти» та впровадження DevOps інфраструктури. Встановлена мета обумовлює наступні завдання:

- аналіз процесів, пов'язаних з роботою зі здобувачами вищої освіти та уніфікація етапів задля універсального використання у інформаційній системі (надалі ІС);
- побудова оптимальної архітектури;
- обґрунтування засобів реалізації системи;
- розробка інформаційного та математичного забезпечення системи.
- аналіз процесів розробки ПЗ та знаходження способів вдосконалення інтеграції та розгортання оновленого програмного коду;
- побудова оптимальної процедури CI/CD процесів [23];
- розробка способу швидкого та макетного налаштування серверів клієнтів для розгортання ПЗ;
- інтегрування інструментів перевірки безпеки ПЗ.

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		7

Об'єкт дослідження – інформаційна система онлайн-менеджменту освітніх процесів «Digital University UA».

Предмет дослідження – технології, методи та засоби розробки модуля «Здобувачі освіти» та DevOps-інфраструктури інформаційної системи онлайн-менеджменту освітніх процесів «Digital University UA».

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		8

РОЗДІЛ 1. АНАЛІЗ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

1.1 Постановка задачі

Модуль «Здобувачі освіти» представляє собою частину веб платформи зі зручним для користувача інтерфейсом та серверною складовою, що складається з алгоритмів взаємодії з базою даних для управління, отримання інформації та синхронізації з ЄДЕБО списків здобувачів освіти університету.

Необхідна реалізація таких функцій:

1) Реалізація зручного інтерфейсу:

- враховуючи специфіку працівників закладів освіти, інтерфейс має бути простим та зрозумілим для усіх вікових категорій, а різні доступні дії повинні супроводжуватися відповідними повідомленнями про виконання;

2) Реалізація ядра модулю:

- створення таблиць та моделей для зберігання даних;
- реалізація сервісів для обрахунку та пошуку;
- розробка контролерів для обробки сторінок та дій;
- конфігурація та налаштування сталих показників модулю.

3) Налаштування DevOps інфраструктури:

- аналіз робочих процесів команди під час розробки ПЗ;
- вибір відповідних ПЗ для підтримки CI/CD;
- розробка скриптів та процесів автоматизації рутинних завдання;
- розробка платформи для розгортання ПЗ на сервері клієнта з підтримкою автоматичного CD;
- розробка скриптів для шаблонного налаштування системи у разі відмови клієнта від Docker;
- конфігурація вебсерверів Nginx/Apache, та генерація SSL сертифікатів.

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		9

1.2 Аналіз аналогів програмного продукту

Сьогодні на ринку існує чимало рішень, які частково або повністю охоплюють окремі аспекти освітнього процесу. Проте більшість із них орієнтовані на курс-центричну модель і не надають інструментів для повноцінного адміністрування освітнього контингенту чи взаємодії з державними реєстрами (як-от ЄДЕБО). Нижче наведено огляд найближчих аналогів.

Moodle — одна з найпопулярніших систем управління навчанням з відкритим вихідним кодом, яка застосовується в закладах освіти по всьому світу [2]. Вона дозволяє організовувати курси, завдання, оцінювання, форумну комунікацію та збирання домашніх завдань (рис. 1.1):

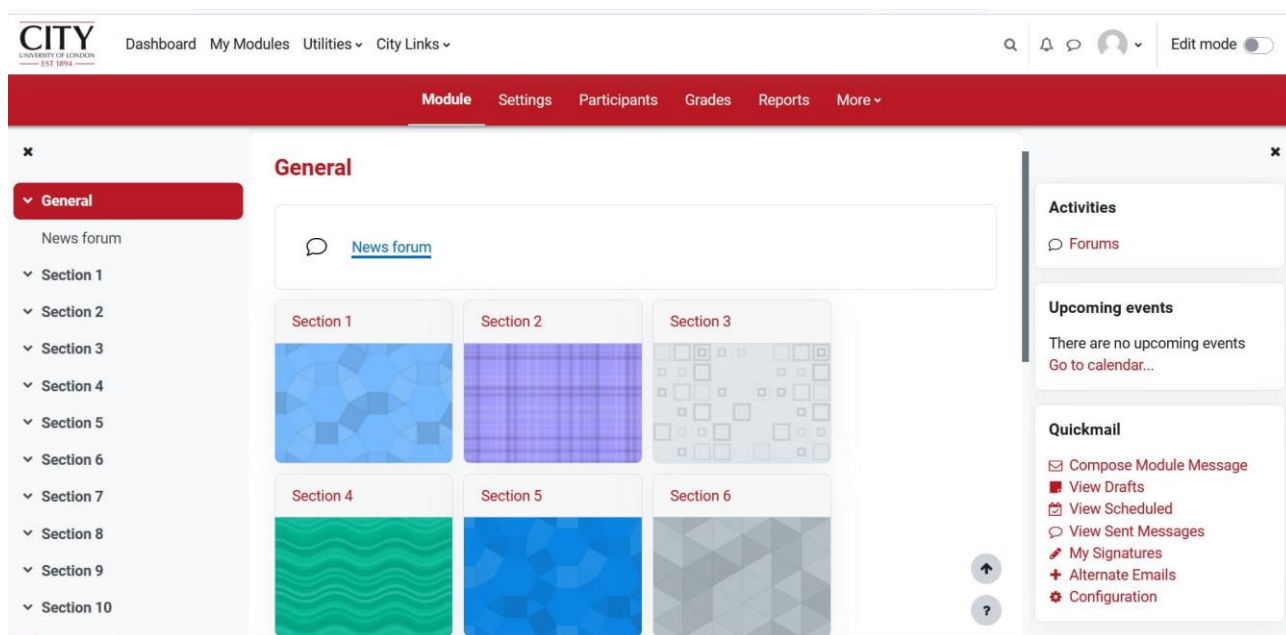


Рисунок 1.1 - Moodle

Переваги:

- відкритий код, можливість модифікації під власні потреби;
- багата екосистема плагінів (календарі, SCORM, Zoom, Google Meet, тощо);
- активна міжнародна спільнота, документація українською;
- підтримка великої кількості користувачів;
- зручні інструменти для асинхронного та синхронного навчання.

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		10

Недоліки:

- складність першого налаштування та адаптації під структуру ЗВО;
- не підтримує зведення офіційних відомостей чи ЄДЕБО;
- немає штатної підтримки підписання документів КЕП;
- застарілий дизайн без глибокої кастомізації;
- не передбачена робота з повноцінними академічними потоками чи відомостями з предметів.

Google Classroom – це безкоштовний інструмент Google для шкіл і університетів, що дозволяє керувати курсами, завданнями та взаємодією студентів і викладачів [4]. Підходить для дистанційного або змішаного навчання (рис. 1.2):

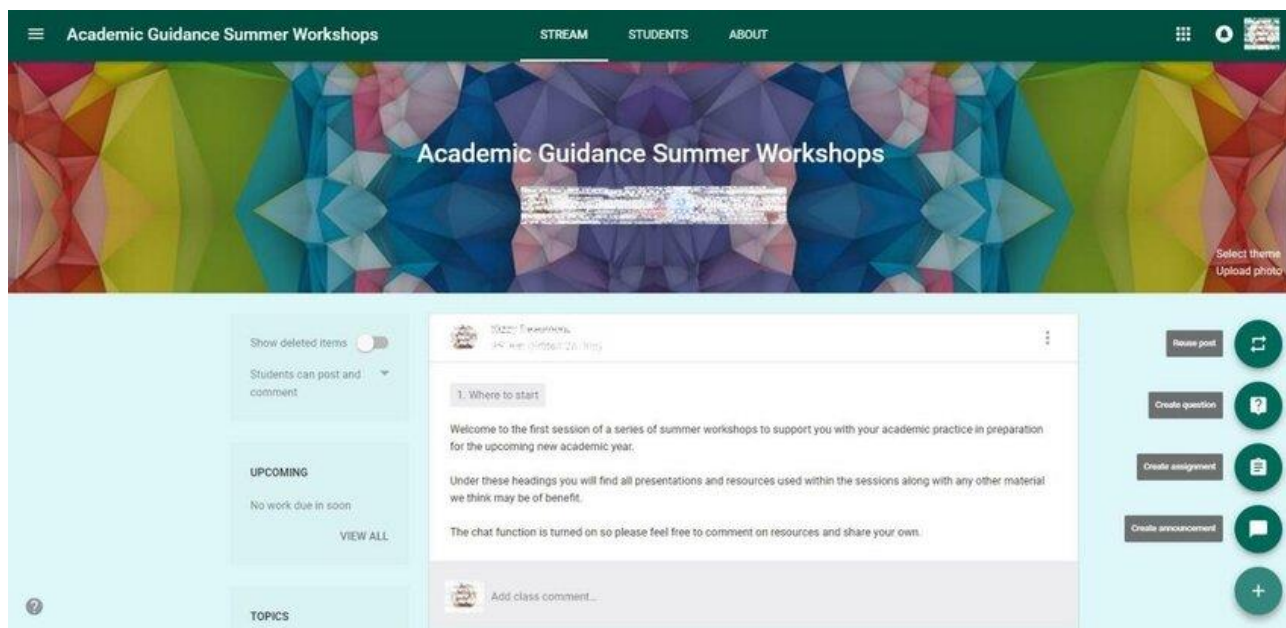


Рисунок 1.2 – Google Classroom

Переваги:

- проста та швидка реєстрація через акаунти Google;
- інтеграція з Google Drive, Docs, Calendar, Meet;
- push-сповіщення про дедлайни та нові завдання;
- мобільні додатки з чудовою адаптацією.

Недоліки:

- жорстка залежність від Google Workspace;

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

- відсутність модуля оцінювання з університетською специфікою (наприклад, бали за модулі, сесії);
- не підтримує підписання документів, генерацію зведених звітів;
- слабка кастомізація прав доступу;
- не передбачена робота з базами вступників чи інтеграція з національними реєстрами.

Blackboard Learn – це комерційна LMS-система, популярна у великих вишах Північної Америки, підтримує детальний моніторинг успішності, створення курсів, опитувань, аналітики [5] (рис. 1.3):

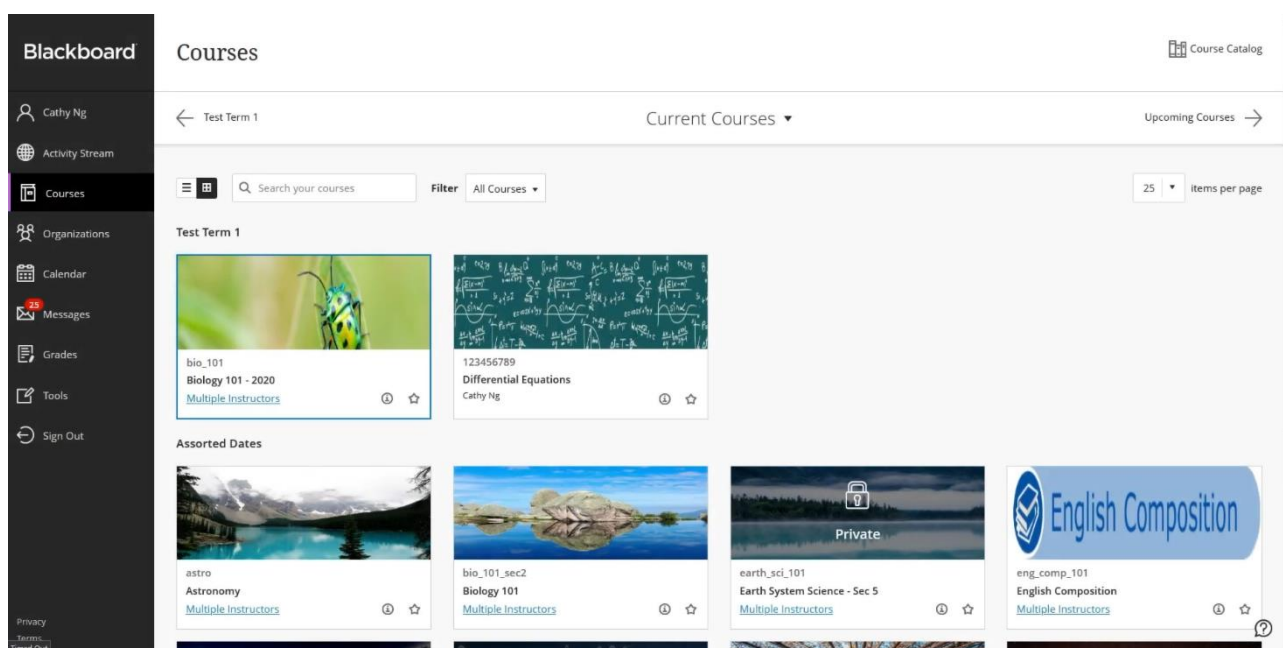


Рисунок 1.3 - Blackboard Learn

Переваги:

- потужні інструменти аналітики та динамічного відстеження результатів;
- інтеграція з SIS та ERP-системами;
- підтримка SCORM, LTI, та інтеграція з Microsoft Teams, Zoom;
- власні мобільні застосунки з фокусом на викладачів та студентів.

Недоліки:

- висока вартість ліцензії, обмеження для невеликих ЗВО;
- перевантажений інтерфейс, що потребує навчання персоналу;

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

- адаптація до локального законодавства вимагає додаткової розробки;
- залежність від зовнішніх серверів та складність розгортання на локальних ресурсах.

Canvas LMS – хмарна платформа для освітніх установ, яка фокусується на зручному інтерфейсі, мобільній доступності та сучасному API [3]. Активно використовується в американських університетах (рис. 1.4):

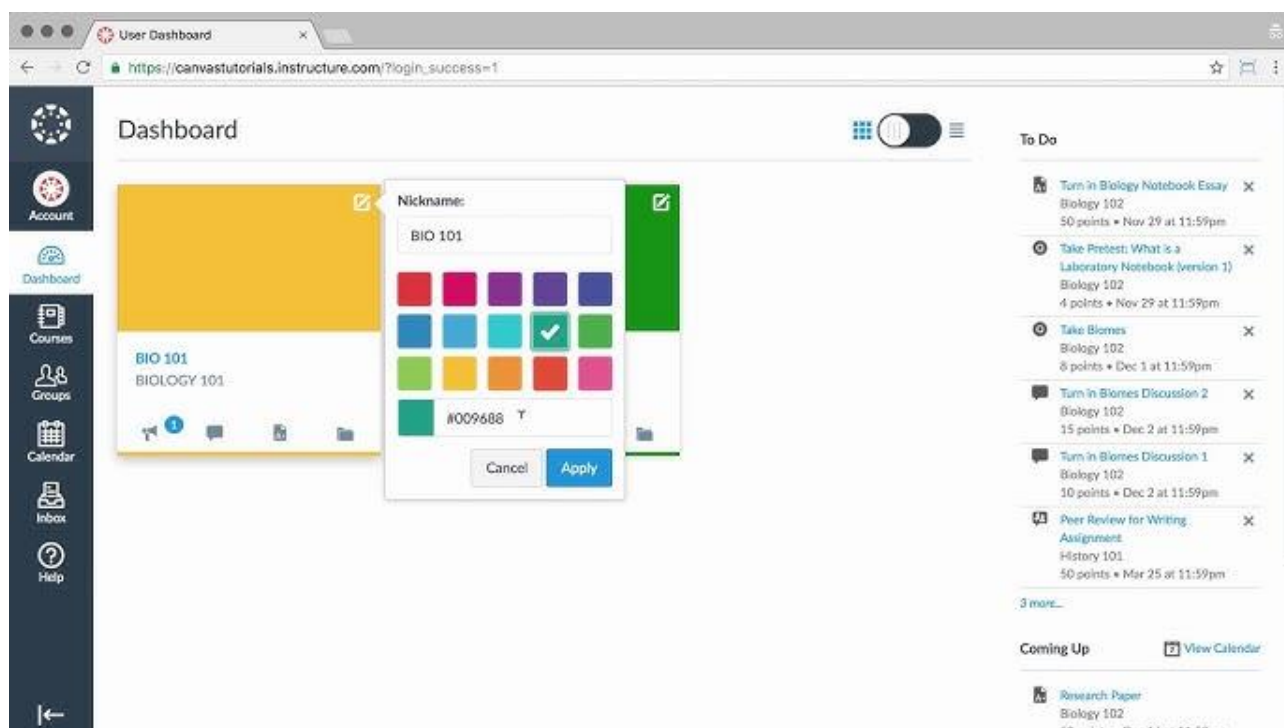


Рисунок 1.4 - Canvas LMS

Переваги:

- інтерфейс користувача розроблено за принципом mobile-first;
- вбудований підтримка SpeedGrader (перевірка завдань);
- гнучке налаштування курсів і доступів;
- можливість розгортання як у хмарі, так і локально (версія Canvas Open Source);
- REST API та підтримка інтеграцій через LTI.

Недоліки:

- комерційна модель з щомісячною оплатою (для повного функціоналу);
- обмежена локалізація для українських вишів;

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		13

- потребує налаштування політик безпеки згідно з українськими нормами.

Усі вищезгадані навчальні платформи було внесено до таблиці для здійснення порівняльної характеристики, за шкалою дуже добре (++), добре (+), погано (-) та дуже погано (--) (табл. 1.1):

Таблиця 1.1

Характеристика	Moodle	Google Classroom	Blackboard	Canvas
Вартість	++	++	--	--
Інтерфейс	-	++	-	++
Функціональність	++	-	++	++
Мобільна підтримка	+	++	-	++
Інтеграції	++	+	++	++
Складність впровадження	-	++	-	+
Підтримка спільноти	++	+	-	+
Гнучкість налаштувань	++	-	++	++
Ширина застосування	++	+	++	+
Вимоги до ресурсів	+	++	-	+

Як видно із табл. 1.1, кожна із систем має свої переваги, однак жодна не презентує повний набір функціоналу, що вкрай необхідний для правильного функціонування менеджменту освітніх процесів в університеті.

1.3 Вибір архітектури системи

При розробці платформи «Цифровий Університет», зокрема модуля «Здобувачі Освіти», ключовим аспектом є вибір архітектури системи, який впливає на масштабованість, продуктивність та можливості подальшої еволюції проекту. Основною дилемою є порівняння монолітного підходу та мікросервісної архітектури, що побудована на базі NodeJs [8] і працює у середовищі контейнерів Docker [20]. Розглянемо детально основні характеристики обох підходів, їх переваги, недоліки, а також як вони співвідносяться з вимогами сучасних цифрових платформ у вищих навчальних закладах.

