

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВЕТЕРИНАРНОЇ
МЕДИЦИНИ ТА БІОТЕХНОЛОГІЙ ІМЕНІ С.З. ГЖИЦЬКОГО**

**ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ
ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

другого (магістерського) рівня вищої освіти

**на тему: “ Розробка та дослідження інформаційної системи
для підтримки освітнього процесу онлайн ”**

Виконав: ст. гр. ІТ-61

Спеціальності 126 – «Інформаційні системи та
технології»

(шифр і назва)

Гурський Володимир Васильович

(прізвище та ініціали)

Керівник: к.ф-м.н., в.о. доц., Чухрай Л.В.

(прізвище та ініціали)

Рецензент: _____

(прізвище та ініціали)

ДУБЛЯНИ-2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВЕТЕРИНАРНОЇ
МЕДИЦИНИ ТА БІОТЕХНОЛОГІЙ ІМ.С.З. ГЖИЦЬКОГО
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ

КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

другий (магістерський) рівень вищої освіти
ОС «Магістр» за спеціальністю – 126– «Інформаційні системи та технології»

“ЗАТВЕРДЖУЮ”

Завідувач кафедри _____

д.т.н., проф. А.М. Тригуба

“ ” _____ 2025р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Гурському Володимирі Васильовичу

Тема: «Розробка та дослідження інформаційної системи для підтримки освітнього процесу онлайн»

1 Керівник роботи к. ф-м н., в.о.доц., Чухрай Л.В.,

2 Затверджені наказом №140/к-с від 28.02.2025.

3 Строк подання студентом роботи 01.12.2025 р.

4 Початкові дані до роботи: конфігураційні файли Docker, Docker Compose та Dockerfile для компонентів Nginx, PHP-FPM і MySQL; налаштована база даних навчальних курсів, користувачів і результатів навчання; веб-інтерфейс адміністратора,

5 Зміст розрахунково-пояснювальної записки: ВСТУП, АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ, ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ПІДТРИМКИ ОСВІТНЬОГО ПРОЦЕСУ ОНЛАЙН, ТЕХНІЧНА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ, ТЕСТУВАННЯ ТА ОЦІНЮВАННЯ ПРАЦЕЗДАТНОСТІ ПРОГРАМНОГО ПРОДУКТУ,

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ,
ВИСНОВКИ, СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ, Додаток А

5. Перелік презентаційного матеріалу (з зазначенням обов'язкових елементів): _____

6. Консультанти з розділів:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1, 2, 3, 4, 5,6	<i>Чухрай Л.В., кандидат фізико-математичних наук, в.о., доцента кафедри інформаційних технологій</i>		
5	<i>Городецький І.М., доцент кафедри управління проектами та безпеки виробництва</i>		

7. Дата видачі завдання 28 лютого 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту	Строк виконання етапів роботи	Примітка
1.	<i>Написання першого розділу та означення головних завдань роботи</i>	<i>01.03-01.05.25</i>	
2.	<i>Виконання другого розділу та опис інформаційних технологій для виконання завдань роботи</i>	<i>01.03-01.05.25</i>	
3.	<i>Виконання третього розділу, методика вирішення завдань та елементи наукових досліджень</i>	<i>01.05-01.07.25</i>	
4.	<i>Виконання четвертого розділу, представлення результатів та їх узагальнення</i>	<i>01.05-01.07.25</i>	
5.	<i>Написання розділу: «Охорона праці та безпека в надзвичайних ситуаціях»</i>	<i>01.07-01.09.25</i>	
6.	<i>Завершення оформлення розрахунково-пояснювальної записки та презентаційних матеріалів</i>	<i>01.09-01.11.25</i>	
7.	<i>Завершення роботи в цілому</i>	<i>01.11-01.12.25</i>	

Студент _____ Гурський В.В.
(підпис) Керівник роботи _____ Чухрай Л.В.

УДК 004.9:37.018.43

Кваліфікаційна робота 58 сторінок, 28 рисунків, 4 таблиці, 1 дод., 22 джерел

Розробка та дослідження інформаційної системи для підтримки освітнього процесу онлайн, Гурський В.В., Кафедра ІТ-Дубляни, Львівський НУВМБ, 2025.

Метою кваліфікаційної роботи є розроблення та програмна реалізація інформаційного вузла системи дистанційного навчання на основі технологій контейнеризації та методу k -найближчих сусідів для забезпечення масштабованості, надійності та можливості відновлення параметрів інформаційних об'єктів у межах єдиного інформаційного простору.

Актуальність теми сучасний розвиток цифрових технологій та зростання потреби у гнучких формах навчання вимагають створення надійних і масштабованих систем дистанційної освіти. Традиційні підходи не забезпечують достатньої гнучкості для швидкого розгортання та обробки великих обсягів даних. Використання контейнеризації та методів інтелектуального аналізу, зокрема алгоритму k -найближчих сусідів, підвищує модульність, стійкість і функціональність освітніх систем. Тому дослідження архітектури вузла дистанційного навчання з можливістю відновлення параметрів інформаційних об'єктів є актуальним і відповідає сучасним вимогам цифрової освіти.

Об'єктом дослідження є процеси функціонування та взаємодії компонентів системи дистанційного навчання в умовах контейнеризованої інфраструктури.

Ключові слова: Docker, Docker-Compose, вузол системи, інформаційні об'єкти, відновлення параметрів.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	12
1.1. Дистанційна форма навчання: поняття та значення..	12
1.2. Форми організації дистанційного навчання: синхронна та асинхронна.....	16
1.3. Програмні платформи та інструменти для організації онлайн-навчання..	19
РОЗДІЛ 2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ПІДТРИМКИ ОСВІТНЬОГО ПРОЦЕСУ ОНЛАЙН... ..	24
2.1. Порівняння монолітної та мікросервісної архітектур при розробці системи дистанційного навчання.....	24
2.2. Технології контейнеризації	28
2.3. Технологія Docker у системі контейнеризації.....	30
РОЗДІЛ 3 ТЕХНІЧНА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ	33
3.1 Реалізація вузла системи дистанційного навчання.....	33
3.2. Розгортання середовища виконання системи Moodle.....	35
3.3. Інтеграція розгорнутого вузла Moodle у єдиний інформаційний простір системи.....	39
РОЗДІЛ 4 ТЕСТУВАННЯ ТА ОЦІНЮВАННЯ ПРАЦЕЗДАТНОСТІ ПРОГРАМНОГО ПРОДУКТУ.....	43
4.1. Перевірка відповідності серверного середовища.....	43
4.2. Налаштування конфігураційного файлу Moodle.....	45
4.3. Створення адміністратора систем.....	48
РОЗДІЛ 5 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ.....	51

5.1. Структурно-функціональний аналіз технологічного процесу.....	51
5.2. Розрахунок освітлення приміщення комп'ютерного кабінету.	52
5.3. Безпека в надзвичайних ситуаціях	55
ВИСНОВКИ.....	56
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	58
Додаток А.....	61

ВСТУП

Трендом сучасності стає забезпечення доступної та якісної освіти для кожного, незалежно від місця перебування чи часу. Розвиток інформаційних технологій сприяв виникненню та поширенню дистанційного навчання — форми освіти, у якій взаємодія між викладачем і студентом відбувається через цифрові платформи та мережу Інтернет. Події останніх років, зокрема пандемія COVID-19, прискорили цей процес і змусили освітні заклади у стислі терміни перейти на онлайн-формат, що виявило як переваги, так і проблеми організації дистанційної освіти. Відсутність єдиного інформаційного простору, розпорошеність навчальних матеріалів, різноманітність програмних платформ і нестача інструментів для моніторингу навчальної активності створюють труднощі як для студентів, так і для викладачів.