Монолітна архітектура представляє собою єдину інтегровану систему, де всі компоненти – від логіки бізнесу до управління даними – розробляються та розгортаються як один єдиний додаток [12].

Переваги монолітного підходу:

- Простота розгортання: Завдяки однорідній структурі система розгортається як єдиний продукт. Це знижує складність CI/CD процесів, дозволяючи впроваджувати оновлення за мінімальний час.
- Цілісність коду та злагодженість: Оскільки всі модулі і компоненти знаходяться в одному додатку, розробникам легше координувати взаємодію між компонентами. Це особливо важливо на початкових етапах розробки, коли продукт ще формується.
- Спрощене налагодження та моніторинг: Відсутність розподіленої структури дозволяє зосередитися на вирішенні завдань всередині єдиного середовища. Завдяки використанню централізованих лог-систем та моніторингових інструментів можна оперативно виявляти та усувати недоліки.

Недоліки монолітного підходу:

- Обмеження масштабованості: Монолітний додаток важче адаптувати під зростання навантаження. Масштабування зазвичай відбувається

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		15

шляхом копіювання всієї системи, що може стати неефективним як з точки зору використання ресурсів, так і економічних витрат.

- Складність підтримки: В міру зростання кількості функціоналу кодова база стає все більш громіздкою. Додавання нових функцій або внесення змін у вже розроблену логіку може викликати негайні та непередбачувані залежності, що впливають на стабільність усієї системи.
- Обмежена адаптивність та гнучкість: Внесення змін в архітектурну структуру системи може бути надзвичайно складним процесом [13], особливо коли вимоги до масштабованості змінюються під час зростання кількості користувачів та обсягу даних.

Мікросервісний підхід базується на поділі системи на окремі сервіси, кожен з яких відповідає за певний функціональний блок [11]. Використання контейнеризації за допомогою Docker [21] дозволяє ізолювати сервіси один від одного, забезпечуючи їх незалежне розгортання та масштабування.

Переваги мікросервісної архітектури:

- Масштабованість: Кожен мікросервіс можна масштабувати окремо, що дозволяє гнучко використовувати ресурси під конкретні вимоги. Цей підхід сприяє оптимальному розподілу навантаження під час пікових навантажень, що дуже важливо для роботи зі студентською базою та великими обсягами даних.
- Гнучкість розробки та розгортання: Оскільки кожен сервіс працює незалежно, команди розробників можуть зосередитись на своїх доменних областях без впливу на решту системи. Це сприяє скороченню часу розробки, а також дозволяє використовувати різні технології для різних компонентів проекту.
- Полегшена підтримка та гарантія безпеки: Якщо один з сервісів виходить з ладу, це не впливає на роботу інших компонентів. Така ізоляція дозволяє оперативно виявляти проблемні місця та

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				16
Змн.	Арк.	№ докум.	Підпис	Дата		

впроваджувати спеціальні моніторингові інструменти для кожного сервісу окремо.

- Контейнеризація: Використання Docker забезпечує уніфіковане середовище для розгортання кожного сервісу [25]. Завдяки цьому процес розгортання стає стандартизованим і менш залежним від апаратного середовища, що сприяє підвищенню мобільності та швидкості масштабування.

Недоліки мікросервісного підходу:

- Складність інтеграції: Розподілена архітектура вимагає налагодження окремих каналів зв'язку між сервісами. Це може збільшити складність системи, особливо на початку розробки, коли потрібно визначити стандарти комунікації та протоколи передачі даних.
- Підвищені вимоги до DevOps: Для стабільної роботи мікросервісної системи потрібно ретельне налагодження процесів CI/CD, моніторингу та логування. Незважаючи на те, що контейнеризація спрощує розгортання, комплексна оркестрація та управління різними сервісами вимагають додаткових зусиль та інструментів.
- Зростання витрат на комунікацію: Через те, що сервіси взаємодіють один з одним через мережу, може виникати додаткове навантаження на інфраструктуру зв'язку. Неправильне налаштування або використання неефективних протоколів може призвести до затримок у роботі системи.

Розглядаючи вимоги до масштабованості, продуктивності та оперативної підтримки системи, можна відзначити, що використання мікросервісної архітектури на NodeJs у поєднанні з Docker є більш відповідним варіантом для проекту «Цифровий Університет». При виборі було зосереджено увагу на таких позитивних якостях поданого підходу:

- Підвищену гнучкість: Можливість незалежної розробки, розгортання та масштабування окремих компонентів дозволить швидко реагувати

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				17
Змн.	Арк.	№ докум.	Підпис	Дата		

на змінні вимоги користувачів, підтримуючи безперебійність роботи модуля «Здобувачі Освіти».

- Оптимізацію ресурсів: Завдяки ізоляції сервісів та їх незалежному масштабуванню забезпечується максимальна продуктивність навіть при зростанні навантаження.
- Зменшення ризиків: Ізоляція функціональності дозволяє ізолювати помилки окремих модулів, мінімізуючи ризик негативного впливу проблем однієї частини системи на роботу решти продукту.
- Покращення процесів DevOps: Налагодження CI/CD процесів для окремих мікросервісів дозволяє швидше впроваджувати оновлення та забезпечувати безперебійний процес розгортання, що вкрай необхідно для платформ з високими вимогами до продуктивності та безпеки.

Однак варто зазначити, що впровадження мікросервісної архітектури вимагає додаткових ресурсів для налагодження інтеграції, обрання стандартів комунікації меж компонентів та забезпечення стабільної роботи оркестрації.

Отже, на етапі початкової розробки необхідно ретельно зважити витрати на організацію процесів розгортання та підтримки проти переваг гнучкості, масштабованості та можливості індивідуального налаштування окремих сервісів.

З огляду на вищесказане, впровадження мікросервісної архітектури у проекті «Цифровий Університет» відповідає сучасним вимогам до управління великими базами даних і високої кількості користувачів. Такий підхід дозволить забезпечити адаптацію платформи до поступово зростаючих потреб навчальних закладів, а також впровадження новітніх технологій та стандартів безпеки в цифровий освітній простір.

Вибір між монолітом та мікросервісною архітектурою в кінцевому рахунку повинен базуватись на особливостях проекту, завданнях та очікуваних навантаженнях. У контексті сучасної цифрової трансформації освітніх закладів, де швидкість впровадження інновацій та гнучкість системи є критично важливими, мікросервісна архітектура стає пріоритетом, забезпечуючи

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				18
Змн.	Арк.	№ докум.	Підпис	Дата		

стабільну, продуктивну та адаптивну інфраструктуру для управління освітніми процесами.

1.4 Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення

Надалі було необхідно визначитися із інструментальними засобами для розробки майбутнього ПЗ. Одним із варіантів було використати мову PHP [7], яка успішно зарекомендувала себе у світі веб-розробки і є чудовим та зваженим рішенням – величезна кількість навчальних матеріалів, ком'юніті, а також існуючі фреймворки доводять її доцільність для нашого випадку. У якості основного фреймворку для бекенду обрано Yii2 [5], що забезпечує гнучкість у розробці та легкість інтеграції з іншими компонентами системи. У якості системи управління базами даних використовується MySQL [15], що гарантує високу продуктивність та надійність. Окремо буде розроблено мікросервіс для імпорту даних з ЄДЕБО на платформі Node.js [8], що дозволить забезпечити швидку обробку та інтеграцію даних.

Вимоги до сервера (Мікросервіс Node.js):

- Процесор:
 - не менше 4 ядер;
 - тактова частота не менше 3.00 GHz;
 - підтримка багатопоточності;
 - архітектура x86_64;
- Оперативна пам'ять (RAM):
 - не менше 32 GB;
 - бажано — підтримка ECC (для підвищення надійності в умовах інтенсивного логування та передачі даних);
- Дискова підсистема:
 - SSD накопичувач обсягом не менше 100 GB;
 - зарезервований простір для логів, тимчасових файлів і черг повідомлень (RabbitMQ / Redis);

		Цалко Я.В.			ПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				19
Змн.	Арк.	№ докум.	Підпис	Дата		

- Операційна система:
 - LINUX (debian-based або Alpine Linux);
 - підтримка systemd або supervisord для автозапуску процесів;
 - можливість запуску у Docker-контейнерах або на віртуальній машині (KVM);
- Мережа:
 - гарантована пропускна здатність не менше 100 Mbps;
 - стабільна внутрішня комунікація з основним сервером (ping < 1 ms);
 - відкриті порти для REST API та WebSocket (80, 443, 3000+);
- Інше:
 - резервне копіювання конфігурацій та логів щонайменше раз на добу;
 - підтримка автоматичного розгортання через CI/CD (GitLab CI, GitHub Actions або подібні).

Висновки до першого розділу

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

Проведене дослідження аналогічних програмних рішень дало змогу глибше зрозуміти поточний стан ринку, оцінити наявні рішення та виділити їхні сильні й слабкі сторони. Це стало основою для обґрунтування шляхів удосконалення, які дозволять запропонованому продукту бути конкурентоспроможним та відповідати сучасним вимогам користувачів. У

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				20
Змн.	Арк.	№ докум.	Підпис	Дата		

результаті аналізу було сформульовано рекомендації щодо покращення функціональних можливостей, підвищення ефективності роботи системи та забезпечення її адаптивності до майбутніх змін.

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

Таким чином, виконаний аналіз та прийняті рішення формують надійну концептуальну базу для наступних етапів розробки. Визначені напрямки вдосконалення, обрані інструменти та розроблена загальна архітектура системи забезпечують ефективний старт для подальшої реалізації програмного продукту, сприяючи його відповідності сучасним технологічним вимогам та очікуванням користувачів.

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		21

РОЗДІЛ 2. ПРОЕКТУВАННЯ СИСТЕМИ ОНЛАЙН-МЕНЕДЖМЕНТУ ОСВІТНІХ ПРОЦЕСІВ

2.1 Визначення варіантів використання та структури системи

Для забезпечення стабільної, надійної та ефективної роботи модуля «Здобувачі освіти» сформульовано перелік високорівневих вимог, що охоплюють функціональні та нефункціональні аспекти. Особливу увагу приділено автоматизації рутинних дій, зручності доступу до інформації, взаємодії з ЄДЕБО, а також збереженню цілісності та актуальності даних здобувачів освіти.

Бізнес-вимоги

- 1. Основна мета:** Забезпечення централізованого обліку всіх здобувачів освіти у закладі, ведення їх особових справ, контроль стипендіального забезпечення, узгодження даних з ЄДЕБО, а також автоматизація процесів, пов'язаних із переведенням, випуском та відрахуванням студентів.
- 2. Реалізація проєкту:** Модуль буде реалізовано як частина інформаційної системи в архітектурі «клієнт-сервер». Серверна частина розробляється на базі фреймворку Yii2, що дозволяє створити розширювану, масштабовану та безпечну систему, інтегровану з центральною базою даних та зовнішніми сервісами (зокрема ЄДЕБО) [28].

Вимоги адміністрації

У межах функціоналу, необхідного адміністрації закладу освіти, передбачено повноцінний облік усіх здобувачів із можливістю здійснювати фільтрацію, пошук та експорт списків у зручному форматі. Система також має підтримувати автоматичне формування обхідних листів для студентів, які завершують навчання або підлягають відрахуванню, що дозволяє значно пришвидшити внутрішні адміністративні процедури.

Передбачено зберігання фотографій здобувачів із прив'язкою до їхніх особових справ, що спрощує ідентифікацію осіб і підвищує точність обліку.

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		22

Управління академічними групами реалізується через зручний механізм створення, редагування, переведення студентів між групами, з урахуванням поточного статусу кожного здобувача.

Інформаційна система повинна бути синхронізована з Єдиною державною електронною базою з питань освіти (ЄДЕБО), забезпечуючи актуальність групових та індивідуальних даних. Окрему увагу приділено можливості формування актуальних списків академічних груп та їх подальшому експорту.

Крім цього, реалізується інструмент для призначення навчальних курсів та фіксації статусів студентів — активне навчання, відрахування, академічна відпустка тощо. Система повинна своєчасно виявляти невідповідності в даних, наприклад, коли студент не має призначеного курсу або перебуває в групі з неактуальним статусом, що дозволяє адміністрації оперативно реагувати на такі ситуації.

Вимоги власників

Власники системи висувують чіткі вимоги щодо контролю доступу та безпеки. Основним принципом є розмежування прав користувачів відповідно до їхніх функціональних обов'язків, що дозволяє уникнути несанкціонованого втручання в критичні розділи системи. Кожна роль має доступ лише до тих функцій, які відповідають її компетенції в реальних робочих процесах.

Для забезпечення прозорості та відповідальності передбачено ведення повного журналу дій усіх користувачів, особливо у випадках, коли здійснюються зміни в особових даних здобувачів чи їх груповій приналежності. Такий підхід дозволяє виявляти й аналізувати всі ключові дії в системі та при необхідності відновлювати їхню хронологію.

Окремо наголошується на можливості адаптації системи під конкретні потреби університету — з гнучкими параметрами налаштування, що дозволяють враховувати організаційні особливості кожного закладу освіти.

Характеристика об'єкта комп'ютеризації

Користувачами модуля «Здобувачі освіти» є представники деканатів, навчальних відділів, бухгалтерії, а також технічні адміністратори. Для

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		23

здобувачів освіти реалізується обмежений функціонал для перегляду власних даних. Усі дії здійснюються через веб-інтерфейс, адаптований для роботи як на десктопах, так і на мобільних пристроях. Головний акцент — на зручність, швидкий пошук та мінімізацію помилок під час обробки великої кількості даних.

Функціональні вимоги

Функціональність системи має відповідати потребам різних категорій користувачів, забезпечуючи гнучке розмежування прав доступу на основі визначених ролей — адміністратора, представників деканату, бухгалтерії та студентів. Після авторизації кожен користувач отримує доступ лише до тих модулів, які відповідають його повноваженням.

Система повинна дозволяти перегляд, редагування та гнучке фільтрування інформації про здобувачів освіти. Передбачено також імпорт даних із ЄДЕБО з можливістю їх подальшої актуалізації, зокрема в частині групової належності, курсів і статусів навчання [30].

Автоматизовано процес формування обхідних листів, із можливістю фіксації проходження кожного підрозділу. Для кожного здобувача можна завантажити фотографію, здійснювати її перегляд або заміну.

Система надає інструменти для управління академічними групами, формування списків і виявлення помилок або невідповідностей у даних, що потребують корекції. Крім того, передбачено можливість створення різноманітних звітів у популярних форматах — .pdf, .xlsx, .docx, з розподілом за групами, курсами, стипендіями тощо.

Для своєчасного інформування користувачів система має підтримувати автоматичну розсилку сповіщень щодо змін у статусі, необхідності оновлення або підтвердження персональних даних.

Нефункціональні вимоги

1. Сприйняття

- Час навчання персоналу для роботи з модулем — до 5 годин;
- Час відгуку інтерфейсу — до 2 секунд;

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		24

- Короткі пояснення та підказки до ключових функцій у вигляді підказок при наведенні мишкою.

2. Надійність

- Безперервна робота системи — не менше 95% часу;
- Середній період безвідмовної роботи — понад 20 робочих днів;
- Повна відповідність синхронізованих даних з ЄДЕБО.

3. Продуктивність

- Підтримка великої кількості записів без втрати швидкодії;
- Оптимізація запитів при масовій обробці або фільтрації даних;
- Адаптивний інтерфейс для будь-яких пристроїв.

4. Можливість експлуатації

- Модульна архітектура з можливістю розширення функцій;
- Інтеграція з іншими модулями (сесія, розклад, контингент тощо);
- Систематичне резервне копіювання всіх даних.

Таким чином, було визначено ключових акторів системи, їх вимоги до функціоналу майбутнього програмного забезпечення, а також визначено низку нефункціональних вимог, що покращать взаємодію користувачів із додатком, і як наслідок було спроектовано діаграму акторів для демонстрації наочних потреб кожного з них (рис. 2.1):

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		25



Рисунок 2.1 - Діаграма акторів системи

Модуль «Здобувачі освіти» є одним із ключових елементів інформаційної системи, орієнтованим на забезпечення повного управління академічними та особовими даними студентів. Основним користувачем модуля виступає адміністрація університету – працівники деканатів, навчальних відділів, бухгалтерії, яким надано повноваження відповідно до їхньої ролі в освітньому процесі.

Адміністрація має змогу вести повний облік здобувачів освіти, включаючи гнучке сортування, фільтрацію, пошук та експорт списків. У процесі завершення навчання або відрахування здобувача система автоматично формує обхідний лист, що істотно спрощує документообіг. Також передбачено можливість підтвердження фотографій до особових справ студентів, що використовується в процесі формування студентських квитків.

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		26

Окрему роль у модулі відіграє адміністрування академічних груп – створення, редагування, переведення здобувачів, формування списків груп та їх експорт. Для підтримки актуальності даних реалізовано синхронізацію з ЄДЕБО, яка дозволяє автоматично оновлювати особові та навчальні дані. Також передбачено призначення курсів і статусів навчання (наприклад, активний, академвідпустка, відрахований), а система виявляє невідповідності у курсах та групах – наприклад, якщо студент не має призначеного курсу або має неактуальний статус.

Другим актором є власник системи — команда або структурна одиниця, відповідальна за налаштування та технічну підтримку системи. Власник має повноваження щодо розмежування доступів користувачів відповідно до їх посадових обов’язків, здійснює логування всіх дій у системі з метою забезпечення прозорості та безпеки. Крім того, він відповідає за загальні налаштування системи під потреби конкретного університету: від структури навчальних статусів до організації інтерфейсу та політик доступу.

2.2 Обґрунтування вибору DevOps-інфраструктури

У процесі розробки й підтримки інформаційної системи Digital University було впроваджено комплексне DevOps-рішення, орієнтоване на забезпечення безперервної інтеграції, тестування, доставки та розгортання програмного забезпечення. Враховуючи використання каскадної (Waterfall) моделі розробки, а також необхідність підтримки різноманітного середовища клієнтських машин, особливу увагу приділено інфраструктурним аспектам, автоматизації та забезпеченню надійності розгортання. Нижче подано характеристику ключових технологій, що формують DevOps-екосистему проєкту.

Одним із центральних компонентів CI/CD-процесів став Jenkins [16] — сервер з відкритим кодом для організації безперервної інтеграції (CI) та доставки (CD). Його вибір зумовлений широким визнанням у професійному середовищі, стабільністю роботи, високим рівнем кастомізації, а також невисокими системними вимогами, що дозволяє розгорнути його навіть на обмежених хост-

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		27

середовищах. Jenkins забезпечив гнучке управління пайплайнами, інтеграцію з системами контролю версій, статичним аналізом коду та засобами розгортання, що є критично важливим для підтримки життєвого циклу розробки.

Для збереження, версіювання та координації командної роботи над кодовою базою використовується GitHub як віддалений Git-репозиторій [17]. Репозиторій виконує роль централізованого сховища програмного забезпечення, водночас підтримуючи поділ на середовища (development та production), політику гілок, а також інтеграцію з Jenkins для запуску CI/CD сценаріїв. Гілка dev відповідає за розробницьке середовище, тоді як main — за стабільну продуктивну версію, що забезпечує чітке розділення відповідальності в процесі релізів.

З метою забезпечення якості коду та зменшення кількості помилок на ранніх етапах впроваджено SonarQube [18] – сервер для статичного аналізу коду. Цей інструмент дозволяє автоматично ідентифікувати вразливості, технічний борг, проблеми стилю та інші недоліки в коді, що, у свою чергу, сприяє підвищенню надійності та підтримуваності програмного забезпечення. Інтеграція SonarQube із Jenkins дала змогу включити етапи перевірки якості безпосередньо в CI-процес.

Суттєву роль у побудові середовищ відіграє Docker [19] – система контейнеризації, що забезпечує ізольоване середовище для запуску застосунків, незалежно від конфігурації хост-систем. Контейнеризація сприяє забезпеченню ідентичності середовищ на всіх етапах — від розробки до продуктивного використання. Використання Docker Compose як додаткового інструмента оркестрування дозволило визначити й керувати багатоконтейнерними застосунками, спрощуючи конфігурацію сервісів, залежностей та мережових зв'язків між компонентами.

Для забезпечення автоматичного оновлення контейнерів на основі змін в реєстрі Docker-образів було інтегровано Watchtower [26] – сервіс, що відстежує оновлення у Docker Registry та самостійно завантажує нові версії контейнерів без

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				28
Змн.	Арк.	№ докум.	Підпис	Дата		

переривання сервісу. Це рішення дозволяє підтримувати актуальність програмного забезпечення без ручного втручання.

Docker Registry відіграє роль внутрішнього сховища для контейнерів, що використовуються в проєкті. Завдяки цьому забезпечується централізоване зберігання та контроль версій Docker-образів, а також інтеграція з пайплайнами Jenkins для автоматизованого розгортання.

З метою посилення безпеки інфраструктури було впроваджено Trivy [27] – засіб з відкритим кодом для сканування вразливостей у Docker-образах. Це дозволило системно перевіряти компоненти застосунку ще до їх розгортання в продуктивному середовищі, зменшуючи ризик експлуатації відомих вразливостей.

Для компіляції та збирання програмного забезпечення в готовий до дистрибуції пакет використовуються необхідні пакети PHP [6] та Node.js [9], які інсталюються у CI-середовище Jenkins або відповідні контейнери, в залежності від типу сервісу.

Інфраструктура розділена на два середовища: Dev (Development) та Prod (Production). Розробницьке середовище використовується для тестування нових функцій та інтеграцій, тоді як продуктивне середовище — це фінальна платформа, де працює стабільна версія системи для кінцевих користувачів. Такий підхід дозволяє уникнути ризиків, пов’язаних із негайним впровадженням змін, забезпечуючи контрольоване тестування й впровадження функціоналу.

Загалом, побудована DevOps-архітектура демонструє високий ступінь автоматизації, гнучкості та масштабованості. Вона дозволяє не лише пришвидшити розгортання нових версій, а й забезпечити стабільність, контроль якості та безпеку програмного забезпечення, що є критично важливим для довготривалого підтримання навчальної цифрової платформи.

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		29

2.3 Розробка бази даних системи

Під час розробки бази даних було використано ключові підходи до проектування у реляційних БД, що зрештою дозволило побудувати наступну структуру:

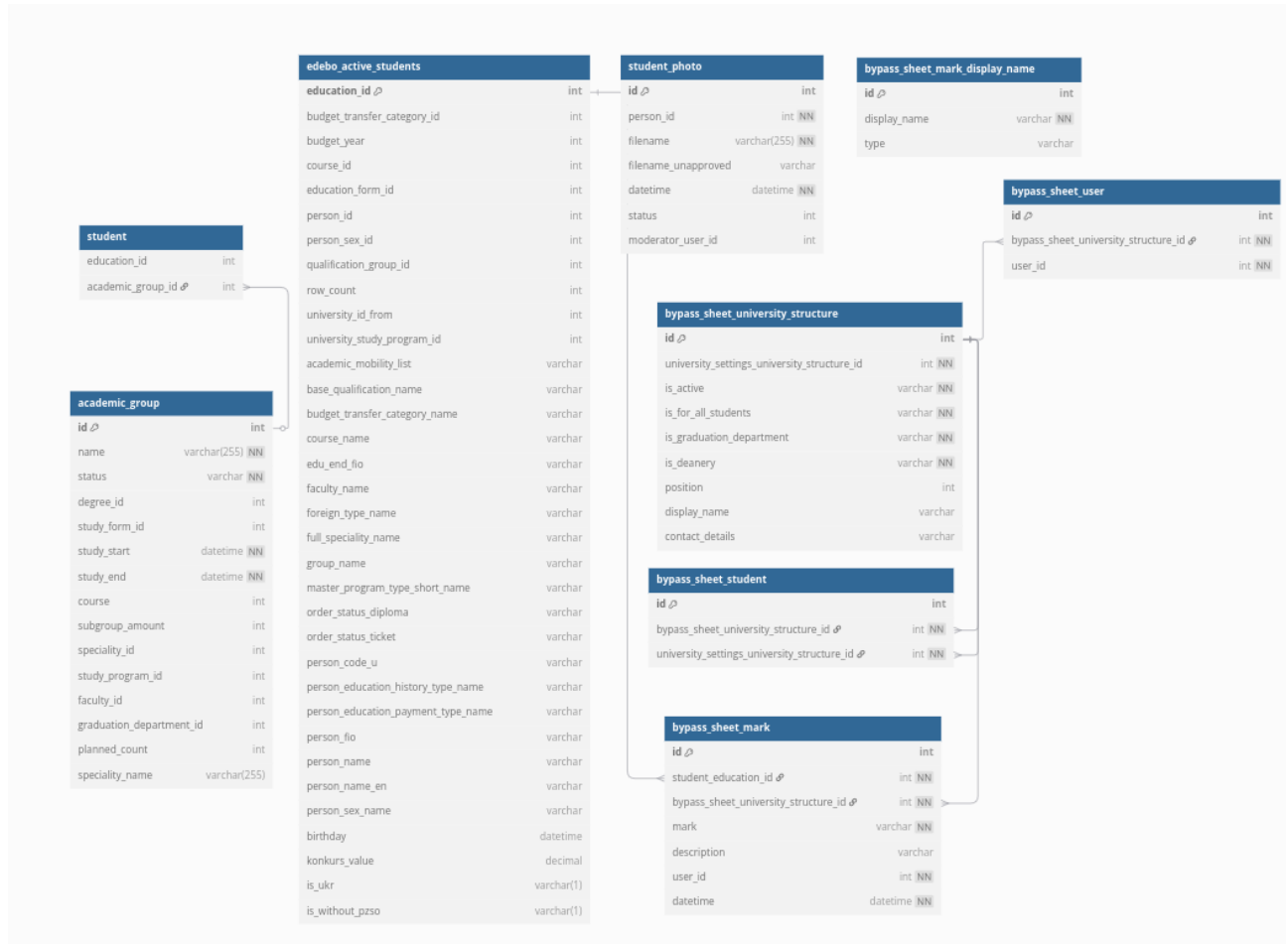


Рисунок 2.2 – Спроектowana база даних

У модулі «Здобувачі освіти» реалізовано функціональність, пов'язану з управлінням інформацією про студентів, їх навчальні групи, освітні програми та внутрішньоуніверситетські процеси, зокрема заповнення обхідного листа при завершенні навчання. База даних модуля структурована у відповідності до вимог логічного поділу функцій та забезпечення цілісності даних, а також враховує взаємозв'язок із інформацією, отриманою з Єдиної державної електронної бази з питань освіти (ЄДЕБО).

Центральними сутностями є таблиці edebo_active_students, яка містить узагальнені відомості про студентів і є імпортованою з ЄДЕБО, а також academic_group та student, що формують зв'язок між локальними навчальними

групами та конкретними здобувачами освіти. Кожен студент визначається за унікальним ідентифікатором освіти (education_id), що дозволяє інтеграцію зовнішніх джерел даних з внутрішньою структурою системи. Додаткова інформація, така як фотографії студентів (student_photo), також включена до модуля і використовується в процесах створення студентських квитків - для затвердження фотографії її необхідно надіслати студенту на портал через особистий кабінет, де в подальшому вона буде розглянута і схвалена або відхилена комісією.

Особливу роль у модулі відіграє підсистема обхідних листів — процесу, який реалізується наприкінці навчання студента. Вона охоплює підрозділи, що мають підтвердити відсутність заборгованостей або наявність усіх необхідних підписів. Для цього створено таблиці bypass_sheet_university_structure (перелік підрозділів), bypass_sheet_user (відповідальні працівники), bypass_sheet_student (маршрути обходу для кожного здобувача освіти) та bypass_sheet_mark (фіксація результатів обходу). Всі ці сутності побудовані з урахуванням можливості масштабування та гнучкої кастомізації назв відміток — зокрема, через таблицю bypass_sheet_mark_display_name.

Загальна модель модуля «Здобувачі освіти» відображає логічну взаємодію між організаційною структурою університету та освітнім процесом, забезпечуючи як зручний інтерфейс для працівників, так і повноцінне ведення обліку кожного здобувача освіти на різних етапах його навчального шляху. Інформаційна система таким чином виконує функції не лише бази даних, але й управлінського інструменту для автоматизації внутрішніх процедур університету.

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		31

2.4 Реалізація функціоналу системи управління навчальним процесом

На момент написання роботи процес безперервної інтеграції (CI) був реалізований у спрощеному вигляді через обмежену кількість тестувальників на проєкті. Пайплайн обмежується побудовою Docker-образу, його публікацією у внутрішній реєстр та базовою перевіркою безпеки [22]. Для реалізації використано Jenkins — Java-орієнтований CI-сервер, а сам сценарій автоматизації описаний мовою Groovy у вигляді декларативного Jenkins Pipeline. Він передбачає можливість окремо збирати фронтенд-застосунок, якщо цього вимагає користувач, і містить необхідні змінні середовища для керування параметрами збірки та публікації образів. Такий підхід дозволяє частково автоматизувати публікацію актуальних версій застосунку, зберігаючи гнучкість конфігурації.

```
pipeline {
  agent any
  parameters{
    booleanParam(name: 'BUILD_FRONTEND', defaultValue: false,
description: 'Trigger build for frontend')
  }
  environment {
    NODE_OPTIONS="--max-old-space-size=512"
    DOCKER_REGISTRY = 'registry.ztu.edu.ua'
    DOCKER_IMAGE = 'digital-university/prod'
    DOCKER_IMAGE_FRONTEND = 'digital-university-nginx/prod'
    DOCKER_TAG = 'latest'
    DOCKER_CREDENTIALS = 'docker-login'
  } // environment
```

У пайплайні передбачено окремий етап для збірки фронтенд-частини застосунку у Docker-образ, який виконується лише за умови, якщо під час запуску збірки було ввімкнено параметр BUILD_FRONTEND. Така умова дозволяє уникати непотрібної збірки фронтенду при кожному запуску CI, що особливо корисно під час роботи лише з бекендом [29]:

```
stage('Build Frontend Docker Image') {
  when {
    expression { params.BUILD_FRONTEND }
  } // when
  steps {
    script {
```

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		32

```

        docker.withRegistry("https://${DOCKER_REGISTRY}/"
, "${DOCKER_CREDENTIALS}") {
            dockerImageFrontend =
docker.build("${DOCKER_REGISTRY}/${DOCKER_IMAGE_FRONTEND}:${DOCKER_TA
G}", "-f Dockerfile.frontend .")
        } // docker
    } // script
} // steps
} // stage

```

Усередині етапу відбувається автентифікація у Docker-реєстрі за допомогою збережених облікових даних Jenkins, після чого відбувається створення образу на основі окремого Dockerfile для фронтенду. Сформований образ тегується як latest та записується у змінну, що забезпечує подальшу можливість його публікації. Така реалізація підтримує модульність збірки і забезпечує гнучке керування CI-процесом.

```

stage('Vulnerability Scan') {
    steps {
        script {
            sh """
                trivy image --exit-code 1 --severity
HIGH,CRITICAL ${DOCKER_REGISTRY}/${DOCKER_IMAGE}:${DOCKER_TAG} ||
true
            """
        } // script
    } // steps
} // stage

```

Етап «Vulnerability Scan» відповідає за перевірку backend Docker-образу на наявність потенційних вразливостей безпеки [24]. Для цього використовується інструмент Trivy, який проводить детальний аналіз образу, звертаючи увагу лише на вразливості високого (HIGH) та критичного (CRITICAL) рівня. Виконання команди з параметром --exit-code 1 означає, що у разі виявлення таких проблем Trivy поверне код помилки, а додаткова конструкція || true запобігає зупинці самого пайплайну, дозволяючи продовжити подальші кроки.

```

stage('Push Docker Images') {
    steps {
        script {
            docker.withRegistry("https://${DOCKER_REGISTRY}/"
, "${DOCKER_CREDENTIALS}") {
                dockerImage.push()
                dockerImage.push("${params.DOCKER_TAG_VERSION
}")
            }

            if (params.BUILD_FRONTEND) {

```

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		33

```

                                dockerImageFrontend.push()
                                }
                                } // docker
                                } // script
                                } // steps
                                } // stage
                                } // stages

```

Етап Push Docker Images відповідає за завантаження готових Docker-образів у віддалений реєстр. Тут відбувається авторизація в Docker Registry через Jenkins credentials, після чого бекенд-образ пушиться з тегом latest та додатковим тегом із параметра DOCKER_TAG_VERSION. Якщо увімкнена збірка фронтенду, відповідний образ також відправляється в реєстр. Таким чином, цей крок забезпечує оновлення образів у сховищі для подальшого розгортання.

Конфігурація Docker compose була виконана наступним чином:

```

services:
  website:
    container_name: digital-${UNI_NAME}-website
    image: dregistry.ztu.edu.ua/digital-university/dev:latest
    depends_on:
      - mysql
    volumes:
      -
        ${WEB_SOCKET_CONFIG_PATH}:/var/www/diguni/backend/web/scripts/config/
        config.js
        - ${PHP_CONFIG_PATH}:/var/www/diguni/common/config/main-
        local.php
        - files:/var/www/diguni/files
        - edebo_files:/var/www/Files
        - backend_web:/var/www/diguni/backend/web
        - frontend_web:/var/www/diguni/frontend/web
    ports:
      - 9000:9000
    networks:
      - digital-university-private-network
      - digital-university-public-network
    labels:
      - "com.centurylinklabs.watchtower.enable=true"

```

Сервіс website відповідає за роботу вебінтерфейсу цифрового університету та розгортається у вигляді Docker-контейнера, що базується на актуальному девелоперському образі. Його запуск залежить від готовності бази даних, адже для повноцінної роботи застосунок повинен мати доступ до сховища даних. У контейнер монтуються конфігураційні файли та директорії з файлами, які забезпечують гнучке налаштування та збереження важливої інформації,

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		34

включаючи файли користувачів, взаємодію з ЄДЕБО та ресурси інтерфейсу. Вебзастосунок доступний через порт 9000, що дозволяє користувачам працювати з системою через браузер. Контейнер підключено до двох мереж — для внутрішньої комунікації та зовнішнього доступу. Крім того, реалізовано підтримку автоматичного оновлення через Watchtower, що дозволяє застосунку самостійно оновлюватися при виході нової версії, без потреби ручного втручання адміністратора.

```
rabbitmq:
  container_name: digital-${UNI_NAME}-rabbitmq
  image: rabbitmq:3
  volumes:
    - rabbitmq_data:/var/lib/rabbitmq
  environment:
    RABBITMQ_DEFAULT_USER: ${RABBITMQ_DEFAULT_USER}
    RABBITMQ_DEFAULT_PASS: ${RABBITMQ_DEFAULT_PASS}
  networks:
    - digital-university-private-network
```

Сервіс rabbitmq відповідає за обробку повідомлень у системі та запускається як контейнер на основі офіційного образу RabbitMQ. Для збереження стану черг після перезапуску використовується окремий том. Параметри доступу задаються через змінні оточення, що дозволяє гнучко керувати обліковими даними. Контейнер підключено лише до внутрішньої мережі, завдяки чому RabbitMQ залишається доступним виключно для інших компонентів системи, зокрема бекенду, і не відкривається для зовнішніх з'єднань.

```
edebo:
  container_name: digital-${UNI_NAME}-edebo-microservice
  image: registry.ztu.edu.ua/digital-university-node-
service/dev:latest
  depends_on:
    - mysql
    - mongodb
    - rabbitmq
  volumes:
    - edebo_files:/Files
  env_file:
    - configs/diguni-edebo/.env
  networks:
    - digital-university-private-network
  labels:
    - "com.centurylinklabs.watchtower.enable=true"
```

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		35

Сервіс edebo є мікросервісом, який відповідає за взаємодію з базою ЄДЕБО. Він запускається в окремому контейнері на основі Node.js-образу та працює в оточенні приватної мережі системи. Для повноцінної роботи сервіс залежить від баз даних MySQL і MongoDB, а також брокера повідомлень RabbitMQ. Всі потрібні змінні оточення підтягуються з окремого .env-файлу, а збереження робочих файлів здійснюється через спеціальний том. Сервіс підтримує автоматичні оновлення за допомогою Watchtower, що спрощує його обслуговування та підтримку.

```
mysql:
  container_name: digital-${UNI_NAME}-mysql-server
  image: diguni-basic-mysql-image:latest
  volumes:
    - mysql_data:/var/lib/mysql
  restart: always
  environment:
    MYSQL_DATABASE: ${MYSQL_DATABASE}
    MYSQL_ROOT_PASSWORD: ${MYSQL_ROOT_PASSWORD}
  ports:
    - ${MYSQL_PORT_FORWARD}:3306
  networks:
    - digital-university-private-network
```