В умовах цифрової трансформації освіти постає потреба у створенні ефективних інформаційних систем, здатних об'єднати всі компоненти освітнього процесу — навчальні курси, засоби комунікації, аналітику успішності, тестування та зворотний зв'язок — у єдиному інтегрованому середовищі. Розробка таких систем дозволяє підвищити рівень організації онлайн-навчання, забезпечити персоналізацію освітнього процесу, контроль знань і гнучкість доступу до навчальних ресурсів.

Метою даної магістерської роботи є розробка інформаційної системи для підтримки освітнього процесу онлайн, яка забезпечить централізовану взаємодію між учасниками навчання, ефективне управління навчальними матеріалами та автоматизований моніторинг результатів студентів. Реалізація такого рішення сприятиме підвищенню якості освіти, доступності знань і створенню єдиного цифрового простору для навчання в сучасних умовах.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Дистанційна форма навчання: поняття та значення

Дистанційне навчання є сучасною формою організації освітнього процесу, яка здійснюється у синхронному (on-line) та асинхронному (off-line) режимах із використанням спеціалізованого науково-методичного, програмного та системно-технічного забезпечення. Такий формат навчання базується на застосуванні інформаційно-комунікаційних технологій, що забезпечують взаємодію між учасниками освітнього процесу незалежно від їхнього фізичного місця перебування. На сучасному етапі розвитку суспільства дистанційна форма освіти набуває особливої актуальності та відіграє важливу роль у наданні освітніх послуг широким верствам населення. Вона створює умови для реалізації конституційного права кожної людини на здобуття освіти незалежно від соціального статусу, матеріального забезпечення, територіального розташування, стану здоров'я, вікових особливостей, релігійних переконань та інших індивідуальних чинників, сприяючи формуванню рівного доступу до знань.

Активний розвиток науково-технічного прогресу, процеси цифровізації суспільства, а також зростання вимог до мобільності та гнучкості освітніх траєкторій зумовили становлення та подальший розвиток дистанційного навчання як самостійної і повноцінної форми освіти. Історичні витoki дистанційної освіти простежуються ще з 1840 року, коли у Великій Британії Ісаак Пітман започаткував навчання за допомогою поштового листування, використовуючи цей підхід для викладання стенографії. Надалі дана практика отримала поширення в Німеччині та інших європейських країнах, де почали з'являтися перші заочні та кореспондентські курси.

У середині ХХ століття дистанційна освіта зазнала суттєвих змін завдяки впровадженню засобів масової комунікації, зокрема радіо та телебачення, що дозволило охопити значно більшу аудиторію слухачів. Разом із тим такий формат навчання мав певні обмеження, пов'язані з відсутністю оперативного зворотного зв'язку між викладачем і здобувачем освіти. Подальший розвиток комп'ютерних технологій, поява персональних комп'ютерів і глобальної мережі Інтернет наприкінці ХХ — на початку ХХІ століття стали ключовими чинниками переходу дистанційного навчання до якісно нового рівня, забезпечивши інтерактивність, мультимедійність та можливість постійної комунікації в режимі реального часу.



Рисунок 1.1. . Дистанційне навчання через радіозв'язок у середині ХХ століття приклад ранніх форм використання технологій у освіті.

Вагомим етапом у становленні дистанційної освіти став 1969 рік, коли у Великій Британії було засновано Відкритий університет, діяльність якого була спрямована на розширення доступу до вищої освіти для широких верств населення. Досвід цього навчального закладу продемонстрував ефективність альтернативних форм навчання та став поштовхом для створення подібних

установ у різних країнах світу. У 1970–1980-х роках сформувалася мережа провідних закладів дистанційної освіти, серед яких Гагенський університет у Німеччині, Іспанський національний університет дистанційної освіти (UNED), INTEC, а також Національний технологічний університет США. Їх діяльність сприяла популяризації дистанційного навчання як повноцінної складової освітньої системи.

Подальший розвиток інформаційних технологій, зокрема поява персональних комп'ютерів та широке впровадження мережі Інтернет у 1990–2000-х роках, зумовили перехід дистанційної освіти до нового, сучасного етапу розвитку. Освітній процес почав активно використовувати вебінари, інтерактивні онлайн-курси, електронні навчальні платформи та системи керування навчанням, що забезпечило постійний зворотний зв'язок, інтерактивність і доступ до навчальних матеріалів у будь-який час.

В Україні становлення дистанційної освіти розпочалося у 2002 році, коли Міністерством освіти і науки було ініційовано експеримент із впровадження відповідної форми навчання. У 2004 році на нормативному рівні було затверджено перше «Положення про дистанційне навчання», що визначило основні принципи організації та реалізації освітнього процесу з використанням інформаційних технологій. Подальше вдосконалення нормативно-правової бази відбулося у 2013 році, коли було прийнято нову редакцію зазначеного документа.

Особливої значущості дистанційна форма здобуття освіти набула під час пандемії COVID-19, коли вона фактично стала основним інструментом забезпечення безперервності навчального процесу. У 2020 році було ухвалено оновлене «Положення про дистанційне навчання», яке надало можливість гнучкого поєднання очної та онлайн-форм освіти, а також широкого впровадження моделей змішаного навчання. Станом на сьогодні в Україні функціонує понад 200 закладів загальної середньої освіти з дистанційною формою навчання, серед яких перша ліцензована «Дистанційна школа “Оптіма”». Використання таких освітніх технологій дозволяє забезпечити

рівний доступ до знань, створити безпечні умови навчання в умовах надзвичайних ситуацій та врахувати індивідуальні освітні потреби здобувачів освіти.

Подальша еволюція дистанційного навчання нерозривно пов'язана з впровадженням інформаційних систем, що забезпечують централізоване управління освітнім процесом, зберігання та обробку навчальних матеріалів, а також організацію ефективної взаємодії між усіма учасниками навчання. Сучасні онлайн-платформи інтегрують функціональні можливості систем керування навчанням (LMS), інструментів комунікації та аналітичних модулів, що дозволяє не лише здійснювати передачу знань, але й контролювати якість освітнього процесу в режимі реального часу.

В умовах активної цифровізації освітньої сфери особливої актуальності набувають питання технічної надійності, продуктивності та масштабованості таких інформаційних систем. Збільшення кількості користувачів, обсягів навчального контенту та інтерактивних сервісів потребує застосування сучасних підходів до проектування програмного забезпечення, зокрема мікросервісної архітектури та технологій контейнеризації. Це створює передумови для переходу від використання окремих онлайн-інструментів до формування цілісних інформаційних вузлів, здатних забезпечити безперервність, гнучкість та адаптивність освітнього процесу.

Саме зазначені тенденції обґрунтовують необхідність розробки та впровадження інформаційних систем для підтримки освітнього процесу в онлайн-середовищі, які поєднують сучасні програмні платформи, веб-інтерфейси та контейнеризовані середовища розгортання, що і визначає актуальність теми даної кваліфікаційної роботи.

1.2 Форми організації дистанційного навчання: синхронна та асинхронна

Дистанційне навчання у сучасній освітній практиці реалізується у трьох базових формах: синхронній, асинхронній та гібридній (змішаній). Кожна з них має власні організаційні особливості, переваги та обмеження, що визначають доцільність їх застосування залежно від цілей навчання, технічних можливостей та індивідуальних потреб здобувачів освіти.

Синхронна форма дистанційного навчання передбачає одночасну участь усіх учасників освітнього процесу у навчальних заходах. У такому форматі взаємодія між викладачем і студентами здійснюється в режимі реального часу з використанням сучасних інформаційно-комунікаційних технологій, зокрема платформ для відеоконференцій, вебінарів, онлайн-лекцій і телеконференцій. Основною перевагою синхронного навчання є можливість оперативного зворотного зв'язку, безпосереднього спілкування, обговорення навчального матеріалу, виконання групових завдань та участі у дискусіях. Водночас цей формат характеризується меншою гнучкістю, оскільки вимагає присутності викладача і студентів у заздалегідь визначений час. Це може створювати труднощі для осіб з індивідуальним графіком, різними часовими поясами або обмеженими можливостями для постійної участі в онлайн-заняттях.