Сервіс бази даних MySQL відповідає за зберігання усіх структурованих даних інформаційної системи. Контейнер створюється з кастомного образу, спеціально підготовленого для проєкту, і автоматично перезапускається у разі помилок або зупинки. Дані зберігаються в окремому томі, що забезпечує їхню стійкість до втрати після перезапуску. Налаштування сервісу здійснюється через змінні середовища, які задають ім'я бази даних та пароль адміністратора. Для зручності розробки або інтеграції з іншими інструментами порт бази даних відкривається назовні через змінну перенаправлення. Сервіс функціонує виключно у внутрішній мережі, що підвищує безпеку доступу до даних.

```
mongodb:
  container_name: digital-${UNI_NAME}-mongodb-server
  image: mongo:5.0
  restart: unless-stopped
  ports:
    - "27017:27017"
  environment:
    MONGO_INITDB_ROOT_USERNAME: ${MONGODB_ROOT_USER}
    MONGO_INITDB_ROOT_PASSWORD: ${MONGODB_ROOT_PASS}
```

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		36

```
volumes:
  - mongo_data:/data/db
networks:
  - digital-university-private-network
```

Сервіс MongoDB забезпечує зберігання неструктурованих даних, які використовуються в системі для обробки певних типів інформації, наприклад, логів або крос-сервісних повідомлень. Контейнер створюється на основі офіційного образу MongoDB версії 5.0 і автоматично перезапускається. Для первинного налаштування використовуються змінні середовища, що задають root-користувача та пароль. Порт бази 27017 відкритий для доступу ззовні, що дозволяє іншим сервісам або адміністраторам зчитувати або змінювати дані за потреби. Робота MongoDB ізольована у внутрішній мережі системи.

```
nginx:
  container_name: digital-${UNI_NAME}-nginx
  image: nginx:1.27.2
  volumes:
    - ${NGINX_CONFIGS_FOLDER_PATH}:/etc/nginx/conf.d/
    - ${PORTAL_SSL_FOLDER_PATH}:/etc/ssl/live/portal.host/
    - ${CABINET_SSL_FOLDER_PATH}:/etc/ssl/live/cabinet.host/
    - files:/var/www/diguni/files
    - edebo_files:/var/www/Files
    - backend_web:/var/www/diguni/backend/web
    - frontend_web:/var/www/diguni/frontend/web
  ports:
    - ${NGINX_HTTP_PORT_FORWARD}:80
    - ${NGINX_HTTPS_PORT_FORWARD}:443
  networks:
    - digital-university-public-network
```

Контейнер Nginx виконує роль веб-сервера та зворотного проксі, що забезпечує доступ до інтерфейсів системи “Digital University” через HTTP та HTTPS. Його конфігурація надходить із зовнішньої директорії, що дозволяє легко оновлювати правила маршрутизації без перевипуску образу. Для захищеного з’єднання використовуються SSL-сертифікати для двох окремих доменів — порталу та особистого кабінету. Крім того, до контейнера монтуються спільні ресурси системи, зокрема файли користувачів, бекенд і фронтенд, що дозволяє централізовано обслуговувати весь вебконтент. Завдяки публічній мережі та відкритим портам 80 і 443 сервер є точкою входу для користувачів системи.

```
watchtower:
```

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		37

```

image: containrrr/watchtower
container_name: watchtower
restart: unless-stopped
volumes:
  - /var/run/docker.sock:/var/run/docker.sock
environment:
  WATCHTOWER_CLEANUP: "true"
  WATCHTOWER_POLL_INTERVAL: "300"
  REPO_USER: ${REPO_USER}
  REPO_PASS: ${REPO_PASS}
networks:
  - digital-university-private-network

```

Watchtower у цій системі відповідає за автоматичне оновлення контейнерів до актуальних версій без втручання адміністратора. Контейнер запускається на основі офіційного образу і постійно відслідковує наявність оновлень за допомогою доступу до Docker-сокету. Інтервал перевірки встановлено на п'ять хвилин, а при виявленні нових версій образів, Watchtower виконує оновлення та очищає застарілі. Завдяки змінним середовища передбачена можливість підключення до приватного реєстру з автентифікацією, що забезпечує підтримку закритих середовищ. Працює виключно в межах внутрішньої мережі, залишаючись недоступним ззовні.

```

networks:
  digital-university-private-network:
  digital-university-public-network:
volumes:
  files:
  edebo_files:
  backend_web:
  frontend_web:
  mysql_data:
  rabbitmq_data:
  mongo_data:

```

Даний код визначає дві мережі: digital-university-private-network та digital-university-public-network, які використовуються для ізоляції і організації зв'язку між контейнерами. Також описані томи для збереження даних і спільного використання файлів між контейнерами: files, edebo_files, backend_web, frontend_web, mysql_data, rabbitmq_data та mongo_data. Томам не призначені конкретні налаштування, вони створюються як стандартні Docker томи для збереження персистентних даних і ресурсів.

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		38

За допомогою проведеної DevOps-інфраструктури забезпечується робота усієї системи онлайн-менеджменту освітніх процесів, зокрема і модуля «Здобувачі освіти». Процес роботи із модулем розпочинається з імпортування груп із таблиць ЄДЕБО до таблиць локальних даних. У державній базі відсутнє поняття єдиної колекції груп, однак інформація про них зустрічається у інформації про студентів у вигляді назви, через що ми можемо знайти усіх студентів заданої групи:

```
$groupName = Yii::$app->request->post('groupName');
$academicGroup = new AcademicGroup();
$edeboAcademicGroup = ActiveStudent::find()
    ->where(['group_name' => $groupName])
    ->orderBy(['education_date_begin' => SORT_ASC])
    ->one();
```

Після успішного знаходження групи ми заповнюємо новостворену модель частиною даних з ActiveStudent:

```
$academicGroup->name = $groupName;
$academicGroup->status = 1;
$academicGroup->degree_id = $edeboAcademicGroup-
>qualification_group_id;
$academicGroup->study_form_id = $edeboAcademicGroup-
>education_form_id;
$academicGroup->study_start = $edeboAcademicGroup-
>education_date_begin;
$academicGroup->study_end = $edeboAcademicGroup->education_date_end;
$academicGroup->course = $edeboAcademicGroup->course_id;
```

Якщо ми маємо назву освітньої програми, ми здійснюємо її пошук у відповідній таблиці EdeboUniversityStudyPrograms:

```
if ($edeboAcademicGroup->study_program_name) {
    $program = EdeboUniversityStudyPrograms::findOne([
        "speciality_name" => $edeboAcademicGroup-
>full_speciality_name,
        "qualification_group_id" => $edeboAcademicGroup-
>qualification_group_id,
```

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		39

```

        "study_program_name"      => $sedeboAcademicGroup->study_program_name,
    ];
}

```

Також для повноцінного заповнення запису групи інформацією необхідно визначити факультет, до якого вона належатиме – аналогічно, здійснюється пошук за назвою:

```

$faculty =
EdeboUniversityStructures::findOne(["university_facultet_full_name"
=> $sedeboAcademicGroup->faculty_name]);

```

А далі цей об'єкт буде використаний при записуванні інформації до AcademicGroup:

```

$specialtyId = $program->specialty_id;
$specialtyName = $sedeboAcademicGroup->full_specialty_name;
$studyProgramId = $sedeboAcademicGroup->study_program_name ? $program->university_study_program_id : null;
$facultyId = $faculty->id_university_facultet;

```

Не менш необхідним є додавання студентів до групи - це здійснюється за допомогою прив'язування в проміжній таблиці Student, що містить ідентифікатор студента з ЄДЕБО, а також ключ групи:

```

private function addStudentsToLocalGroup($groupName): void
{
    $students = EdeboActiveStudents::find()->where(['group_name' =>
$groupName])->all();
    $academicGroup = AcademicGroup::findOne(["name" => $groupName]);
    foreach ($students as $student) {
        $newStudent = new Student();
        $newStudent->education_id = $student->education_id;
        $newStudent->academic_group_id = $academicGroup->id;
        $newStudent->save(false);
    }
}

```

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ІЗ	Арк.
		Морозов А.В.				40
Змн.	Арк.	№ докум.	Підпис	Дата		

Запит знаходить усіх людей, що навчаються у заданій групі за її назвою, після чого в циклі виконує їх прив'язку в локальній таблиці.

За необхідності групу можна сформувати вручну, скориставшись роутом створення в контролері `AcademicGroupController`, що також передбачає можливість створення через AJAX-запит:

```
$model = new AcademicGroup();  
$model->status = 1;  
if ($model->loadAll(Yii::$app->request->post())) {  
    if ($model->save(false)) {  
        if ($model->speciality_id) {$model->speciality_name = $this->getSpecialityNameBy($model->speciality_id);  
        $model->save(false);  
    }  
    if (Yii::$app->request->isAjax) {  
        Yii::$app->response->format = Response::FORMAT_JSON;  
        return ['success' => true];  
    }  
}  
}
```

Далі стає можливою взаємодія із імпортованими елементами в систему. Одним із найпростіших видів взаємодії є перегляд складу групи, що надає можливість їх переведення. Для цього необхідно відобразити студентів, що знаходяться поза групами:

```
$groupInfo = $this->findModel($id);  
$students = $this->getStudentsInGroup($groupInfo->id);  
$studentsNoGroup = $this->getStudentsNoGroup($groupInfo);
```

У свою чергу, метод `getStudentsNoGroup()` звертається до сервісу `AcademicGroupOwnerService`, де спочатку здійснюється пошук назви спеціальності за її ідентифікатором:

```
$specialityName =  
EdeboUniversityStudyPrograms::findOne(["speciality_name" =>  
$groupInfo->speciality_name])->speciality_name;
```

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		41

Далі визначається список допустимих освітніх статусів, які будуть враховуватись у вибірці — це перелік статусів на кшталт «навчається», “переведено” тощо.

```
$studyStatusesIds = [1, 10, 11, 12, 13, 14, 15, 17, 20, 21, 25, 30, 50, 61, 64, 70, 80, 81, 90];
```

Допоміжний підметод формує запит, що починає вибірку усіх активних студентів, які наразі навчаються, однак не виконує його, а надає можливість побудувати:

```
$query = AcademicGroupOwnerService::getAllEdeboStudentsQuery([
    "$student.id",
    "$edeboStudents.education_id",
    "$student.academic_group_id",
    //...
]);
```

До нього додається частина запиту, що виконує фільтрацію студентів з метою виявлення тих, хто має пусті рядки або null в комірці його групи:

```
$query->andWhere(['or', ['!=', "$edeboStudents.group_name", new
\yii\db\Expression("$group.name")],
    ["$group.name" => null],
    ["$edeboStudents.group_name" => null],
    ["$edeboStudents.education_id" => null],
]);
```

На завершення додаються додаткові умови, які перевіряють відповідність навчальної програми, форми навчання, спеціальності та дати завершення навчання. Це забезпечує точне співпадіння профілю студента з характеристиками заданої групи. Результатом є відсортований за датою народження список студентів, які можуть бути додані до локальної групи, але ще не прив’язані до неї:

```
->andWhere(["$student.academic_group_id" => null])
->andWhere(["$edeboStudents.history_type_id" => $studyStatusesIds])
```

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		42

```

->andWhere([
    'or',
    ["$edeboStudents.university_study_program_id" => $studyProgramId],
    ["$edeboStudents.university_study_program_id" => null]
])
->andWhere(["$edeboStudents.education_form_id" => $studyFormId])
->andWhere(["$edeboStudents.full_speciality_name" =>
$specialityName])

```

Така логіка пошуку також застосовується на сторінці перегляду осіб, що мають невідповідність курсів між записами всередині системи та інформацією, яка розміщена в ЄДЕБО. Однак використовується інша умова, спрямована на пошук курсу студента:

```

$query
->andWhere("$group.course != $edeboStudents.course_id")
->andWhere(["$edeboStudents.education_id" => $educationIds]);
;

```

Щоб змінити інформацію про студента, необхідно відправити запит на ЄДЕБО, і в разі успішного його виконання здійснити перезапис рядка в базі даних проекту. Зв'язком із зовнішнім мікросервісом, що підтримує зв'язок із державною базою, є EdeboNodeService. Ось як виглядає метод для переведення студента до іншої групи:

```

public function changeGroupForStudent(int $educationId, string|null
$groupName, int $qualificationGroupId): bool
{
    $url = $this->getUrl() . "/info-update/group/student/add";
    $data = [
        'educationId' => $educationId,
        'groupName' => $groupName,
        'qualificationGroupId' => $qualificationGroupId,
    ];
    return $this->sendPost($url, $data);
}

```

Знаючи хост та порт мікросервісу, отримані із конфігураційних файлів, формується URL-адреса для здійснення запиту. Набір даних містить

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		43

ідентифікатор навчання студента, необхідну назву групи, а також його освітній ступінь, так як ці параметри є встановленими як обов'язкові на рівні API ЄДЕБО. POST-запит відправиться на мікросервіс, який перенаправить отримані дані до державної бази.

Аналогічно виконується зміна номеру курсу студента:

```
public function changeStudentCourse(int $educationId, int
$qualificationGroupId, int $course): bool
{
    $url = $this->getUrl() . "/info-update/group/student/set-course";
    $data = [
        'educationId' => $educationId,
        'qualificationGroupId' => $qualificationGroupId,
        'course' => $course
    ];
    return $this->sendPost($url, $data);
}
```

Під кінець кожного навчального року курс групи збільшується на 1, тому було передбачено функціонал для переведення кожного студента на наступний курс:

```
public function incrementGroupCourse(string $groupName, int
$qualificationGroupId): mixed
{
    $url = $this->getUrl() . "/info-update/group/increment-course";
    $data = [
        'groupName' => $groupName,
        'qualificationGroupId' => $qualificationGroupId,
    ];
    return $this->sendPostWithResponse($url, $data);
}
```

Не менш важливим є можливість перегляду усіх студентів, де найголовнішою проблемою було забезпечити оптимізованість рішення. Університет здатен зберігати інформацію про десятки тисяч студентів, які навчались у різні роки, тому це завдання потребувало додаткових функцій, що допомогли відобразити сторінку максимально швидко.

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				44
Змн.	Арк.	№ докум.	Підпис	Дата		

Під час роботи з паспортними даними було надано перевагу віртуальним зв'язкам між таблицями на рівні моделей Yii2, щоб уникнути додаткового завантаження полів [7]:

```
$query->joinWith('passportData', false)->distinct();
```

Натомість, дані з паспортів студентів брались не із JOIN, а окремим запитом із сервісу StudentService, що повертав лише ті записи, які наразі вибрані в пагінації:

```
public static function getPassportsOf(array $studentsIds, $fields =
    "*", $searchType = null): array
{
    if (!$searchType) {
        $searchType = "Паспорт";
    }
    $query = self::getManyPersonDocumentsQuery($studentsIds, $fields)
        ->andWhere(['person_document_type_name' => $searchType]);
    return $query->all();
}
```

Запит виконує вибірку в таблиці усіх документів здобувачів за ключовим словом “Паспорт”, а також накладає обмеження у вигляді ідентифікаторів студентів, які відображатимуться на сторінці.

Аналогічний підхід практикується і в інших даних, і таким чином була застосована логіка додавання JOIN до запиту лише в тому разі, якщо поле цієї таблиці було включено до сортування чи пагінації:

```
if ($this->hasSearchFields($searchFields)) {
    $query->joinWith(['requestTmp'])->distinct(true);
    //...
}
```

Не менш ресурсозатратним є автоматичний запит, що рахує кількість записів під час фільтрації та пагінації - це виконує побудований важкий запит двічі. Щоб уникнути цього, було налаштовано спеціальний запит замість нього, що прибирає умови й обмеження, які впливають на кількість та сортування, що зробило систему більш оптимізованою:

```
$countQuery = clone $query;
```

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				45
Змн.	Арк.	№ докум.	Підпис	Дата		

```
$countQuery->limit(-1)->offset(-1)->orderBy([]);
```

Нові здобувачі освіти повинні завантажити свою фотографію із власного кабінету, після чого вона буде відправлена на розгляд. Адміністрація має можливість підтвердити фотографію, якщо вона зроблена з дотриманням усіх її **ВИМОГ**:

```
$model->status = 1;
$model->filename = $model->filename_unapproved;
$model->filename_unapproved = null;
$model->moderator_user_id = Yii::$app->user->identity->getId();
$model->save(false);
```

Статус фотографії змінюється на “Підтверджено”, а файл із тимчасового поля переміщається в постійне, зберігаючи інформацію про того, хто здійснив цю дію. Аналогічним чином, працює і відхилення фотографії:

```
$model->status = 2;
if($model->filename){
    $model->filename_unapproved = $model->filename;
    $model->filename = null;
    $model->moderator_user_id = Yii::$app->user->identity->getId();
}
```

Система активно застосовується для обробки інформації про здобувачів освіти, і одним із важливих етапів є формування обхідного листа по завершенню навчання. Для цього для кожного студента потрібно сформувати список етапів, що мають бути пройденими для отримання диплому за допомогою методу `getLetterForStudent`:

```
if (static::isDeaneryExist($isActive)) {
    $deanery = self::getStudentDeanery($educationId, $isActive);
    $deaneryMark = self::getDeaneryMark($educationId);
    $data[] = [
        "name" => "Деканат",
        "value" => $deaneryMark->mark ?? null,
        "bypass_structure_id" => $deanery->id ?? null,
        "description" => $deaneryMark->description ?? null,
        'contacts' => $deanery->contact_details ?? null,
    ];
}
```

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				46
Змн.	Арк.	№ докум.	Підпис	Дата		

```
}
```

Спочатку відбувається додавання деканату до списку. Зберігається назва структури, відмітка про проходження, а також опис та контакти підрозділу. Аналогічно, відбувається пошук його випускової кафедри:

```
$department = self::getGraduationDepartment($educationId, $isActive);  
$departmentMark = self::getGraduationDepartmentMark($educationId);
```

Для знаходження випускової кафедри необхідно знайти факультет та кафедру студента за допомогою інформації про його освітню програму, що здійснюється одним SQL-запитом:

```
protected static function findStudentStructureCathedralQuery(int  
$educationId): ActiveQuery  
{  
    $query = BypassSheetUniversityStructure::find()  
        ->innerJoin($universityStructure, "$universityStructure.id =  
$bypassSheetUniversityStructure.university_settings_university_struct  
ure_id")->innerJoin($cathedralStudyProgram,  
"$cathedralStudyProgram.cathedral_id = $universityStructure.id")  
        ->innerJoin($students, "$students.university_study_program_id  
= $cathedralStudyProgram.study_program_id")  
        ->andWhere(["$students.education_id" => $educationId]);  
    return $query;  
}
```

Крім обов'язкових структур, система надає функціонал для можливості додавання власних, що включаються в обхідний лист. Для додавання їх до списку виконується наступний метод, що вибирає всі решту структур і враховує порядок їх відображення у таблиці:

```
public static function getAllStructure(int $educationId, $isActive =  
true): array  
{  
    $bypassSheetUniversityStructure =  
    BypassSheetUniversityStructure::tableName();  
    $data = static::getAllStructureQuery($educationId, $isActive)-  
>orderBy(["$bypassSheetUniversityStructure.position" => SORT_ASC])-  
>asArray()->all();  
    return $data;
```

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				47
Змн.	Арк.	№ докум.	Підпис	Дата		

```
}
```

Всі дані будуть складені в один масив за допомогою `array_merge`:

```
return array_merge($data, self::getAllStructure($educationId,  
$isActive));
```

Подана функція отримання статусів заповненості обхідного листа працює в циклі для кожного студента на поточній сторінці:

```
foreach ($allRecords as $record) {  
    $educationId = $record['education_id'];  
    $l =  
CircularLettersStudentService::getLetterForStudent($educationId);  
    $studentLetters[$educationId] = $l;  
}
```

Для виставлення відмітки надсилається запит на сервер, де виконується наступний код:

```
$sheetMark =  
CircularLettersUniversityStructureService::findBypassSheetMarkOf($edu  
cation_id, $bypass_structure_id);  
if (is_null($sheetMark)) {  
    $sheetMark = new BypassSheetMark();  
    $sheetMark->student_education_id = $education_id;  
    $sheetMark->bypass_sheet_university_structure_id =  
$bypass_structure_id;  
    if(!is_null($description)) {  
        $sheetMark->mark = false;  
    }  
}
```

Створюється запис в таблиці, куди зберігається ідентифікатор студента, підрозділу, а також статус відмітки.

		Цалко Я.В.			ПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		48

Висновки до другого розділу

У процесі розробки інформаційної системи для адміністрування освітніх процесів у закладі вищої освіти були визначені ключові функціональні та нефункціональні вимоги, що охоплюють потреби як адміністрації, так і власників системи. Сформовано повний перелік завдань, серед яких — забезпечення обліку здобувачів освіти, управління академічними групами, синхронізація з ЄДЕБО, а також реалізація багаторівневого контролю доступу й журналювання дій користувачів.

Архітектура системи побудована на принципах модульності та масштабованості, що дозволяє ефективно підтримувати велику кількість одночасних користувачів без зниження продуктивності. Упроваджено адаптивний інтерфейс, зручний для використання як на настільних пристроях, так і на мобільних екранах. Реалізовано механізми фільтрації, сортування та експорту великих обсягів даних, що значно спрощує роботу адміністративного персоналу.

На етапі проєктування було передбачено можливість гнучкого налаштування системи відповідно до індивідуальних потреб закладу, а також забезпечено підтримку API для інтеграції з освітніми реєстрами. Особлива увага приділена безпеці: система підтримує розмежування прав доступу, контроль за змінами критичних даних і регулярне резервне копіювання. Наявність DevOps-інфраструктури покращить можливість розгортання системи і сприятиме масштабованості, а також дозволить швидше та ефективніше додавати новий функціонал на існуючих додатках.

У результаті реалізовано повнофункціональний інструмент, який не лише автоматизує рутинні адміністративні процеси, а й забезпечує актуальність, достовірність і захищеність навчальних даних. Таким чином, поставлену мету досягнуто — створено інформаційну систему, що ефективно підтримує управлінські процеси в освітньому середовищі й відповідає сучасним вимогам до університетських IT-рішень.

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		49

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

3.1 Порядок встановлення та налаштування параметрів додатку

Перед впровадженням програмної системи на реальних серверних і клієнтських машинах було розроблено діаграму розгортання [27], яка наочно відображає взаємозв'язки між ключовими компонентами інформаційної системи (рис. 3.1). Ця діаграма слугує інструментом для візуалізації фізичної структури системи, включаючи сервери, клієнтські пристрої, програмні модулі та мережеві взаємодії між ними:

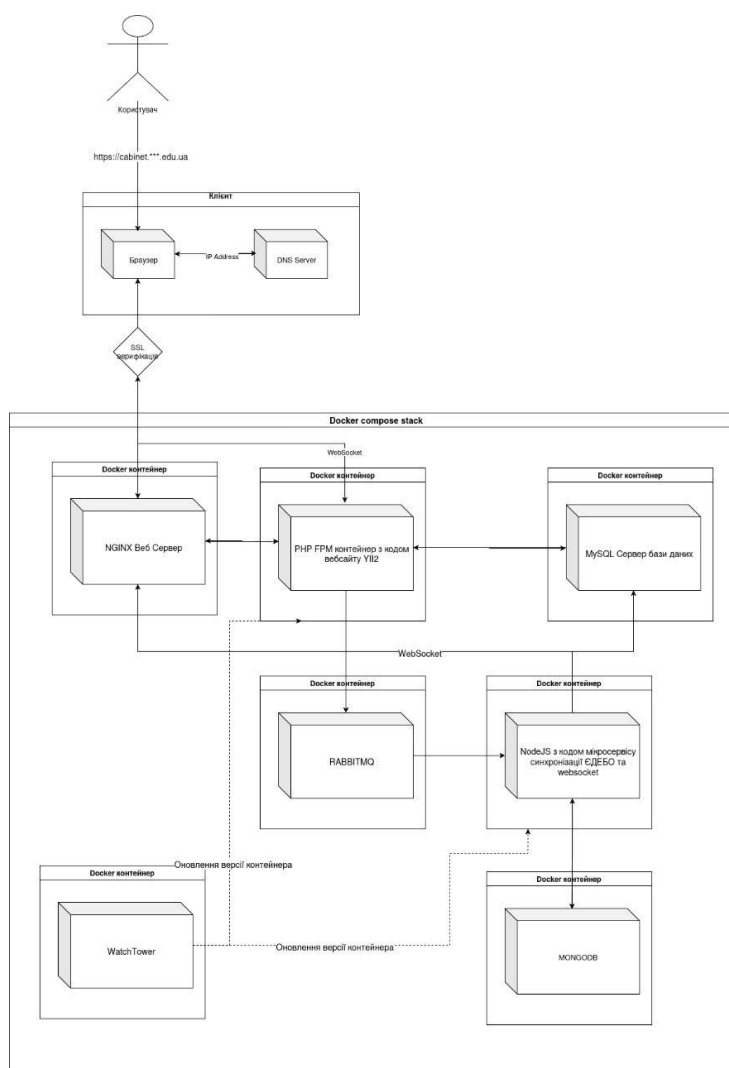


Рисунок 3.1 - Діаграма розгортання додатку

3.2 Структура інтерфейсу додатку

Для користування додатком спочатку виконується його розгортання, доступ до якого мають лише розробники проєкту. Цей аспект є дуже важливим та забезпечує максимальну універсальність та масштабованість рішення.

В першу чергу, ми повинні створити Pull/Merge Request на гілку dev/main та змерджити (рис.3.2):

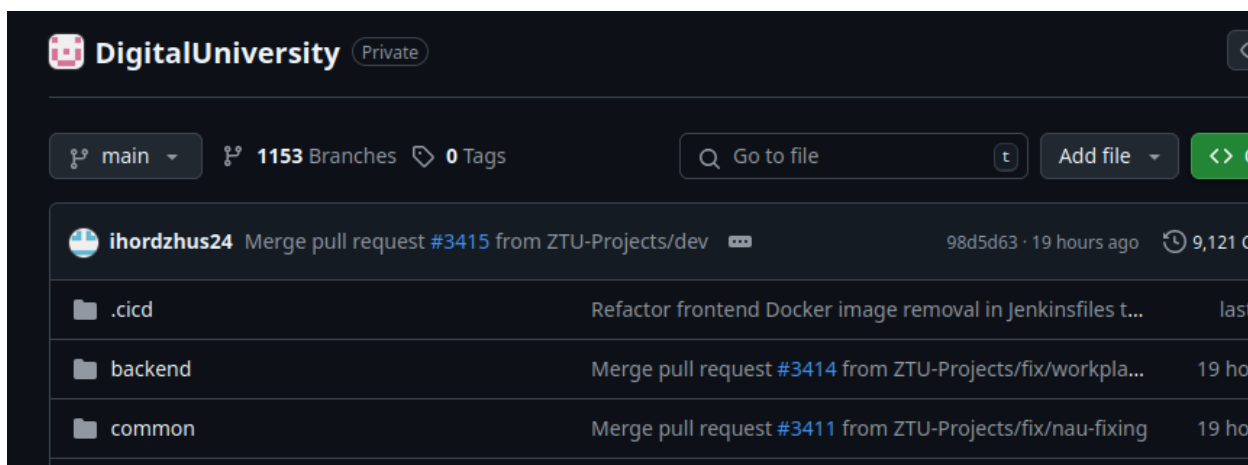


Рисунок 3.2 – Виконання pull request

Після ми повинні перейти на Jenkins та запустити процес збірки пайплайну, під час якого буде протестовано код та створено і запущено Артефакт (Docker image) до реєстру (рис.3.3):

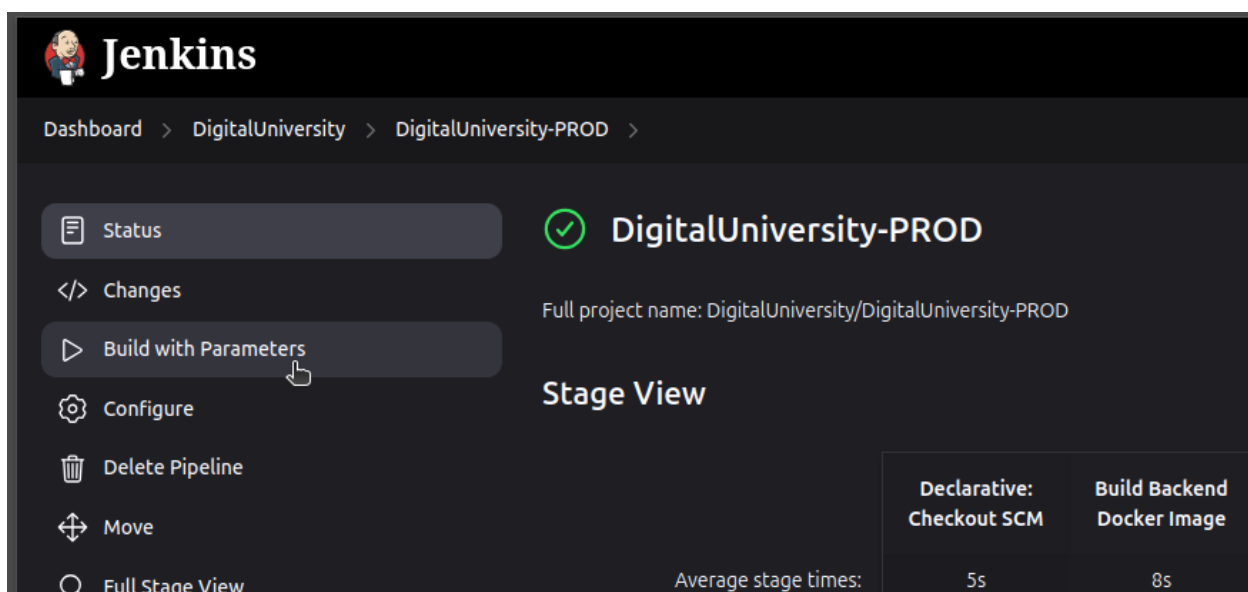


Рисунок 3.3 – Jenkins

Вказуються потрібні параметри для кастомізації білду пайплайну та натискається кнопка Build (рис.3.4):

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		51

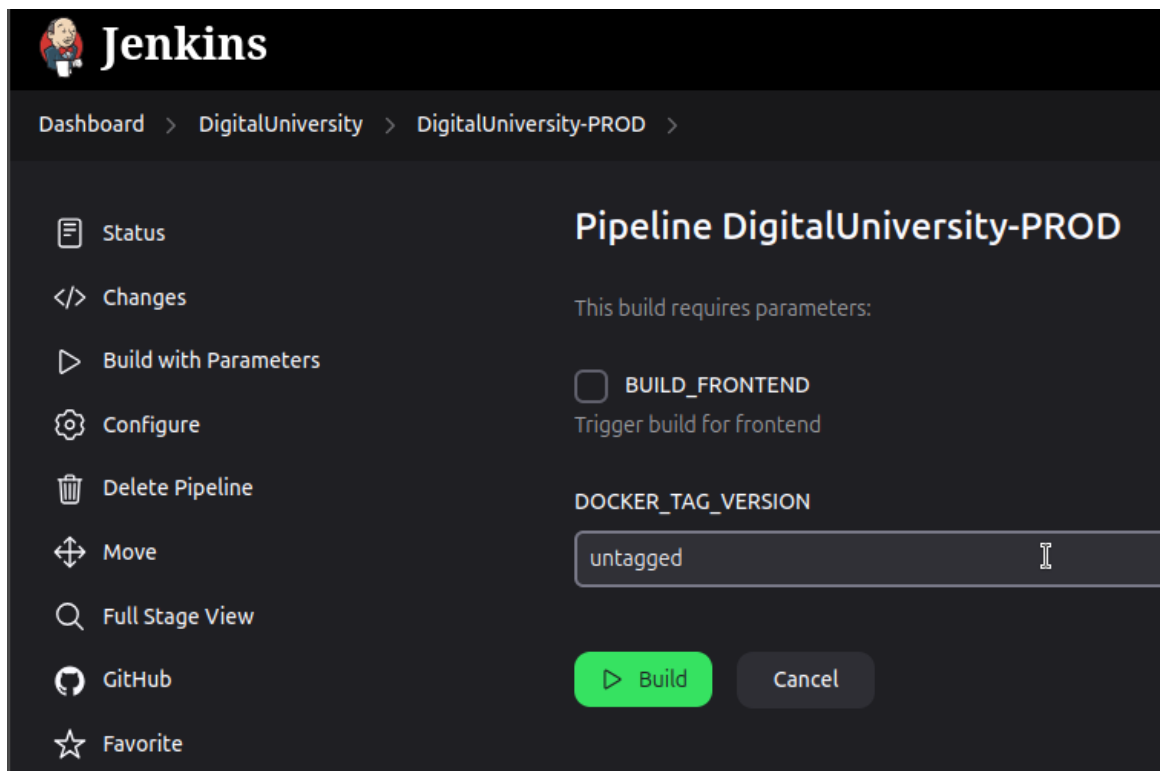


Рисунок 3.4 – Збірка пайплайну на Jenkins

Запускається процес білду, який займає декілька хвилин, після чого потрібно дочекатись його закінчення (рис. 3.5):

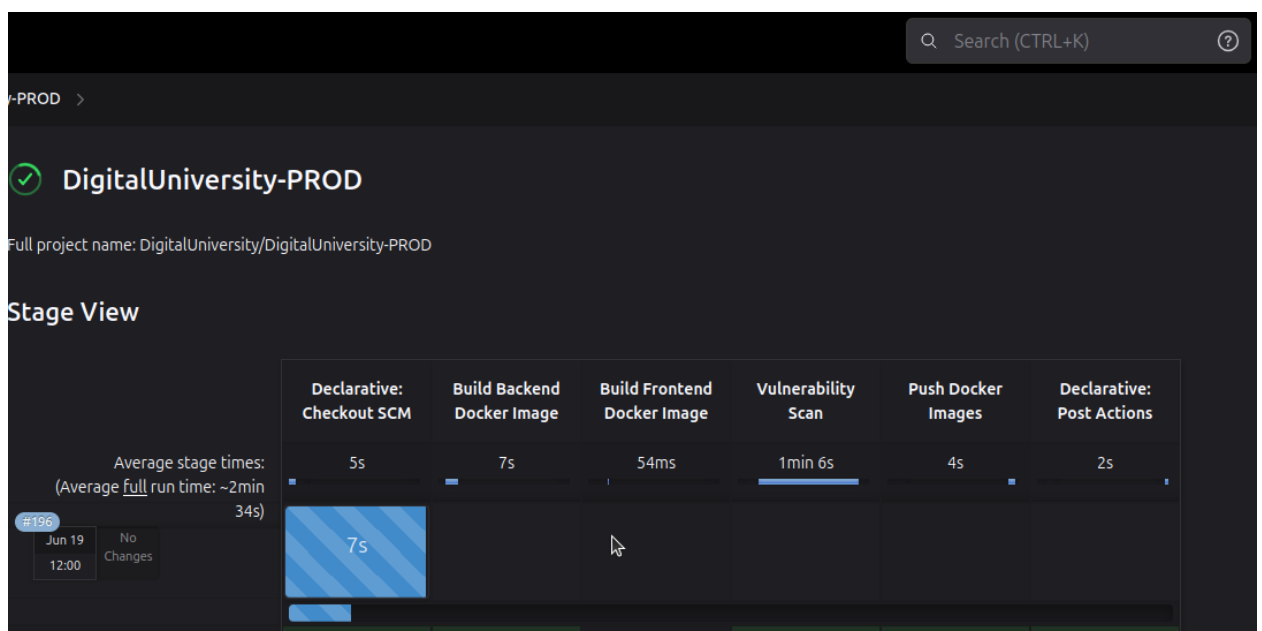


Рисунок 3.5 – Процес збірки

Після закінчення збірки пайплайну оновлений образ Docker був завантажений на реєстр, потрібно почекати перезавантаження контейнеру проекту та перейти на клієнтський сервер, щоб перевірити оновлення (рис. 3.6):

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		52

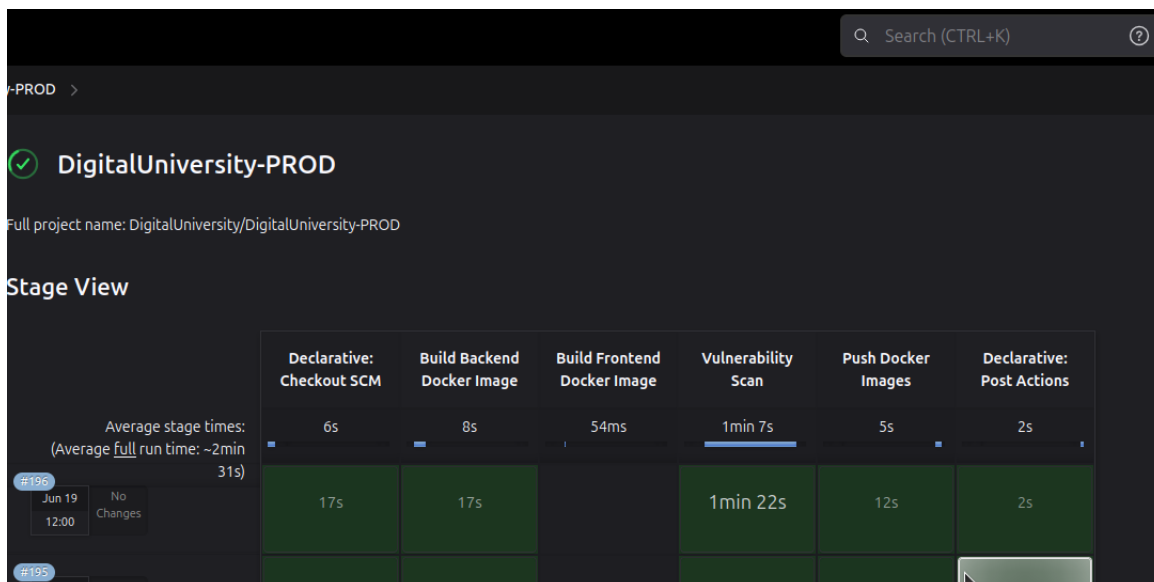


Рисунок 3.6 – Зібраний проект

Watchtower повідомив, що контейнер було успішно оновлено, оскільки параметр в консолі Updated дорівнює 1, що означає, що один контейнер був щойно оновлений (рис. 3.7):

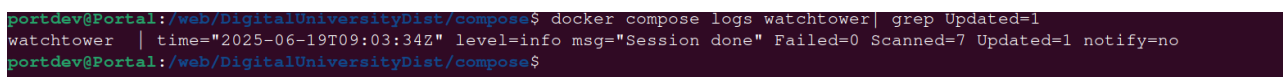


Рисунок 3.7 – Інформація про оновлення контейнеру

Завдяки проведеній DevOps-інфраструктурі, забезпечується можливість вчасних оновлень програмного продукту та стабільний доступ до нього з клієнтської частини.