Асинхронна форма дистанційного навчання є більш гнучкою та орієнтованою на самостійну роботу здобувачів освіти. Вона не передбачає обов'язкової одночасної присутності викладача і студентів, а дозволяє навчатися у зручний час та власному темпі. У межах асинхронного навчання студенти отримують доступ до навчальних матеріалів, відеолекцій, презентацій, тестів і завдань із визначеними термінами виконання. Комунікація з викладачем та іншими учасниками навчального процесу здійснюється за допомогою електронної пошти, форумів, чатів або систем керування навчанням (LMS). Такий формат сприяє розвитку навичок самостійної роботи, відповідальності за власні результати навчання та глибшому осмисленню

навчального матеріалу. Для викладачів асинхронне навчання надає можливість заздалегідь підготувати якісний навчальний контент і організувати доступ до нього без прив'язки до конкретного часу проведення занять.

Гібридна або змішана форма дистанційного навчання поєднує елементи синхронного та асинхронного підходів, використовуючи переваги кожного з них. Такий формат передбачає комбінацію живого спілкування з викладачем (онлайн або очно) та самостійної роботи студентів із цифровими навчальними матеріалами. У процесі змішаного навчання активно застосовуються мультимедійні ресурси, інтерактивні освітні платформи, віртуальні лабораторії, електронні курси та інші інноваційні засоби навчання. Основною метою гібридного підходу є підвищення ефективності освітнього процесу, забезпечення його гнучкості та адаптації до індивідуальних потреб студентів. Змішане навчання дозволяє оптимально поєднати контроль з боку викладача з можливістю самостійного опрацювання матеріалу, що робить освітній процес більш результативним і доступним.



Рисунок 1.2. Форми реалізації дистанційного навчання: синхронна, асинхронна та гібридна.

Для реалізації потенціалу змішаного навчання широко використовуються цифрові інструменти, що забезпечують ефективну організацію освітнього процесу як в онлайн-, так і в офлайн-форматах. До таких засобів належать насамперед платформи для проведення відеоконференцій, зокрема Zoom, Google Meet та Microsoft Teams. Вони надають можливість організувати лекційні та практичні заняття, семінари й консультації у режимі реального часу, забезпечуючи безпосередню взаємодію між викладачем і здобувачами освіти. Функції спільного доступу до екрана, запису занять, використання віртуальних дошок і чатів значно розширюють дидактичні можливості традиційних форм навчання та сприяють підвищенню якості подання навчального матеріалу.

Важливе місце у змішаному навчанні займають месенджери та системи обміну повідомленнями, які забезпечують оперативну та безперервну комунікацію між усіма учасниками освітнього процесу. За їх допомогою здійснюється швидке інформування студентів, координація виконання навчальних завдань, надання індивідуального зворотного зв'язку та підтримка постійної взаємодії поза межами синхронних занять. Використання таких інструментів сприяє формуванню єдиного інформаційного освітнього середовища та підвищує доступність навчальних ресурсів у будь-який час.

Суттєву роль у реалізації змішаного навчання відіграють хмарні сервіси для зберігання даних і спільної роботи з документами, зокрема Google Drive, OneDrive та Dropbox. Вони забезпечують централізований доступ до навчальних матеріалів незалежно від місця перебування користувачів і використовуваних пристроїв. Застосування хмарних технологій спрощує поширення файлів, організацію групових проєктів, спільне редагування документів та контроль версій, що є особливо актуальним у межах командної та проєктно-орієнтованої діяльності.

Окрім цього, у змішаному навчанні активно використовуються інтерактивні веб-додатки для створення презентацій, інфографіки, візуалізацій і демонстрації навчального контенту. Такі засоби дозволяють підвищити

наочність подачі інформації, адаптувати матеріал до різних стилів сприйняття та забезпечити активне залучення студентів до навчального процесу. Інтерактивні елементи сприяють розвитку аналітичного мислення, креативності та навичок самостійної роботи.

Загалом використання цифрових технологій у межах змішаного навчання дає змогу поєднати переваги асинхронного опрацювання навчального матеріалу з ефективністю живого спілкування між викладачем і студентами. Такий підхід забезпечує високий рівень залученості здобувачів освіти, підвищує мотивацію до навчання та сприяє покращенню якості освітнього процесу в цілому, що робить змішане навчання одним із найбільш перспективних напрямів розвитку сучасної освіти.

1.3. Програмні платформи та інструменти для організації онлайн-навчання

Період пандемії COVID-19 став потужним каталізатором активного розвитку та масового впровадження цифрових засобів для організації дистанційного навчання. Особливе місце серед таких засобів зайняли платформи для проведення відеоконференцій, зокрема Zoom, Google Meet та Cisco Webex. Саме ці сервіси стали базовими інструментами забезпечення безпосередньої взаємодії між викладачами та студентами, що є одним із ключових чинників ефективності освітнього процесу. Їх використання дозволило навчальним закладам оперативно адаптуватися до нових умов та зберегти безперервність навчання в умовах обмежень на очне спілкування.

Завдяки функціональним можливостям платформ для відеоконференцій стало можливим проведення лекційних і практичних занять, семінарів, консультацій, захистів кваліфікаційних робіт, демонстрацій презентацій та організації колективних обговорень у режимі реального часу. Підтримка аудіо-та відеозв'язку, спільного доступу до екрана, чатів і запису занять забезпечила

збереження елементів традиційного навчального процесу навіть у віртуальному освітньому середовищі. Візуальний і слуховий контакт з викладачем, а також активна комунікація між студентами сприяли підтриманню навчальної дисципліни та залученості учасників освітнього процесу.

У сучасних системах дистанційної освіти сервіси відеоконференцій розглядаються як один із базових функціональних модулів, що забезпечує реалізацію синхронної форми навчання. Кожна з популярних платформ має власні особливості, переваги та обмеження, які впливають на вибір конкретного інструменту для освітніх потреб. Так, платформа Google Meet у безкоштовній версії підтримує проведення онлайн-зустрічей тривалістю до 60 хвилин із можливістю одночасної участі до 100 користувачів, тоді як Zoom обмежує тривалість безкоштовної конференції 40 хвилинами, але також дозволяє підключення до 100 учасників. Microsoft Teams та Cisco Webex орієнтовані на освітні й корпоративні середовища, пропонуючи розширені засоби керування заняттями, інтеграцію з навчальними матеріалами та підвищений рівень безпеки.

Таким чином, використання платформ для відеоконференцій стало невід'ємною складовою сучасного дистанційного навчання, забезпечивши реалізацію синхронної взаємодії та створивши умови для підтримки якості освітнього процесу в умовах цифрової трансформації освіти.

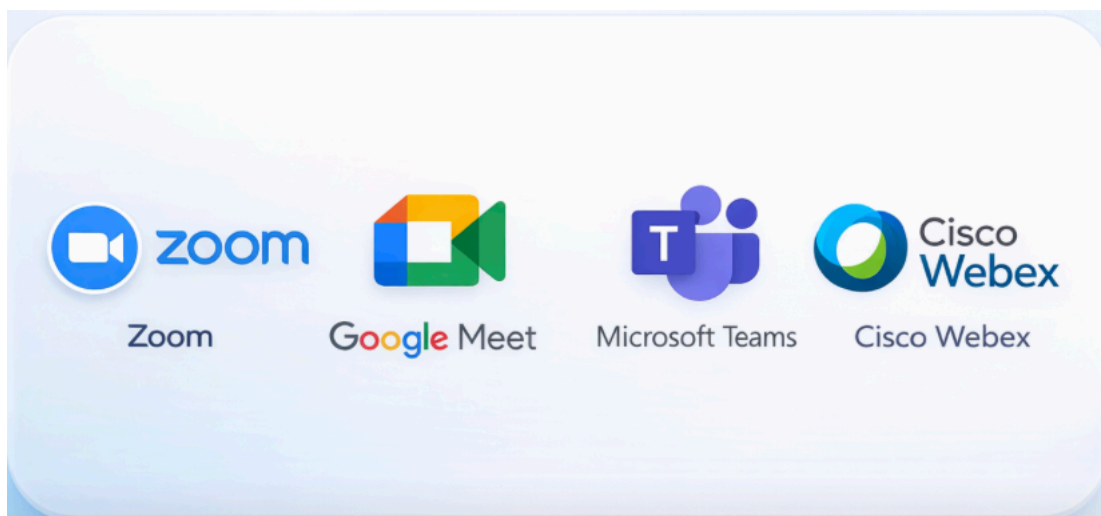


Рисунок. 1.3 Найпопулярніші сервіси відеоконференцій

Важливою складовою організації дистанційного та змішаного навчання є забезпечення спільного доступу до навчальних матеріалів для всіх учасників освітнього процесу. З цією метою широко застосовуються хмарні сервіси зберігання даних, зокрема Google Drive, Dropbox, NextCloud, які дозволяють централізовано розміщувати навчальні ресурси та надавати до них доступ відповідно до ролей користувачів. Також для обміну матеріалами активно використовуються електронна пошта та месенджери, що забезпечують швидку передачу файлів і підтримку оперативної комунікації між викладачами та студентами.

Контроль та оцінювання рівня засвоєння навчального матеріалу в умовах дистанційного навчання здійснюється із використанням як цифрових, так і традиційних підходів. Для автоматизованої перевірки знань часто застосовуються онлайн-інструменти, зокрема Google Forms, які дозволяють швидко створювати тестові завдання, налаштовувати типи запитань, обмеження за часом та автоматичне оцінювання результатів. Поряд із цим використовуються й класичні методи контролю знань, такі як виконання письмових робіт, рефератів або розрахункових завдань із подальшим надсиланням їх у вигляді сканованих копій або електронних файлів викладачеві для перевірки.

Поєднання різних способів поширення навчальних матеріалів і перевірки знань дозволяє забезпечити гнучкість освітнього процесу, адаптацію до технічних можливостей студентів та підтримку безперервності навчання в умовах дистанційної взаємодії.

	Microsoft OneDrive	Dropbox	Google Drive
FREE	15 GB	2 GB	15 GB
50 GB	x	x	x
100 GB	\$1.99	\$9.99	\$1.99
200 GB	\$3.99	\$19.99	x
500 GB	x	\$49.99	x
1 TB	\$6.99 w/ Office 365	\$75 for 5 users	\$9.99
10 TB	x	x	\$99.99
20 TB	x	x	\$199.99
30 TB	x	x	\$299.99

Рисунок. 1.4 Порівняння цін для популярних файлових сховищ

Провідні світові ІТ-компанії, зокрема Microsoft та Google, активно долучилися до цифрової трансформації освіти, розробивши спеціалізовані платформи для підтримки навчальних закладів у цифровому середовищі — Google Classroom та Microsoft Teams. Дані програмні рішення орієнтовані на комплексну організацію освітнього процесу та надають широкий набір інструментів для створення, розповсюдження й оцінювання навчальних завдань у електронному форматі. Їх використання значно спрощує адміністрування навчальної діяльності, підвищує керованість освітнього процесу та сприяє зростанню його загальної ефективності.

Платформа Google Classroom тісно інтегрована з хмарною екосистемою Google і використовує сервіси Google Drive, Docs, Sheets та Slides для зберігання й обробки навчальних матеріалів. Така інтеграція забезпечує зручний доступ до ресурсів, можливість спільної роботи над документами та оперативний обмін інформацією між викладачами і здобувачами освіти. Крім того, Google Classroom дозволяє автоматизувати процес розподілу завдань, встановлювати терміни їх виконання, відстежувати прогрес студентів і здійснювати оцінювання результатів навчальної діяльності в електронному вигляді.

Microsoft Teams, у свою чергу, поєднує засоби комунікації з офісними застосунками пакету Microsoft Office, зокрема Word, Excel, PowerPoint та іншими інструментами для роботи з документами. Це формує єдине інтегроване середовище для проведення онлайн-занять, організації групової та

проектної роботи, обміну файлами та управління навчальними командами. Завдяки поєднанню відеоконференцій, чатів, каналів спільної роботи й календарів Microsoft Teams забезпечує високий рівень взаємодії між усіма учасниками освітнього процесу та сприяє ефективній комунікації в межах навчальних курсів.

Обидві платформи відзначаються інтуїтивно зрозумілим користувацьким інтерфейсом, гнучкими механізмами керування доступом і розвиненими засобами комунікації, що стимулює активну участь студентів у навчанні. Їх впровадження дозволяє зменшити адміністративне навантаження на викладачів, автоматизувати рутинні процеси та забезпечити прозорість і об'єктивність оцінювання результатів навчання.

Застосування таких програмних рішень є необхідною умовою ефективного функціонування дистанційного та змішаного навчання в сучасних умовах. Вони забезпечують безперервність освітнього процесу, сприяють підвищенню якості засвоєння навчального матеріалу, підсилюють мотивацію студентів та дозволяють частково компенсувати обмеження очного навчання, зокрема в періоди карантинних заходів або інших надзвичайних ситуацій.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ПІДТРИМКИ ОСВІТНЬОГО ПРОЦЕСУ ОНЛАЙН

2.1. Порівняння монолітної та мікросервісної архітектур при розробці системи дистанційного навчання

Під час проєктування інформаційної системи для підтримки освітнього процесу в онлайн-середовищі одним із визначальних етапів є вибір архітектурного підходу. Найбільш поширеними моделями побудови сучасних програмних рішень є монолітна та мікросервісна архітектури. Від обраної архітектури залежать зручність розробки, можливості масштабування, простота супроводу та ефективність функціонування системи в умовах багатокористувацького доступу.

Монолітна архітектура

Монолітна архітектура є класичним підходом, за якого всі основні компоненти системи - користувацький інтерфейс, бізнес-логіка та база даних - реалізуються у межах одного програмного застосунку. Така архітектура відзначається простотою реалізації та розгортання, не потребує складних механізмів взаємодії між складовими й забезпечує високу швидкодію завдяки виконанню всіх операцій у межах одного середовища.

Переваги монолітної архітектури:

- простота розробки та розгортання програмного продукту;
- висока продуктивність за рахунок відсутності мережових викликів між компонентами;
- спрощений процес тестування, оскільки система функціонує як єдине ціле.

Недоліки монолітного підходу:

- обмежена гнучкість і складність внесення змін, оскільки навіть незначні модифікації потребують повторного розгортання всієї системи;

- неможливість незалежного масштабування окремих функціональних модулів;
- тісна залежність компонентів, що ускладнює підтримку та розвиток великих систем.

Мікросервісна архітектура

Мікросервісна архітектура ґрунтується на поділі системи на окремі незалежні сервіси, кожен з яких відповідає за виконання конкретної функції, наприклад перевірку знань, організацію відеоконференцій або обмін навчальними матеріалами. Взаємодія між сервісами реалізується за допомогою API, що забезпечує гнучкість і можливість масштабування.

Переваги мікросервісної архітектури для систем дистанційного навчання:

- автономність і масштабованість, що дозволяє розвивати окремі сервіси незалежно один від одного;
- свобода вибору технологій для реалізації кожного модуля;
- підвищена надійність, оскільки збій одного сервісу не зупиняє роботу всієї системи;
- можливість оновлення або заміни окремих компонентів без переривання роботи користувачів.

Недоліки мікросервісної архітектури:

- підвищена складність проєктування та налаштування взаємодії між сервісами;
- зростання вимог до інфраструктури та її обслуговування;
- необхідність впровадження централізованих механізмів моніторингу, журналювання та безпеки.