Перейдемо до тестування додатку. При вході в систему користувачеві відображається головна панель веб-порталу, яка розроблена з адаптивністю до різних розмірів екрану [10] (рис.3.8):

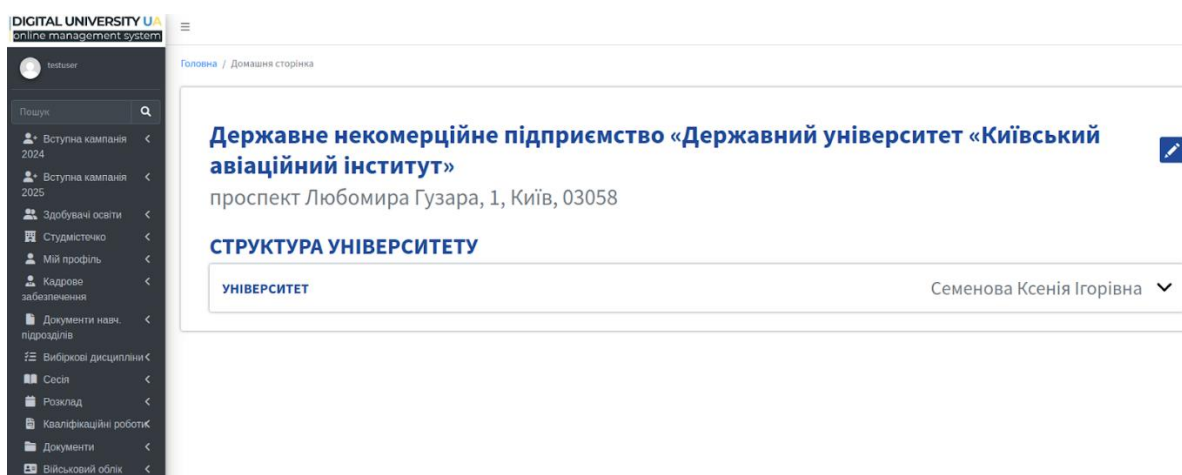


Рисунок 3.8 - Головна сторінка порталу

Буде розглядатись модуль “Здобувачі освіти, що складається із декількох підмодулів (рис. 3.9):

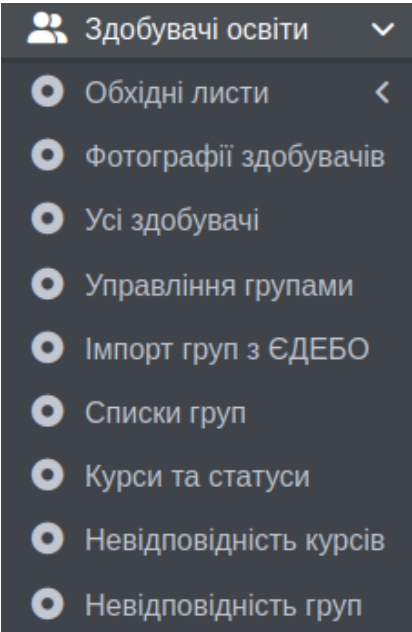


Рисунок 3.9 - Модуль “Здобувачі освіти”

Спочатку адміністрація повинна імпортувати академічні групи в систему, таким чином автоматизуючи процес створення груп та їх подальше заповнення студентами (3.10-3.11):

Групи, яких немає в ІС

Загальна к-сть: 9

Імпортувати

Експорт

Пошук по групі

Виберіть форму...

Виберіть ступінь...

Виберіть курс...

Виберіть спеціальність...

Виберіть освітню програму...

☐ Група	Кількість студентів в групі	Форма	Ступінь	Курс	Початок навчання	Завершення навчання	Спеціальність	Програма
☐ Phd-075-20-1	1	Денна	Доктор філософії	4	07.11.2024	30.09.2025	075 Маркетинг	Маркетинг
☐ Phd-081-20-1	1	Денна	Доктор філософії	2	07.11.2024	01.07.2026	081 Право	Право
☐ Phd-142-20-1	1	Денна	Доктор філософії	4	07.11.2024	31.12.2025	142 Енергетичне машинобудування	Енергетичне машинобудування
☐ Phde-051-21-1	1	Вечірня	Доктор філософії	3	01.10.2021	30.09.2026	051 Економіка	Економіка
☐ Phde-054-20-1	1	Вечірня	Доктор філософії	4	01.10.2020	30.09.2025	054 Соціологія	Соціологія
☐ Phde-152-20-1	1	Вечірня	Доктор філософії	4	07.11.2024	30.09.2025	152 Метрологія та інформаційно-вимірювальна техніка	Метрологія та інформаційно-вимірювальна техніка
☐ Baiz-073-22-2-ЗД	10	Заочна	Бакалавр	3	27.10.2022	28.02.2027	073 Менеджмент	Менеджмент зовнішньоекономічної діяльності
☐ Бзст-051-24-1-МЕ	2	Заочна	Бакалавр	1	01.09.2024	28.02.2028	051 Економіка	Міжнародна економіка
☐ Бзст-075-24-1-МР	7	Заочна	Бакалавр	1	01.09.2024	28.02.2028	075 Маркетинг	Маркетинг

Групи, яких немає в ІС

Загальна к-сть: 9

Імпортувати

Пошук по групі

Виберіть форму...

Виберіть ступінь...

Виберіть курс...

Виберіть спеціальність...

Виберіть освітню програму...

☐ Група	Кількість студентів в групі	Форма	Ступінь	Курс	Початок навчання	Завершення навчання	Спеціальність	Програма
☒ Phd-075-20-1	1	Денна	Доктор філософії	4	07.11.2024	30.09.2025	075 Маркетинг	Маркетинг

Рисунок 3.10-3.11 - Імпортування груп

Як наслідок, на сторінці «Академічні групи» імпортований запис з'явився вгорі сторінки (рис. 3.12):

Академічні групи

+Показані 1-15 із 1107 записів.

Експорт

#	Назва	Курс	Статус	Супільн	Форма	Факультет	Початок навчання	Завершення навчання	Підгрупи	К-ть здобувачів	Пл
			X		X		X				
1	PdD-075-20-1	4	✓	Доктор філософії	Денна	Факультет економіки та бізнес-адміністрування	07.11.2024	30.09.2025	1	1	
2	Bx-192-23-1-PB	2	✓	Бакалавр	Зачнна	Факультет архітектури, будівництва та дизайну	01.09.2023	28.02.2028	1	0	
3	Bx-054-21-1-SL	4	✓	Бакалавр	Зачнна	Факультет психології, комунікацій та перекладу	15.09.2020	29.02.2028	1	0	
4	Bai-272-21-1-TO	4	✓	Бакалавр	Денна	Аерокосмічний факультет	01.09.2019	30.06.2025	1	13	
5	B-292-24-1-MB	1	✓	Бакалавр	Денна	Факультет права та міжнародних відносин	01.09.2024	30.06.2028	1	0	
6	B-291-24-1-MI	1	✓	Бакалавр	Денна	Факультет права та міжнародних відносин	01.09.2024	30.06.2028	1	0	
7	RnDz-293-23-1	2	✓	Доктор філософії	Зачнна	Факультет права та міжнародних відносин	01.04.2023	01.04.2027	1	2	
8	RnDz-293-22-1	3	✓	Доктор філософії	Зачнна	Факультет права та міжнародних відносин	01.04.2022	01.04.2026	1	6	
9	RnDz-292-22-1	3	✓	Доктор філософії	Зачнна	Факультет права та міжнародних відносин	01.10.2022	01.10.2026	1	1	
10	RnDz-281-24-1	1	✓	Доктор філософії	Зачнна	Факультет економіки та бізнес-адміністрування	01.10.2024	01.10.2028	1	1	
11	RnDz-281-22-1	3	✓	Доктор філософії	Зачнна	Факультет економіки та бізнес-адміністрування	01.01.2022	01.01.2026	1	3	
12	RnDz-272-23-1	2	✓	Доктор філософії	Зачнна	Факультет авіонавігації, електроніки та телекомунікацій	01.01.2023	01.01.2027	1	1	

Рисунок 3.12 - Академічні групи

Для орієнтування серед великої кількості груп було додано можливість фільтрації за кожним із полів, що спрощує необхідності групу також можна створити вручну (рис. 3.13):

Створення групи

Назва

Активна

Освітній ступінь

Форма навчання

Початок навчання

Завершення навчання

К-ть підгруп

Курс навчання

Плановий обсяг набору

Оберіть спеціальність...

Оберіть освітню програму...

Факультет

Випускова кафедра

Зберегти

Відмінити

Рисунок 3.13 - Ручне створення академічної групи

Крім створення та редагування академічних груп, наявний перегляд їх студентів. У лівій частині екрану розміщені усі студенти даної групи, у правій - ті, хто не мають запису про приналежність до жодної з груп, однак за потреби

можуть бути переведені. Підтягування студентів, що не мають групи, здійснюється за їх ключовими ознаками, а саме, їх освітній ступінь, форма навчання та освітня програма (рис. 3.14):

01.09.2023 - 30.06.2027

Б-134-23-1-ОС

АКФ

134 Авіаційна та ракетно-космічна техніка

Обладнання повітряних суден

2 курс

Денна

Бакалавр

Підгруп: 1

Студенти групи

Оберіть групу

Змінити групу

☐	#	ПІБ	Дата народження	Фінансування	Термін навчання	Поч. навчання	Зав. навчання	☐
☐	1	Байдала Олександр Сергійович	18.01.2006	Бюджет	Повний	01.09.2023	30.06.2027	☐
☐	2	Білий Денис Юрійович	02.11.2005	Бюджет	Повний	01.09.2023	30.06.2027	☐
☐	3	Бурейко Софія Максимівна	24.12.2004	Бюджет	Повний	01.09.2023	30.06.2027	☐
☐	4	Глібе Софія Сергіївна	01.07.2006	Бюджет	Повний	01.09.2023	30.06.2027	☐
☐	5	Гончаренко Ігор Андрійович	17.06.2003	Бюджет	Повний	01.09.2023	30.06.2027	☐
☐	6	Данильченко Максим Юрійович	27.08.2005	Бюджет	Повний	01.09.2023	30.06.2027	☐
☐	7	Даченко Денис Вікторович	30.07.2006	Бюджет	Повний	01.09.2023	30.06.2027	☐
☐	8	Кобилінський Дмитро Валерійович	09.09.2006	Бюджет	Повний	01.09.2023	30.06.2027	☐
☐	9	Мулач Георгій Вікторович	29.04.2006	Бюджет	Повний	01.09.2023	30.06.2027	☐
☐	10	Опаленіс Валерій Михайлович	23.02.2006	Бюджет	Повний	01.09.2023	30.06.2027	☐
☐	11	Панас Ангеліна Станіславівна	29.11.2005	Бюджет	Повний	01.09.2023	30.06.2027	☐

Студенти без групи

Пошук за ПІБ...

Додати до групи

☐	#	ПІБ	Дата народження	Фінансування
☐	1	Старушко Владислав Сергійович	25.11.2003	Контракт

Рисунок 3.14 - Студенти обраної групи

Якщо перейти на сторінку «Невідповідність груп», ми побачимо цю ж людину, а саме порівняння інформації з ЄДЕБО та в нашій інформаційній системі. За потреби, можливо здійснити переведення вибором відповідної групи із випадного списку та натисканням на кнопку «Змінити групу» (рис. 3.15);

Головна / Здобувачі освіти / Конфлікти груп студентів									
Конфлікти груп студентів									
Показані 1-15 із 81 запису.									
Змінити групу									
#	ID картки	ПІБ	Група в ЄДЕБО	Група в ІС	Оберіть групу	Факультет	Початок навчання	Завершення навчання	Скорочений термін
1	12387243	Старушко Владислав Сергійович	Б-134-23-1-ОС	(не зайдено)	Оберіть групу...	Аерокосмічний факультет	01.09.2021	30.06.2027	×
2	12371109	Абдураманов Азиз	Баз-073-22-2-ЗД	Баз-073-22-2-ЗД	Оберіть групу...	Факультет економіки та бізнес-адміністрування	10.11.2022	28.02.2027	×
3	12371108	Мустафас Рустан	Баз-073-22-2-ЗД	Баз-073-22-2-ЗД	Оберіть групу...	Факультет економіки та бізнес-адміністрування	10.11.2022	28.02.2027	×
4	12371107	Амедова Нурра	Баз-073-22-2-ЗД	Баз-073-22-2-ЗД	Оберіть групу...	Факультет економіки та бізнес-адміністрування	01.11.2022	28.02.2027	×
5	12371106	Дирімова Далер	Баз-073-22-2-ЗД	Баз-073-22-2-ЗД	Оберіть групу...	Факультет економіки та бізнес-адміністрування	27.10.2022	28.02.2027	×

Рисунок 3.15 - Студент без групи

Спробуємо додати студента на сторінці перегляду складу академічної групи. Переведення здійснюється за допомогою натискання на жовту стрілочку вліво. Дана дія змінить інформацію про групу навчання студента локально і синхронізує зміни із ЄДЕБО. На рис. 3.16 зображено поданого студента у складі поточної групи, а справа, відповідно, відсутність здобувачів освіти, що її не мають:

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				56
Змн.	Арк.	№ докум.	Підпис	Дата		

Студенти групи							Оберіть групу	Змінити групу
#	ПІБ	Дата народження	Фінансування	Термін	Поч. навчання	Зав. навчання		
1	Байдала Олександр Сергійович	18.01.2006	Бюджет	Повний	01.09.2023	30.06.2027		
2	Білий Денис Юрійович	02.11.2005	Бюджет	Повний	01.09.2023	30.06.2027		
3	Бурейко Софія Максимівна	24.12.2004	Бюджет	Повний	01.09.2023	30.06.2027		
4	Глібо Софія Сергіївна	01.07.2006	Бюджет	Повний	01.09.2023	30.06.2027		
5	Гончаренко Ігор Андрійович	17.06.2003	Бюджет	Повний	01.09.2023	30.06.2027		
6	Данильченко Максим Юрійович	27.08.2005	Бюджет	Повний	01.09.2023	30.06.2027		
7	Деченко Денис Вікторович	30.07.2006	Бюджет	Повний	01.09.2023	30.06.2027		
8	Кобилінський Дмитро Валерійович	09.09.2006	Бюджет	Повний	01.09.2023	30.06.2027		
9	Мулач Георгій Вікторович	29.04.2006	Бюджет	Повний	01.09.2023	30.06.2027		
10	Опаленік Валерій Михайлович	23.02.2006	Бюджет	Повний	01.09.2023	30.06.2027		
11	Панас Ангеліна Станіславівна	29.11.2005	Бюджет	Повний	01.09.2023	30.06.2027		
12	Паньків Микола Олександрович	19.12.2005	Бюджет	Повний	01.09.2023	30.06.2027		
13	Торчилов Артем Костянтинівич	17.07.2005	Бюджет	Повний	01.09.2023	30.06.2027		
14	Усик Антон Андрійович	12.01.2006	Бюджет	Повний	01.09.2023	30.06.2027		
15	Христофоров Богдан Васильович	30.08.2005	Бюджет	Повний	01.09.2023	30.06.2027		
16	Старушко Владислав Сергійович	25.11.2003	Контракт	Повний	01.09.2021	30.06.2027		

Студенти без групи				Пошук за ПІБ...	Додати до групи
#	ПІБ	Дата народження	Фінансування		
Нічого не знайдено.					