У межах проєкту створення інформаційної системи для підтримки освітнього процесу онлайн мікросервісна архітектура реалізована у вигляді окремих незалежних сервісів: сервісу контролю знань, сервісу відеоконференцій та сервісу розповсюдження навчальних матеріалів. Їх взаємодія здійснюється через API Gateway, який забезпечує захищений обмін

даними та централізовану автентифікацію користувачів. Узагальнену структурну модель розробленої системи подано на відповідному рисунку.

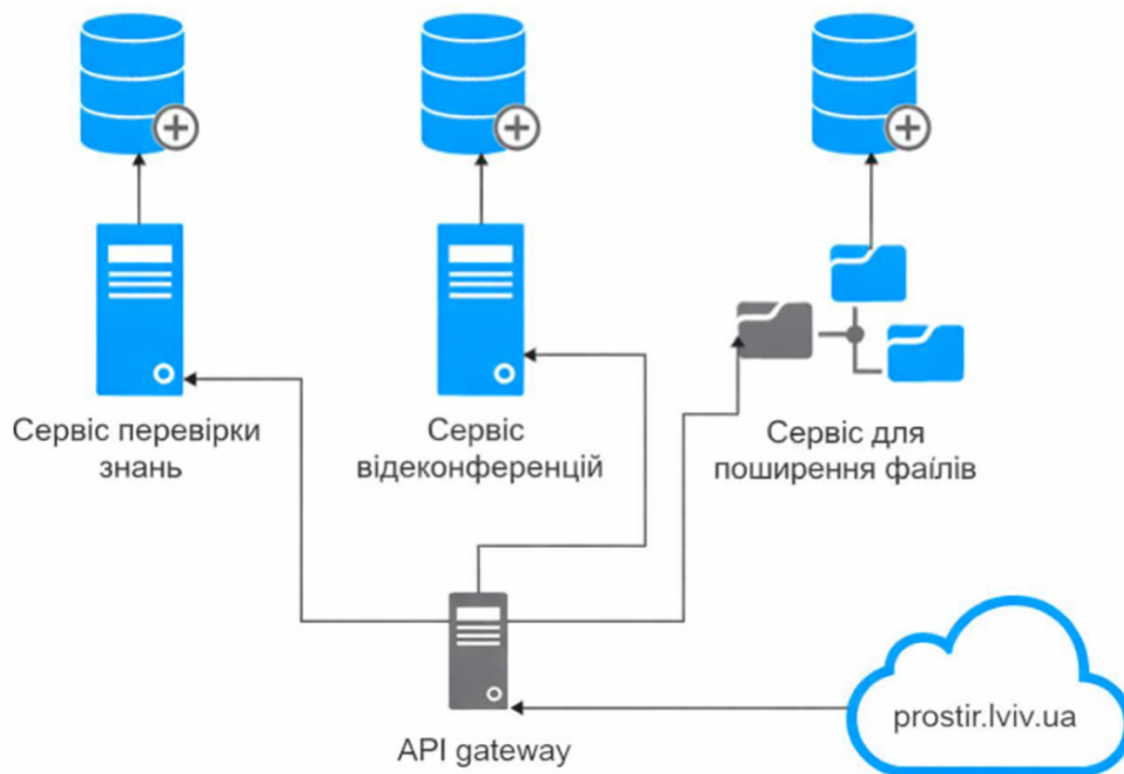


Рисунок 2.1. Структурна модель мікросервісної архітектури системи дистанційного навчання

Для систем дистанційного навчання, які передбачають поєднання кількох незалежних функціональних компонентів, зокрема сервісів контролю знань, організації відеоконференцій, зберігання та розповсюдження навчальних матеріалів, найбільш доцільним є використання мікросервісної архітектури. Запропонована структурна схема, наведена на рисунку 2.1, ілюструє поділ платформи на окремі автономні сервіси, кожен з яких реалізує визначений набір функцій і використовує власне сховище даних.

Центральним елементом такої архітектури виступає API Gateway, який забезпечує єдину точку доступу до системи. Через нього здійснюється обробка та маршрутизація запитів користувачів до відповідних сервісів, централізована автентифікація, керування правами доступу, а також моніторинг стану

компонентів. Використання такого підходу спрощує взаємодію клієнтської частини з серверною інфраструктурою та підвищує рівень захищеності обміну інформацією.

Запровадження мікросервісної архітектури забезпечує високий ступінь модульності інформаційної системи, що дозволяє виконувати розробку, тестування й оновлення окремих сервісів незалежно від інших компонентів платформи. Це є особливо актуальним для систем дистанційного навчання, які працюють у багатокористувацькому режимі та потребують регулярного розширення функціональних можливостей.

Окрім цього, мікросервісний підхід створює умови для гнучкого масштабування системи відповідно до зростання кількості користувачів або навантаження на окремі підсистеми. Наприклад, сервіс відеоконференцій може бути масштабований автономно, без впливу на роботу сервісів контролю знань чи файлового сховища, що дозволяє більш ефективно використовувати обчислювальні ресурси.

Таким чином, застосування мікросервісної архітектури в системах дистанційного навчання забезпечує гнучкість, масштабованість і надійність платформи, а також формує основу для подальшого розширення функціоналу й інтеграції нових сервісів у межах єдиного інформаційного освітнього простору.

2.2. Технології контейнеризації

Для розгортання мікросервісної архітектури в роботі застосовано технології контейнеризації, зокрема Docker і Kubernetes, які забезпечують ізоляцію програмних компонентів, їхню переносимість між різними середовищами та можливість гнучкого масштабування. Контейнеризація є сучасним підходом до розгортання програмних систем і дозволяє досягти стабільної та передбачуваної роботи застосунків незалежно від конфігурації апаратного або програмного середовища. На відміну від класичної віртуалізації, за якої кожен застосунок функціонує у власній гостьовій операційній системі, контейнерні технології передбачають спільне використання ядра базової ОС, зберігаючи при цьому ізоляцію бібліотек, залежностей і середовища виконання.

Застосування контейнеризації суттєво знижує навантаження на апаратні ресурси, скорочує час запуску сервісів і спрощує процес оновлення програмних компонентів. Для мікросервісної архітектури системи дистанційного навчання такий підхід є особливо доцільним, оскільки дозволяє ізолювати окремі сервіси, зокрема модулі перевірки знань, організації відеоконференцій та поширення навчальних матеріалів. Це забезпечує можливість незалежного оновлення, масштабування та супроводження кожного компонента без негативного впливу на роботу всієї платформи.

Серед найбільш поширених інструментів контейнеризації ключову роль відіграє Docker, який використовується для створення, налаштування та керування контейнерами, а також Kubernetes, призначений для оркестрації контейнерів, автоматичного масштабування сервісів і ефективного розподілу навантаження між ними.

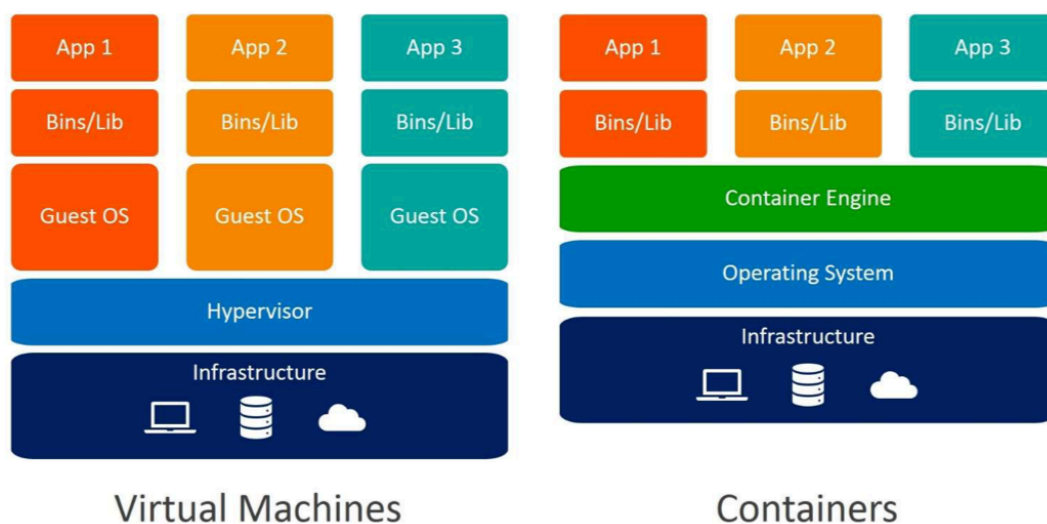


Рисунок 2.2. Порівняння віртуалізації та контейнеризації.