Рисунок 3.16 - Переведений студент

У списку невідповідностей груп ці зміни також будуть зафіксовані, і розглянутий студент зникне із переліку (рис. 3.17):

Головна / Здобувачі освіти / Конфлікти груп студентів									
Конфлікти груп студентів Показані 1-15 із 80 записів.									
Змінити групу									
#	ІД картки	ПІБ	Група в ЄДЕБО	Група в ІС	Оберіть групу	Факультет	Початок навчання	Завершення навчання	Скорочений термін
1	12371109	Абдурахманов Аваз	Баз-073-22-2-ЗД	Баз-073-22-2-ЗД	Оберіть групу...	Факультет економіки та бізнес-адміністрування	10.11.2022	28.02.2027	×
2	12371108	Мустафаєв Рустам	Баз-073-22-2-ЗД	Баз-073-22-2-ЗД	Оберіть групу...	Факультет економіки та бізнес-адміністрування	10.11.2022	28.02.2027	×
3	12371107	Ахмедова Нодіра	Баз-073-22-2-ЗД	Баз-073-22-2-ЗД	Оберіть групу...	Факультет економіки та бізнес-адміністрування	01.11.2022	28.02.2027	×
4	12371106	Даміров Далеє	Баз-073-22-2-ЗД	Баз-073-22-2-ЗД	Оберіть групу...	Факультет економіки та бізнес-адміністрування	27.10.2022	28.02.2027	×
5	12371105	Цейтук Крістіна	Баз-073-22-2-ЗД	Баз-073-22-2-ЗД	Оберіть групу...	Факультет економіки та бізнес-адміністрування	27.10.2022	28.02.2027	×

Рисунок 3.17 - Зміни в конфліктах груп студентів

Сторінка “Управління курсами та статусами” надає аналогічний функціонал для перегляду контингенту групи, однак додатково забезпечує можливість зміни курсу навчання студента, синхронізуючи це з ЄДЕБО. виправимо неточність курсу у студента Матієва Яна Сергіївна (рис. 3.18-3.19):

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				57
Змн.	Арк.	№ докум.	Підпис	Дата		

01.04.2022 - 01.04.2026

PHD3-293-22-1

293 Міжнародне право

3 курс

Заочна

Доктор філософії

Підгруп: 1

Студенти групи		На 2-й курс	На 4-й курс				
#	ПІБ	Курс	Дата народження	Фінансування	Термін	Поч. навчання	Зав. навчання
1	Восводіна Ганна Іванівна	3	25.12.1985	Контракт	Повний	01.10.2022	01.10.2026
2	Євдокименко Тетяна Миколаївна	3	31.05.1984	Контракт	Повний	01.07.2022	01.07.2026
3	Матієва Яна Сергіївна	4	21.07.1974	Контракт	Повний	01.04.2022	01.04.2026
4	Мурачов Роман Вікторович	3	18.06.1988	Контракт	Повний	01.07.2022	01.07.2026
5	Науменко Наталія Василівна	3	24.12.1974	Контракт	Повний	01.07.2022	01.07.2026
6	Ничипоренко Георгій Валентинович	3	15.05.1999	Контракт	Повний	01.10.2022	01.10.2026

01.04.2022 - 01.04.2026

PHD3-293-22-1

293 Міжнародне право

3 курс

Заочна

Доктор філософії

Підгруп: 1

Студенти групи		На 2-й курс	На 4-й курс				
#	ПІБ	Курс	Дата народження	Фінансування	Термін	Поч. навчання	Зав. навчання
1	Восводіна Ганна Іванівна	3	25.12.1985	Контракт	Повний	01.10.2022	01.10.2026
2	Євдокименко Тетяна Миколаївна	3	31.05.1984	Контракт	Повний	01.07.2022	01.07.2026
3	Матієва Яна Сергіївна	3	21.07.1974	Контракт	Повний	01.04.2022	01.04.2026
4	Мурачов Роман Вікторович	3	18.06.1988	Контракт	Повний	01.07.2022	01.07.2026
5	Науменко Наталія Василівна	3	24.12.1974	Контракт	Повний	01.07.2022	01.07.2026
6	Ничипоренко Георгій Валентинович	3	15.05.1999	Контракт	Повний	01.10.2022	01.10.2026

Рисунок 3.18-3.19 - Зміна курсу студента

Також можливо перевести усіх студентів на наступний курс натисканням відповідної кнопки (рис. 3.20):

Студенти групи		На 2-й курс	На 5-й курс				
#	ПІБ	Курс	Дата народження	Фінансування	Термін	Поч. навчання	Зав. навчання
1	Восводіна Ганна Іванівна	4	25.12.1985	Контракт	Повний	01.10.2022	01.10.2026
2	Євдокименко Тетяна Миколаївна	4	31.05.1984	Контракт	Повний	01.07.2022	01.07.2026
3	Матієва Яна Сергіївна	4	21.07.1974	Контракт	Повний	01.04.2022	01.04.2026
4	Мурачов Роман Вікторович	4	18.06.1988	Контракт	Повний	01.07.2022	01.07.2026
5	Науменко Наталія Василівна	4	24.12.1974	Контракт	Повний	01.07.2022	01.07.2026
6	Ничипоренко Георгій Валентинович	4	15.05.1999	Контракт	Повний	01.10.2022	01.10.2026

Рисунок 3.20 - Зміна курсу групи

Для отримання більшої кількості інформації про здобувачів освіти було розроблено сторінку “Усі здобувачі”, де застосовано оптимізаційні рішення для оперативного відображення даних студентів (рис. 3.21):

Головна / Здобувачі освіти / Всі здобувачі

Всі здобувачі
Показані 1-15 із 9 036 записів.

Експорт Активне навчання

#	ПІБ	Дата народження	ID картки	Група(ІС)	Група(ЄДЕБО)	Курс	Основа вступу	Освітній ступінь	Осм. Фін
1	Бажинов Дмитро Романович	17.05.1995	12356766	Б-075-21-1-МР	Б-075-21-1-МР	4	Повна загальна середня освіта	Бакалавр	Контракт
2	Камінський Олександр Васильович	16.09.1996	12367059	Бз-263-22-1-КІ	Бз-263-22-1-КІ	3	Повна загальна середня освіта	Бакалавр	Контракт
3	Войцехівська Надія Ігорівна	11.10.1996	12357374	Бз-162-21-1-ФБ	Бз-162-21-1-ФБ	4	Повна загальна середня освіта	Бакалавр	Контракт
4	Колошинський Іштван	24.04.1993	12364820	Б-125-22-4-БІ	Б-125-22-4-БІ	3	Повна загальна середня освіта	Бакалавр	Контракт
5	Коломієць Віта Сергіївна	01.09.1994	12364983	Бз-075-23-1-МР	Бз-075-23-1-МР	2	Повна загальна середня освіта	Бакалавр	Контракт
6	Ткаченко Олександр Васильович	20.04.1993	12363564	-	Б-192-23-1-ЕА	2	Повна загальна середня освіта	Бакалавр	Контракт
7	Сальцін Лев Сергійович	25.10.1993	12360098	Бз-174-23-1-ІЗ	Бз-174-23-1-ІЗ	2	Повна загальна середня освіта	Бакалавр	Контракт
8	Охріменко Микита Сергійович	20.01.1996	12366930	Бз-263-23-1-КІ	Бз-263-23-1-КІ	2	Повна загальна середня освіта	Бакалавр	Контракт
9	Єрмаков Роман Сергійович	03.03.1996	12366521	Б-263-23-1-КІ	Б-263-23-1-КІ	2	Повна загальна середня освіта	Бакалавр	Контракт

Рисунок 3.21 - Перегляд усіх здобувачів освіти

Наявний фільтр за типом їх навчання - забезпечено розділення студентів на тих, хто наразі навчається, відраховані, а також ті, хто його успішно завершили.

Кожен із студентів має власний кабінет, де йому надається доступ до декількох операцій, в тому числі й завантаження власної фотографії для документів. В подальшому вона може бути застосована з метою створення студентського квитка, однак спочатку вона має пройти процес верифікації комісією (рис. 3.22):

Головна / Здобувачі освіти / Фотографії здобувачів

Фотографії здобувачів
Показані 1-20 із 474 записів.

#	Дії	Статус	Час	ПІБ	Фото	Група	Курс	Освітній ступінь	Спеціальність	Факультет
1	✓ ✕	Нова фотографія	19.06.2025 19:48	Оленич Олександра Олегівна		Бз-162-21-1-ФБ	4 Курс	Бакалавр	162 Біотехнології та біоінженерія	Факультет екологічної безпеки, інженерії та енергетики
2	✓ ✕	Нова фотографія	18.06.2025 19:24	Довженко Оксана Русланівна		Бст-072-22-1-ФК	3 Курс	Бакалавр	072 Фінанси, банківська справа та страхування	Факультет економіки та бізнес-адміністрування
3	✓ ✕	Нова фотографія	18.06.2025 10:56	Довгалик Андрій Сергійович		Бз-173-21-2-АВ	4 Курс	Бакалавр	173 Авіоніка	Факультет авіонавігації, електроніки та інформаційних технологій
4	✓ ✕	Нова фотографія	17.06.2025 17:28	Кулішов Ілля Ігорович		Б-121-22-1-ПІ	3 Курс	Бакалавр	121 Інженерія програмного забезпечення	Факультет комп'ютерних наук та інформаційних технологій
5	✓ ✕	Нова фотографія	17.06.2025 16:59	Савчук Олександр Олександрович		Б-281-22-2-ПУ	4 Курс	Бакалавр	281 Публічне управління та адміністрування	Факультет економіки та бізнес-адміністрування
6	✓ ✕	Нова фотографія	17.06.2025 11:02	Максимчук Андрій Олександрович		Б-151-21-1-АТ	4 Курс	Бакалавр	151 Автоматизація та комп'ютерно-інтегровані технології	Аерокосмічний факультет
7	✓ ✕	Нова фотографія	16.06.2025 11:47	Чебанюк Олексій Юрійович		Бз-151-21-1-ІЗ	4 Курс	Бакалавр	151 Автоматизація та комп'ютерно-інтегровані технології	Факультет авіонавігації, електроніки та інформаційних технологій
8	✓ ✕	Нова фотографія	16.06.2025 10:43	Клишинець Дмитро Дмитрович		Б-173-21-2-АВ	4 Курс	Бакалавр	173 Авіоніка	Факультет авіонавігації, електроніки та інформаційних технологій

Рисунок 3.21 - Перегляд надісланих на розгляд фотографій

При натисканні на зелену кнопку відбувається схвалення фотографії, на червону - її відхилення. Протестуємо процес, схваливши одну із фотографій здобувачів освіти (рис. 3.22):

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		
						59

Головна / Здобувачі освіти / Фотографії здобувачів

Фотографії здобувачів						
Показані 1-20 із 474 записів.						
#	Дії	Статус	Час	ПІБ	Фото	Група
1		Схвалена	19.06.2025 19:48	Оленич Олександра Олегівна		Бз-162-21-1-ФБ
2		Нова фотографія	18.06.2025 19:24	Довженко Оксана Русланівна		Бст-072-22-1-ФК

Рисунок 3.22 - Схвалена фотографія

Для забезпечення процесу заповнення обхідних листів студентами необхідно додати структурні підрозділи університету (рис. 3.23):

Структурні підрозділи						
Показані 1-20 із 84 записів.						
#	Структурний підрозділ			Чи активний	Чи є деканатом	Чи є випусковою кафедрою
1			Аерокосмічний факультет	Так	Так	Ні
2			Факультет аеронавігації, електроніки та телекомунікацій	Так	Так	Ні
3			Факультет екологічної безпеки, інженерії та технологій	Так	Так	Ні
4			Факультет економіки та бізнес-адміністрування	Так	Так	Ні
5			Факультет комп'ютерних наук та технологій	Так	Так	Ні

Рисунок 3.23 - Структурні підрозділи

Створення відбувається на основі існуючих структурних підрозділів - кожен із підрозділів для обхідного листа посилається на реальний існуючий в університеті. Кожен може бути активним/неактивним, для визначення, це є деканат чи випускова кафедра - існують поля для заповнення відповідної інформації (рис. 3.24):

Створення структурного підрозділу

Структурний підрозділ

Кафедра авіаційних двигунів

Чи активний

Так

Чи для всіх студентів

Так

Чи є випусковою кафедрою

Так

Чи є деканатом

Ні

Відображення в обхідному листі

Так

Контактна інформація

Створити

Відмінити

Рисунок 3.24 - Створення підрозділу

Якщо у списку відповідальних осіб додати користувача - він зможе виставляти відмітки для обраного підрозділу на відповідній сторінці (рис. 3.25):

Відмітки в обхідних листах						
Показані 1-15 із 227 записів.						
Освітня програма						
#	П.І.Б	Група	Факультет	Форма навчання	Основа фінансування	Аерокосмічний факультет
1	Лео Беньчю	Б-134-21-1-ЛВ	АКФ	Денна	Контракт	-
2	Харчук Софія Борисівна	Б-134-21-1-ЛВ	АКФ	Денна	Бюджет	+
3	Буркот Артемій Валерійович	Б-134-21-1-ЛВ	АКФ	Денна	Бюджет	-

Рисунок 3.25 - Відмітки у обхідних листах

Знак “+” свідчить про відсутність заборгованостей, “-” означає, що студент має певні проблеми, і в такому разі можна написати коментар, де пояснюватиметься причина проблеми.

Відповідні відмітки будуть відображатися і на загальному списку по усіх студентах - як бачимо, студенти мають ті самі відмітки з деканату, що й були виставлені раніше (рис. 3.26):

☒ Перегляд обхідних листів Показані 1-15 із 3 045 записів.									
Освітня програма		X							
#	П.І.Б.	Група	Факультет	Форма навчання	Основа фінансування	Деканат	Випускова кафедра	Сектор планування соціальних витрат	Управління навчально-методичної роботи
1	Лео Бенчикю	Б-134-21-1-ЛВ	АКФ	Денна	Контракт	-	-	-	-
2	Харчук Софія Борисівна	Б-134-21-1-ЛВ	АКФ	Денна	Бюджет	+	-	-	-
3	Буркот Артемій Валерійович	Б-134-21-1-ЛВ	АКФ	Денна	Бюджет	-	-	-	-

Рисунок 3.27 - Перегляд відміток студентів за різними підрозділами

Якщо поточний користувач буде наділений правами редагувати відмітки у декількох підрозділах - до таблиці буде додано ще одну колонку, яка буде доступна лише тим студентам, які здійснюють навчання за освітньою програмою, що пов'язана з певним підрозділом (рис. 3.28):

Освітня програма		X							
#	П.І.Б.	Група	Факультет	Форма навчання	Основа фінансування	Аерокосмічний факультет	Факультет аеронавігації, електроніки та телекомунікацій		
121	Насонов Нікіта Олегович	Бст-151-22-1-АТ	АКФ	Денна	Бюджет	-			
122	Максимчук Андрій Олександрович	Б-151-21-1-АТ	АКФ	Денна	Контракт	-			
123	Садівський Денис Олексійович	Бст-151-22-1-СУ	ФАЕТ	Денна	Бюджет		-		
124	Павліковський Максим Віталійович	Бст-151-22-1-СУ	ФАЕТ	Денна	Бюджет		-		

Рисунок 3.28 - Редагування відміток декількох підрозділів

Таким чином, забезпечується робота адміністрації над забезпеченням освітнього процесу у вигляді менеджменту академічних груп, здобувачів освіти, а також процесу випуску студентів та заповнення їх обхідного листа.

Висновки до третього розділу

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

У межах функціоналу для адміністративного персоналу реалізовано інструменти перегляду заяв вступників із можливістю фільтрації та сортування за різними критеріями — спеціальностями, датами подачі, статусами тощо. Додатково забезпечено доступ до повного переліку документів абітурієнтів, а також історії їх попереднього навчання.

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		62

Суттєвим етапом стала реалізація експорту даних у формат Excel, що забезпечує зручне формування звітів. Передбачено механізм синхронізації з ЄДЕБО для отримання актуальної інформації. Для зручного аналізу реалізовано блок статистики, який дозволяє оцінювати динаміку подачі заяв, кількість заяв за спеціальностями та показники успішності вступників.

Окремо опрацьовано функціонал, що стосується вступних іспитів: реалізовано відображення зареєстрованих осіб, збереження інформації про результати, а також перегляд списку зарахованих вступників із можливістю побудови статистичних звітів.

Результати тестування підтвердили повну працездатність розробленого вебсайту, відповідність заявленим функціональним вимогам та його готовність до подальшої експлуатації. Таким чином, поставлені задачі були успішно реалізовані, що свідчить про досягнення головної мети розробки.

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		63

ВИСНОВКИ

У межах виконання кваліфікаційної роботи було розроблено функціональний модуль «Здобувачі освіти», що є складовою інформаційної системи цифрового менеджменту освітніх процесів. Основна мета розробки полягала в автоматизації ключових адміністративних процедур, пов'язаних із веденням особових справ студентів, контролем їх навчального статусу, обробкою змін у структурі академічних груп та забезпеченням актуальності даних у взаємодії з ЄДЕБО.

На початковому етапі було здійснено аналіз функціональних потреб закладу вищої освіти, що дозволило чітко сформулювати бізнес-вимоги до системи. Зокрема, було визначено необхідність централізованого обліку здобувачів освіти, фіксації їх навчального статусу, управління груповою належністю та формування службових документів — таких як обхідні листи. В основу програмної реалізації покладено архітектуру «клієнт-сервер», що забезпечила масштабованість, безпеку та надійність взаємодії з базою даних та зовнішніми сервісами.

У процесі проєктування було сформовано перелік як функціональних, так і нефункціональних вимог. Система надає можливість оперативного оновлення інформації щодо академічного статусу студентів, управління їхнім переміщенням між групами, а також фіксації подій, пов'язаних з відрахуванням, поновленням та завершенням навчання.

Фінальний етап включав розгортання модуля в тестовому середовищі, верифікацію його функціональності, перевірку сценаріїв синхронізації з ЄДЕБО, а також оцінку ефективності реалізованих механізмів обліку здобувачів освіти. Проведене тестування підтвердило відповідність модуля заданим вимогам, стабільність його роботи та здатність до масштабування.

Таким чином, реалізований модуль «Здобувачі освіти» є ефективним інструментом для автоматизації та контролю освітніх процесів, що сприяє підвищенню прозорості адміністрування, оперативності прийняття рішень та цифровізації внутрішніх процедур у закладі вищої освіти.

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		64

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ed Tech в освіті: переваги, ризики та перспективи в Україні / Кушнір Л., Хмеляр І., Шелевер О. – Актуальні питання гуманітарних наук, 2024. – 7 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. Google Classroom [Електронний ресурс] – Режим доступу до ресурсу: <https://sites.google.com/view/classrooms-workspace/>
5. Blackboard Learn [Електронний ресурс] <https://www.anthology.com/products/teaching-and-learning/learning-effectiveness/blackboard>
6. Modern PHP: New Features and Good Practices / Josh Lockhart – O'Reilly Media, 2023. – 320 с.
7. Mastering PHP Design Patterns / Junade Ali – Packt Publishing, 2022. – 282 с.
8. The Art of Node: Mastering Non-blocking JavaScript / Max Ogden – Pragmatic Bookshelf, 2023. – 270 с.
9. Learning JavaScript Design Patterns / Addy Osmani – O'Reilly Media, 2023. – 360 с.
10. Web Design with HTML, CSS, JavaScript and jQuery Set / Jon Duckett – Wiley, 2022. – 1152 с.
11. Designing Microservices Platforms with NATS / Chanaka Fernando – Packt Publishing, 2023. – 390 с.
12. Fundamentals of Software Architecture / Mark Richards, Neal Ford – O'Reilly Media, 2022. – 432 с.
13. Grokking Algorithms: An Illustrated Guide for Programmers and Other Curious People / Aditya Bhargava – Manning Publications, 2022. – 304 с.

		Цалко Я.В.			ІІЗ.КР.Б – 121 – 25 – ІІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		65

14. Algorithms Illuminated (Part 1–4) / Tim Roughgarden – Soundlikeyourself Publishing, 2021–2023. – ~1300 с. MySQL Crash Course / Ben Forta – Addison-Wesley, 2024. – 400 с.
15. PHP, MySQL and Apache / Julie C. Meloni – Peachpit Press, 2022. – 512 с.
16. Jenkins: The Definitive Guide: Continuous Integration for the Masses / John Ferguson Smart – O’Reilly Media, 2023. – 420 с.
17. Learning GitHub Actions: Automate your workflow from idea to production / Brent Laster – O’Reilly Media, 2023. – 344 с.
18. SonarQube in Action: Code Quality and Security for DevOps Teams / G. Ann Campbell, Patroklos Papapetrou – Manning Publications, 2022. – 312 с.
19. Docker Deep Dive / Nigel Poulton – Leanpub, 2023. – 468 с.
20. The Docker Book: Containerization is the new virtualization / James Turnbull – James Turnbull, 2023. – 350 с.
21. Mastering Docker: Unlock new opportunities using Docker's most advanced features / Russ McKendrick – Packt Publishing, 2023. – 570 с.
22. Securing Docker: Security best practices for Docker, containers, and the cloud / Scott Surovich, Rex McMillan – Packt Publishing, 2022. – 408 с.
23. Automating DevOps with GitLab CI/CD Pipelines / Christopher Cowell – Packt Publishing, 2023. – 350 с.
24. DevSecOps with Docker and Kubernetes: Secure Your Container Infrastructure / Jose Manuel Ortega – Packt Publishing, 2023. – 366 с.
25. Infrastructure as Code, 2nd Edition: Dynamic Systems for the Cloud Age / Kief Morris – O’Reilly Media, 2023. – 480 с.
26. Watchtower Documentation [Электронный ресурс] – Режим доступа: <https://containrrr.dev/watchtower/>
27. Trivy Documentation [Электронный ресурс] – Режим доступа: <https://aquasecurity.github.io/trivy/>
28. Web Application Development with Yii 2 and PHP / Mark Safronov, Jeffrey Winesett – Packt Publishing, 2014. – 280 с.

		Цалко Я.В.			ІІЗ.КР.Б – 121 – 25 – ІІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		66

29. Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation / Jez Humble, David Farley – Addison-Wesley, 2010. – 512 с.

30. Інструкції по роботі з ЄДЕБО [Електронний ресурс] – Режим доступу до ресурсу: <https://www.inforesurs.gov.ua/edebo/instructions/>

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		67

ДОДАТКИ

		Цалко Я.В.			ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		68

```

pipeline {
    agent any
    parameters{
        booleanParam(name: 'BUILD_FRONTEND', defaultValue: false,
description: 'Trigger build for frontend')
    }
    environment {
        NODE_OPTIONS="--max-old-space-size=512"
        DOCKER_REGISTRY = 'registry.ztu.edu.ua'
        DOCKER_IMAGE = 'digital-university/prod'
        DOCKER_IMAGE_FRONTEND = 'digital-university-nginx/prod'
        DOCKER_TAG = 'latest'
        DOCKER_CREDENTIALS = 'docker-login'
    } // environment

    stage('Build Frontend Docker Image') {
        when {
            expression { params.BUILD_FRONTEND }
        } // when
        steps {
            script {
                docker.withRegistry("https://${DOCKER_REGISTRY}/"
, "${DOCKER_CREDENTIALS}") {
                    dockerImageFrontend =
docker.build("${DOCKER_REGISTRY}/${DOCKER_IMAGE_FRONTEND}:${DOCKER_TA
G}", "-f Dockerfile.frontend .")
                } // docker
            } // script
        } // steps
    } // stage
    stage('Vulnerability Scan') {
        steps {
            script {
                sh """
                    trivy image --exit-code 1 --severity
HIGH,CRITICAL ${DOCKER_REGISTRY}/${DOCKER_IMAGE}:${DOCKER_TAG} ||
true
                """
            } // script
        } // steps
    } // stage

    stage('Push Docker Images') {
        steps {
            script {
                docker.withRegistry("https://${DOCKER_REGISTRY}/"
, "${DOCKER_CREDENTIALS}") {
                    dockerImage.push()
                    dockerImage.push("${params.DOCKER_TAG_VERSION
}")

                    if (params.BUILD_FRONTEND) {
                        dockerImageFrontend.push()
                    }
                }
            }
        }
    }
}

```

		Цалко Я.В.			ІІЗ.КР.Б – 121 – 25 – ІІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		69

```

        } // docker
    } // script
} // steps
} // stage
} // stages

services:
  website:
    container_name: digital-${UNI_NAME}-website
    image: dregistry.ztu.edu.ua/digital-university/dev:latest
    depends_on:
      - mysql
    volumes:
      -
        ${WEB_SOCKET_CONFIG_PATH}:/var/www/diguni/backend/web/scripts/config/
        config.js
        - ${PHP_CONFIG_PATH}:/var/www/diguni/common/config/main-
        local.php
        - files:/var/www/diguni/files
        - edebo_files:/var/www/Files
        - backend_web:/var/www/diguni/backend/web
        - frontend_web:/var/www/diguni/frontend/web
    ports:
      - 9000:9000
    networks:
      - digital-university-private-network
      - digital-university-public-network
    labels:
      - "com.centurylinklabs.watchtower.enable=true"

  rabbitmq:
    container_name: digital-${UNI_NAME}-rabbitmq
    image: rabbitmq:3
    volumes:
      - rabbitmq_data:/var/lib/rabbitmq
    environment:
      RABBITMQ_DEFAULT_USER: ${RABBITMQ_DEFAULT_USER}
      RABBITMQ_DEFAULT_PASS: ${RABBITMQ_DEFAULT_PASS}
    networks:
      - digital-university-private-network

  edebo:
    container_name: digital-${UNI_NAME}-edebo-microservice
    image: registry.ztu.edu.ua/digital-university-node-
    service/dev:latest
    depends_on:
      - mysql
      - mongodb
      - rabbitmq
    volumes:
      - edebo_files:/Files
    env_file:
      - configs/diguni-edebo/.env
    networks:
      - digital-university-private-network

```

		Цалко Я.В.			ІІЗ.КР.Б – 121 – 25 – ІІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		70

```

labels:
  - "com.centurylinklabs.watchtower.enable=true"

mysql:
  container_name: digital-${UNI_NAME}-mysql-server
  image: diguni-basic-mysql-image:latest
  volumes:
    - mysql_data:/var/lib/mysql
  restart: always
  environment:
    MYSQL_DATABASE: ${MYSQL_DATABASE}
    MYSQL_ROOT_PASSWORD: ${MYSQL_ROOT_PASSWORD}
  ports:
    - ${MYSQL_PORT_FORWARD}:3306
  networks:
    - digital-university-private-network

mongodb:
  container_name: digital-${UNI_NAME}-mongodb-server
  image: mongo:5.0
  restart: unless-stopped
  ports:
    - "27017:27017"
  environment:
    MONGO_INITDB_ROOT_USERNAME: ${MONGODB_ROOT_USER}
    MONGO_INITDB_ROOT_PASSWORD: ${MONGODB_ROOT_PASS}
  volumes:
    - mongo_data:/data/db
  networks:
    - digital-university-private-network

nginx:
  container_name: digital-${UNI_NAME}-nginx
  image: nginx:1.27.2
  volumes:
    - ${NGINX_CONFIGS_FOLDER_PATH}:/etc/nginx/conf.d/
    - ${PORTAL_SSL_FOLDER_PATH}:/etc/ssl/live/portal.host/
    - ${CABINET_SSL_FOLDER_PATH}:/etc/ssl/live/cabinet.host/
    - files:/var/www/diguni/files
    - edebo_files:/var/www/Files
    - backend_web:/var/www/diguni/backend/web
    - frontend_web:/var/www/diguni/frontend/web
  ports:
    - ${NGINX_HTTP_PORT_FORWARD}:80
    - ${NGINX_HTTPS_PORT_FORWARD}:443
  networks:
    - digital-university-public-network

watchtower:
  image: containrrr/watchtower
  container_name: watchtower
  restart: unless-stopped
  volumes:
    - /var/run/docker.sock:/var/run/docker.sock
  environment:
    WATCHTOWER_CLEANUP: "true"

```

		Цалко Я.В.			ІІЗ.КР.Б – 121 – 25 – ІІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		71

```

        WATCHTOWER_POLL_INTERVAL: "300"
        REPO_USER: ${REPO_USER}
        REPO_PASS: ${REPO_PASS}
    networks:
        - digital-university-private-network
networks:
    digital-university-private-network:
    digital-university-public-network:
volumes:
    files:
    edebo_files:
    backend_web:
    frontend_web:
    mysql_data:
    rabbitmq_data:
    mongo_data:
$groupName = Yii::$app->request->post('groupName');
$academicGroup = new AcademicGroup();
$edeboAcademicGroup = ActiveStudent::find()
    ->where(['group_name' => $groupName])
    ->orderBy(['education_date_begin' => SORT_ASC])
    ->one();

$academicGroup->name = $groupName;
$academicGroup->status = 1;
$academicGroup->degree_id = $edeboAcademicGroup-
>qualification_group_id;
$academicGroup->study_form_id = $edeboAcademicGroup-
>education_form_id;
$academicGroup->study_start = $edeboAcademicGroup-
>education_date_begin;
$academicGroup->study_end = $edeboAcademicGroup->education_date_end;
$academicGroup->course = $edeboAcademicGroup->course_id;

if ($edeboAcademicGroup->study_program_name) {
    $program = EdeboUniversityStudyPrograms::findOne([
        "speciality_name"          => $edeboAcademicGroup-
>full_speciality_name,
        "qualification_group_id" => $edeboAcademicGroup-
>qualification_group_id,
        "study_program_name"      => $edeboAcademicGroup-
>study_program_name,
    ]);
}

```

		Цалко Я.В.			ІІЗ.КР.Б – 121 – 25 – ІІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		72

```

}

$faculty =
EdeboUniversityStructures::findOne(["university_facultet_full_name"
=> $sedeboAcademicGroup->faculty_name]);

$specialtyId = $program->specialty_id;
$specialtyName = $sedeboAcademicGroup->full_specialty_name;
$studyProgramId = $sedeboAcademicGroup->study_program_name ? $program-
>university_study_program_id : null;
$facultyId = $faculty->id_university_facultet;

private function addStudentsToLocalGroup($groupName): void
{
    $students = EdeboActiveStudents::find()->where(['group_name' =>
$groupName])->all();
    $academicGroup = AcademicGroup::findOne(["name" => $groupName]);
    foreach ($students as $student) {
        $newStudent = new Student();
        $newStudent->education_id = $student->education_id;
        $newStudent->academic_group_id = $academicGroup->id;
        $newStudent->save(false);
    }
}

$model = new AcademicGroup();
$model->status = 1;
if ($model->loadAll(Yii::$app->request->post())) {
    if ($model->save(false)) {
        if ($model->specialty_id) {$model->specialty_name = $this-
>getSpecialtyNameBy($model->specialty_id);
        $model->save(false);
    }
    if (Yii::$app->request->isAjax) {
        Yii::$app->response->format = Response::FORMAT_JSON;
        return ['success' => true];
    }
}
}

```

		Цалко Я.В.			ІІЗ.КР.Б – 121 – 25 – ІІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		73

```

}
$groupInfo = $this->findModel($id);
$students = $this->getStudentsInGroup($groupInfo->id);
$studentsNoGroup = $this->getStudentsNoGroup($groupInfo);

$specialityName =
EdeboUniversityStudyPrograms::findOne(["speciality_name" =>
$groupInfo->speciality_name])->speciality_name;
$studyStatusesIds = [1, 10, 11, 12, 13, 14, 15, 17, 20, 21, 25, 30,
50, 61, 64, 70, 80, 81, 90];

$query = AcademicGroupOwnerService::getAllEdeboStudentsQuery([
    "$student.id",
    "$edeboStudents.education_id",
    "$student.academic_group_id",
    //...
]);

$query->andWhere(['or', ['!=', "$edeboStudents.group_name", new
\yii\db\Expression("$group.name")],
    ["$group.name" => null],
    ["$edeboStudents.group_name" => null],
    ["$edeboStudents.education_id" => null],
]);

->andWhere(["$student.academic_group_id" => null])
->andWhere(["$edeboStudents.history_type_id" => $studyStatusesIds])
->andWhere([
    'or',
    ["$edeboStudents.university_study_program_id" => $studyProgramId],
    ["$edeboStudents.university_study_program_id" => null]
]);

->andWhere(["$edeboStudents.education_form_id" => $studyFormId])
->andWhere(["$edeboStudents.full_speciality_name" =>
$specialityName])
$query
->andWhere("$group.course != $edeboStudents.course id")

```

		Цалко Я.В.			ІІЗ.КР.Б – 121 – 25 – ІІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		74

```

        ->andWhere(["$deboStudents.education_id" => $educationIds]);
    ;
    public function changeGroupForStudent(int $educationId, string|null
    $groupName, int $qualificationGroupId): bool
    {
        $url = $this->getUrl() . "/info-update/group/student/add";
        $data = [
            'educationId' => $educationId,
            'groupName' => $groupName,
            'qualificationGroupId' => $qualificationGroupId,
        ];
        return $this->sendPost($url, $data);
    }
    public function changeStudentCourse(int $educationId, int
    $qualificationGroupId, int $course): bool
    {
        $url = $this->getUrl() . "/info-update/group/student/set-course";
        $data = [
            'educationId' => $educationId,
            'qualificationGroupId' => $qualificationGroupId,
            'course' => $course
        ];
        return $this->sendPost($url, $data);
    }
    public function incrementGroupCourse(string $groupName, int
    $qualificationGroupId): mixed
    {
        $url = $this->getUrl() . "/info-update/group/increment-course";
        $data = [
            'groupName' => $groupName,
            'qualificationGroupId' => $qualificationGroupId,
        ];
        return $this->sendPostWithResponse($url, $data);
    }
    $query->joinWith('passportData', false)->distinct();
    public static function getPassportsOf(array $studentsIds, $fields =
    "*", $searchType = null): array

```

		Цалко Я.В.			ІІЗ.КР.Б – 121 – 25 – ІІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		75

```

{
    if (!$searchType) {
        $searchType = "Паспорт";
    }
    $query = self::getManyPersonDocumentsQuery($studentsIds, $fields)
        ->andWhere(['person_document_type_name' => $searchType]);
    return $query->all();
}

if ($this->hasSearchFields($searchFields)) {
    $query->joinWith(['requestTmp'])->distinct(true);
    //...
}

$countQuery = clone $query;
$countQuery->limit(-1)->offset(-1)->orderBy([]);
$model->status = 1;
$model->filename = $model->filename_unapproved;
$model->filename_unapproved = null;
$model->moderator_user_id = Yii::$app->user->identity->getId();
$model->save(false);
$model->status = 2;
if($model->filename){
    $model->filename_unapproved = $model->filename;
    $model->filename = null;
    $model->moderator_user_id = Yii::$app->user->identity->getId();
}
if (static::isDenearyExist($isActive)) {
    $deanery = self::getStudentDeanery($educationId, $isActive);
    $deaneryMark = self::getDeaneryMark($educationId);
    $data[] = [
        "name" => "Деканат",
        "value" => $deaneryMark->mark ?? null,
        "bypass_structure_id" => $deanery->id ?? null,
        "description" => $deaneryMark->description ?? null,
        'contacts' => $deanery->contact_details ?? null,
    ];
}

```

		Цалко Я.В.			ІІЗ.КР.Б – 121 – 25 – ІІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		76

```

];
}
$department = self::getGraduationDepartment($educationId, $isActive);
$departmentMark = self::getGraduationDepartmentMark($educationId);
protected static function findStudentStructureCathedralQuery(int
$educationId): ActiveQuery
{
    $query = BypassSheetUniversityStructure::find()
        ->innerJoin($universityStructure, "$universityStructure.id =
$bypassSheetUniversityStructure.university_settings_university_struct
ure_id")->innerJoin($cathedralStudyProgram,
"$cathedralStudyProgram.cathedral_id = $universityStructure.id")
        ->innerJoin($students, "$students.university_study_program_id
= $cathedralStudyProgram.study_program_id")
        ->andWhere(["$students.education_id" => $educationId]);
    return $query;
}
public static function getAllStructure(int $educationId, $isActive =
true): array
{
    $bypassSheetUniversityStructure =
BypassSheetUniversityStructure::tableName();
    $data = static::getAllStructureQuery($educationId, $isActive)-
>orderBy(["$bypassSheetUniversityStructure.position" => SORT_ASC])-
>asArray()->all();
    return $data;
}
return array_merge($data, self::getAllStructure($educationId,
$isActive));
foreach ($allRecords as $record) {
    $educationId = $record['education_id'];
    $l =
CircularLettersStudentService::getLetterForStudent($educationId);
    $studentLetters[$educationId] = $l;
}

```

		Цалко Я.В.			ІІЗ.КР.Б – 121 – 25 – ІІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		77

```

$sheetMark =
CircularLettersUniversityStructureService::findBypassSheetMarkOf($edu
cation_id, $bypass_structure_id);
if (is_null($sheetMark)) {
    $sheetMark = new BypassSheetMark();
    $sheetMark->student_education_id = $education_id;
    $sheetMark->bypass_sheet_university_structure_id =
$bypass_structure_id;
    if(!is_null($description)) {
        $sheetMark->mark = false;
    }
}
}

```

		Цалко Я.В.			ІІЗ.КР.Б – 121 – 25 – ІІЗ	Арк.
		Морозов А.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		78