Порівняльний аналіз віртуалізації та контейнеризації, поданий на рисунку 2.2, демонструє ключові відмінності між цими підходами до розгортання програмних систем. У разі використання віртуальних машин кожен застосунок функціонує в межах окремої гостьової операційної системи, що призводить до підвищеного споживання апаратних ресурсів і збільшення часу запуску. Керування такими середовищами здійснюється за допомогою гіпервізора, який відповідає за розподіл обчислювальних ресурсів між окремими віртуальними машинами.

Контейнеризація, на відміну від віртуалізації, базується на спільному використанні ядра базової операційної системи, тоді як кожен контейнер включає лише необхідний набір бібліотек і залежностей для виконання конкретного застосунку. Це дозволяє істотно зменшити накладні витрати, підвищити ефективність використання серверних ресурсів і забезпечити швидкий запуск контейнерів. У результаті контейнеризовані середовища є більш легкими, продуктивними та економічними порівняно з традиційними віртуальними машинами.

Контейнерний рушій, зокрема Docker Engine, відповідає за ізоляцію процесів, керування життєвим циклом контейнерів і уніфікацію середовища виконання. Такий підхід є особливо доцільним у мікросервісних архітектурах,

де кожен сервіс розгортається в окремому контейнері та може масштабуватися незалежно відповідно до поточного навантаження.

Для інформаційних систем дистанційного навчання застосування контейнерних технологій має низку вагомих переваг. Контейнеризація забезпечує швидке розгортання нових сервісів, спрощує процес оновлення компонентів без необхідності зупинки всієї системи та гарантує стабільну роботу платформи за умов високої кількості одночасних користувачів. Окрім цього, використання контейнерів полегшує перенесення системи між різними серверами або хмарними інфраструктурами, що підвищує гнучкість і надійність її експлуатації.

Таким чином, результати порівняння віртуалізації та контейнеризації підтверджують доцільність застосування контейнерних технологій як бази для побудови сучасної мікросервісної інформаційної системи дистанційного навчання, орієнтованої на масштабованість, ефективне використання ресурсів і високу стійкість до навантажень.

2.3. Технологія Docker у системі контейнеризації

Docker - це сучасна платформа для створення, розгортання та керування програмним забезпеченням у контейнерах. Хоча ідея контейнерів існувала ще в операційних системах Linux та Unix, саме Docker -проект із відкритим вихідним кодом, започаткований у 2013 році, - зробив цю технологію популярною та доступною для широкого кола розробників. Його головна концепція - *“одного разу зібрати застосунок і запускати його будь-де”* -стала фундаментом сучасної DevOps-культури.

Історія Docker почалася у 2008 році, коли Соломон Хайкс у Парижі заснував компанію DotCloud, яка спочатку надавала платформу як послугу (PaaS). У 2013 році команда змістила фокус із платформи на основну технологію контейнерів, на якій вона була побудована. Під час конференції

PyCon того ж року Хайкс представив першу версію Docker, після чого інструмент швидко здобув популярність серед спільноти розробників і провідних IT-корпорацій - Microsoft, IBM, Red Hat тощо. Це стало початком «контейнерної революції» в розробці програмного забезпечення.

Docker зробив процес роботи з контейнерами простим і зрозумілим. Його екосистема включає кілька основних компонентів:

- Dockerfile - текстовий файл, який описує, як створити контейнер, які бібліотеки, середовище, змінні та порти використовуються.
- Docker Image - базовий образ контейнера, що містить усі необхідні залежності та конфігурації.
- Docker Engine - ядро платформи, яке створює та запускає контейнери. Воно функціонує за принципом клієнт-серверної взаємодії й керує процесами через API та інтерфейс командного рядка.
- Docker Hub - централізоване сховище для збереження, обміну й завантаження образів контейнерів, аналогічне до GitHub, але для контейнерних середовищ.
- Docker Compose - інструмент, який дозволяє описати й запустити багатоконтейнерні програми, використовуючи конфігураційні файли у форматі YAML.

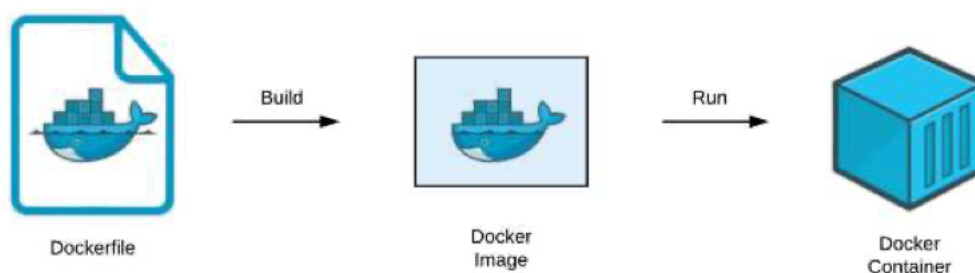


Рисунок 2.3 Воркфлоу контейнеризації за допомогою Docker

На рисунку 2.3 представлено типовий воркфлоу контейнеризації із використанням платформи Docker, який складається з трьох основних етапів: створення Dockerfile, побудови образу (Docker Image) та запуску контейнера

(Docker Container). Dockerfile є текстовим конфігураційним файлом, у якому описуються базовий образ, необхідні бібліотеки, залежності, змінні середовища та команди для підготовки середовища виконання застосунку. На основі цього файлу формується Docker-образ - стандартизований шаблон, що містить усі компоненти, необхідні для коректної роботи програмного забезпечення.

Після створення образу він може бути багаторазово використаний для запуску контейнерів у різних середовищах. Контейнер є ізольованим середовищем виконання, у якому застосунок працює незалежно від конфігурації хостової операційної системи. Такий підхід забезпечує відтворюваність результатів, зменшує ризик помилок під час розгортання та значно спрощує процес супроводження програмних систем.

Контейнери Docker вирізняються компактністю, швидкістю запуску та високою портативністю. Вони дозволяють уникнути конфліктів між різними версіями бібліотек або налаштувань, що є особливо важливим у багатокомпонентних системах. Завдяки цьому забезпечується стабільна робота застосунків незалежно від апаратної платформи або серверного середовища.

Для системи дистанційного навчання, побудованої на основі мікросервісної архітектури, використання Docker є доцільним і ефективним рішенням. Контейнеризація спрощує розгортання окремих сервісів, таких як відеоконференції, перевірка знань та обмін навчальними матеріалами, знижує витрати обчислювальних ресурсів, підвищує гнучкість системи та прискорює процес оновлення функціоналу. Саме завдяки цим перевагам Docker став стандартом у розробці сучасних вебзастосунків і освітніх платформ.

РОЗДІЛ 3

ТЕХНІЧНА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ

3.1 Реалізація вузла системи дистанційного навчання

Для створення сучасної системи дистанційного навчання необхідно забезпечити її гнучкість, масштабованість і здатність до інтеграції з різними освітніми установами. З цією метою було розроблено архітектуру єдиного інформаційного простору, яка передбачає взаємодію окремих вузлів навчальних закладів у межах спільної платформи. Кожен структурний елемент системи функціонує на основі власної інфраструктури, але водночас є складовою частиною єдиної мережі, що об'єднує всіх користувачів і забезпечує централізовані механізми автентифікації, управління та моніторингу. Узагальнену архітектурну схему запропонованого рішення подано на рисунку 3.9.

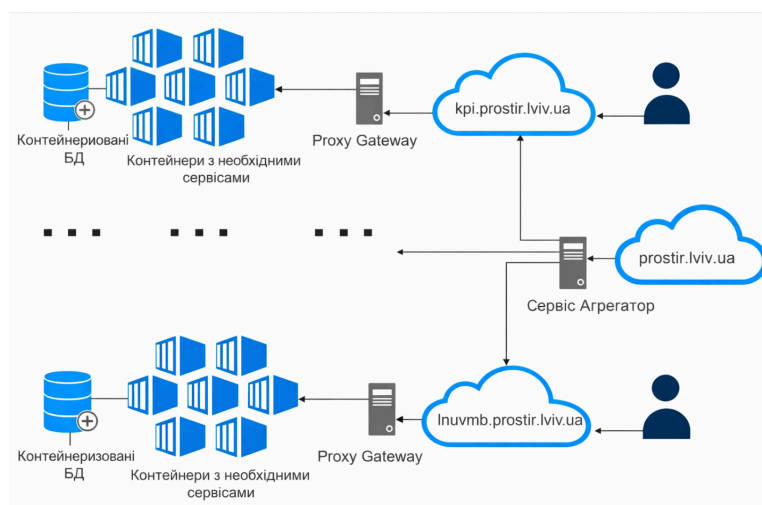


Рисунок 3.1. Архітектура єдиного інформаційного простору системи
дистанційного навчання

Розглянута архітектура представляє модель регіонального масштабу, що побудована за принципами мікросервісної організації та контейнеризації. У її структурі реалізовано багаторівневий підхід, який дозволяє об'єднати у спільне інформаційне середовище навчальні заклади різних рівнів під одним доменом - prostir.kiev.ua. Цей домен виступає центральною точкою доступу, що вказує

на сервіс-агрегатор - головний вебпортал системи, на якому розміщено посилання на всі підрозділи, підключені до спільного освітнього простору.

Кожен навчальний заклад має власний піддомен, наприклад `kpi.prostir.kiev.ua` або `ntu.prostir.kiev.ua`, який відповідає окремому інстансу системи з унікальною конфігурацією сервісів. Такий підхід забезпечує автономність та гнучкість - кожен вузол може самостійно визначати, які модулі системи дистанційного навчання активувати (відеоконференції, тести, файлове сховище, система обліку користувачів тощо), а які залишити неактивними.

Усередині кожного підрозділу реалізовано набір контейнерів, у яких розгорнуті необхідні сервіси: вебсервер, база даних, система керування навчанням (LMS) та допоміжні компоненти. Контейнери взаємодіють через Proxy Gateway, що забезпечує маршрутизацію запитів користувачів та зв'язок із сервісами зберігання даних.

Сервіс-агрегатор виступає посередником між користувачами та окремими вузлами системи, здійснюючи централізовану авторизацію, облік користувачів і моніторинг активності. Завдяки цьому користувач може отримати доступ до свого навчального закладу через єдиний портал, не змінюючи середовища взаємодії.

В основі архітектури лежить принцип контейнеризації: усі сервіси ізольовані в контейнерах, що дає змогу легко масштабувати систему, переносити її між серверами, швидко оновлювати та забезпечувати стабільну роботу навіть при великій кількості користувачів. Для розгортання контейнерів використовується Docker, а для їх оркестрації може застосовуватись Docker Compose або Kubernetes.

Завдяки такій архітектурі створюється єдиний інформаційний простір із можливістю децентралізованого управління, високою надійністю та зручністю інтеграції нових сервісів у майбутньому. Це рішення дозволяє забезпечити безперервність освітнього процесу, незалежно від кількості закладів-учасників, навантаження чи технічних змін на окремих вузлах системи.

3.2. Розгортання середовища виконання системи Moodle

Застосування технологій контейнеризації під час розгортання системи дозволяє забезпечити раціональне використання обчислювальних ресурсів і автоматизоване керування окремими вузлами системи дистанційного навчання. Для реалізації вузла інформаційної системи було використано контейнерний підхід, який гарантує ізоляцію компонентів, спрощує процес масштабування та забезпечує швидке й зручне розгортання програмного середовища.

Оскільки платформа Moodle реалізована з використанням мови програмування PHP, початковим етапом розгортання стало створення окремого контейнера з PHP-FPM, який відповідає за обробку та виконання PHP-скриптів. Для цього було використано базовий образ із репозиторію DockerHub, доповнений необхідними бібліотеками, розширеннями та модулями, потрібними для коректної роботи системи.

```
ARG LARADOCK_PHP_VERSION
ARG BASE_IMAGE_TAG_PREFIX=latest
FROM laradock/php-fpm:${BASE_IMAGE_TAG_PREFIX}-${LARADOCK_PHP_VERSION}
```

Рисунок 3.2 – Налаштування контейнера PHP-FPM

Наступним етапом стало створення контейнера з вебсервером Nginx, який виконує обробку вхідних запитів користувачів, здійснює розподіл навантаження та передає динамічні запити до середовища виконання PHP-FPM. Завдяки високій продуктивності, надійності та ефективній роботі з паралельними з'єднаннями вебсервер Nginx є доцільним вибором для забезпечення стабільного функціонування вузла системи дистанційного навчання.

```
#####
# XSL:
#####

ARG INSTALL_XSL=false

RUN if [ ${INSTALL_XSL} = true ]; then \
  # Install the xsl extension
  apt-get -y install libxslt-dev && \
  docker-php-ext-install xsl \
;fi
```

Рисунок 3.3 – Фрагмент Dockerfile для створення контейнера Nginx

З метою забезпечення коректної взаємодії між усіма компонентами системи було розроблено власний конфігураційний файл, у якому визначено правила обробки запитів, маршрутизацію трафіку та обслуговування статичних ресурсів. Вебсервер налаштовано на роботу через 80 порт із використанням локального доменного імені `kpi.prostir.loc`, при цьому каталог `/var/www/moodle` визначено як кореневу директорію вебзастосунку.

```
FROM nginx:alpine

COPY nginx.conf /etc/nginx/

RUN apk add --no-cache curl

ARG PHP_UPSTREAM_CONTAINER=php-fpm
```

Рисунок 3.4 – Конфігураційний файл Nginx для Moodle

```
ARG PHP_UPSTREAM_PORT=9000

# Set upstream conf and remove the default conf
RUN echo "upstream php-upstream { server
${PHP_UPSTREAM_CONTAINER}:${PHP_UPSTREAM_PORT}; }" >
/etc/nginx/conf.d/upstream.conf \
  && rm /etc/nginx/conf.d/default.conf

ADD ./startup.sh /opt/startup.sh
RUN sed -i 's/\r//g' /opt/startup.sh
CMD ["/bin/bash", "/opt/startup.sh"]

EXPOSE 80 81 443
```

3.5. Налаштування сховища даних та інтеграція компонентів

Для зберігання навчальних матеріалів, даних користувачів, результатів оцінювання та структури курсів у системі було використано систему керування базами даних MySQL. Дана СУБД забезпечує надійне збереження інформації, високошвидкісний доступ до даних і підтримку багатокористувацької роботи, що є важливою умовою стабільного функціонування системи дистанційного навчання.

```
server {
    listen 80 default_server;
    listen [::]:80 default_server ipv6only=on;

    server_name kpi.prostir.loc;
    root /var/www/moodle;
    index index.php index.html index.htm;

    location / {
        try_files $uri $uri/ /index.php$is_args$args;
    }

    location ~ [^/]\.php(/|$) {
        fastcgi_split_path_info ^(.+\.(php|php5))(/.+)$;
        fastcgi_index index.php;
        fastcgi_pass php-upstream;
        include fastcgi_params;
        fastcgi_param PATH_INFO $fastcgi_path_info;
        fastcgi_param SCRIPT_FILENAME
            $document_root$fastcgi_script_name;
    }

    location /dataroot/ {
        internal;
        alias /var/www/moodldata/; # ensure the path ends with /
    }
}
```

Рисунок 3.6 – Dockerfile контейнера MySQL

Для розгортання контейнера бази даних було використано офіційний образ MySQL, який додатково налаштовано за допомогою власного конфігураційного файлу `my.cnf`, що визначає параметри доступу, кодування даних і рівень безпеки. Усі компоненти системи — контейнери PHP-FPM, Nginx та MySQL — об'єднано в єдину віртуальну мережу Docker, яка забезпечує їхню коректну взаємодію та цілісне функціонування всієї системи.

```

ARG MYSQL_VERSION
FROM mysql:${MYSQL_VERSION}
COPY my.cnf /etc/mysql/conf.d/my.cnf

RUN chmod 0444 /etc/mysql/conf.d/my.cnf

CMD ["mysqld"]

EXPOSE 3306

```

Рисунок 3.7 – Dockerfile контейнера MySQL



Structure of container interaction in a Moodle system

Рисунок 3.8– Структура взаємодії контейнерів системи Moodle

Ця структура відображає логіку взаємодії між основними компонентами вузла системи дистанційного навчання Moodle, реалізованого на основі контейнеризації. Кожен елемент виконує власну функцію в загальній архітектурі системи.

Контейнер PHP-FPM забезпечує виконання серверної логіки системи Moodle, обробляючи PHP-скрипти та запити, які надходять від вебсервера. Контейнер Nginx виконує роль проміжного шару між користувачем і додатком - він приймає HTTP-запити, керує маршрутизацією, обслуговує статичні ресурси (зображення, стилі, скрипти) та передає динамічні запити до PHP-FPM. Контейнер MySQL відповідає за збереження та обробку даних — навчальних курсів, користувацьких профілів, результатів тестувань та іншої інформації освітнього процесу.

Усі три контейнери взаємодіють у межах спільної Docker-мережі, що гарантує стабільну комунікацію між компонентами без потреби у зовнішніх з'єднаннях. Така структура забезпечує модульність, ізоляцію процесів і високу

масштабованість системи — кожен компонент може бути оновлений або розгорнутий незалежно, без впливу на інші елементи.

Отже, сформоване контейнеризоване середовище забезпечує можливість оперативного розгортання окремих вузлів системи дистанційного навчання, їх гнучке масштабування відповідно до рівня навантаження та стабільну роботу навіть за значної кількості одночасних користувачів. Розгортання системи за допомогою Docker-контейнерів забезпечує високу адаптивність, стабільність роботи та спрощує подальше супроводження і оновлення інформаційної системи дистанційного навчання

3.3. Інтеграція розгорнутого вузла Moodle у єдиний інформаційний простір системи

Після побудови та конфігурації контейнерів PHP-FPM, Nginx та MySQL наступним етапом є розгортання прикладного середовища Moodle. Для запуску мультимодульної інфраструктури використовується утиліта Docker Compose, яка дозволяє одночасно підняти всі необхідні сервіси та створити спільну мережу для їх взаємодії.

Запуск контейнерів здійснюється командою:

```
docker-compose up -d
```

У результаті система автоматично створює необхідні мережі та томи зберігання, після чого ініціалізує всі контейнери, що належать до проєкту Moodle.

```

PS D:\IASA\2semestr\Diploma\node\laradock> docker-compose up -d nginx mysql adminer
Creating network "moodle_frontend" with driver "bridge"
Creating network "moodle_backend" with driver "bridge"
Creating network "moodle_default" with the default driver
Creating volume "moodle_mysql" with local driver
Creating volume "moodle_percona" with local driver
Creating volume "moodle_mssql" with local driver
Creating volume "moodle_postgres" with local driver
Creating volume "moodle_memcached" with local driver
Creating volume "moodle_redis" with local driver
Creating volume "moodle_neo4j" with local driver
Creating volume "moodle_mariadb" with local driver
Creating volume "moodle_mongo" with local driver
Creating volume "moodle_minio" with local driver
Creating volume "moodle_rethinkdb" with local driver
Creating volume "moodle_phpmyadmin" with local driver
Creating volume "moodle_adminer" with local driver
Creating volume "moodle_aerospike" with local driver
Creating volume "moodle_caddy" with local driver
Creating volume "moodle_meilisearch" with local driver
Creating volume "moodle_elasticsearch" with local driver
Creating volume "moodle_mosquitto" with local driver
Creating volume "moodle_confluence" with local driver
Creating volume "moodle_sonarqube" with local driver
Creating volume "moodle_cassandra" with local driver
Creating volume "moodle_graylog" with local driver
Creating volume "moodle_docker-in-docker" with local driver
Creating volume "moodle_react" with local driver
Creating moodle_docker-in-docker_1 ... done
Creating moodle_mysql_1 ... done
Creating moodle_workspace_1 ... done
Creating moodle_php-fpm_1 ... done
Creating moodle_adminer_1 ... done
Creating moodle_nginx_1 ... done
PS D:\IASA\2semestr\Diploma\node\laradock>

```

Рисунок 3.9– Результат підняття контейнерів

Після ініціалізації контейнерів формується віртуальне інфраструктурне середовище, у межах якого створюються мережі типу *bridge*, окремі файлові томи для кожного сервісу та ізольовані середовища виконання. Контроль за станом роботи контейнерів, перегляд журналів подій і використання ресурсів здійснюється за допомогою графічного інтерфейсу Docker Desktop.

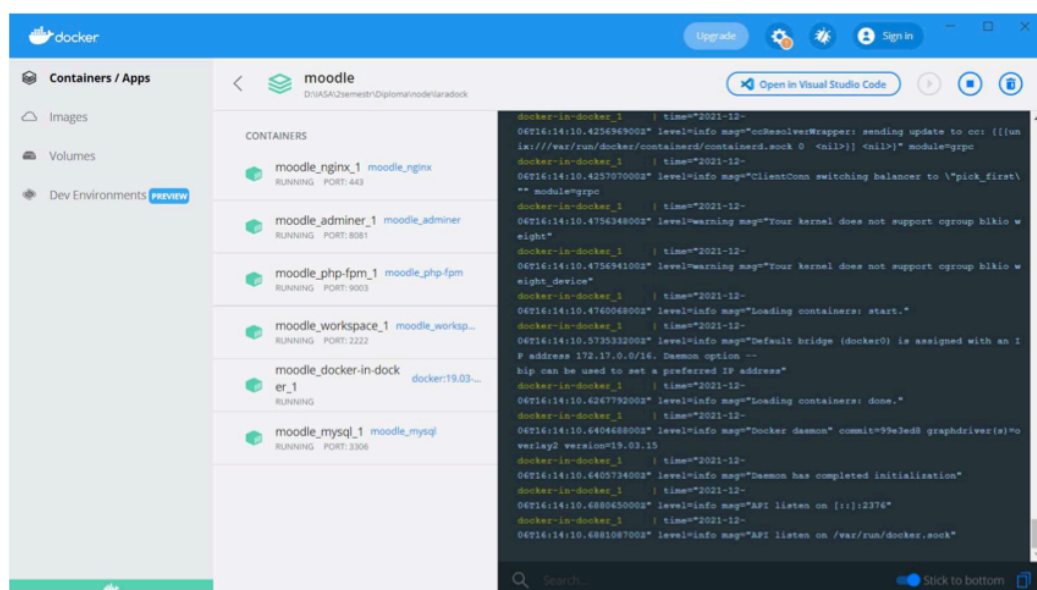
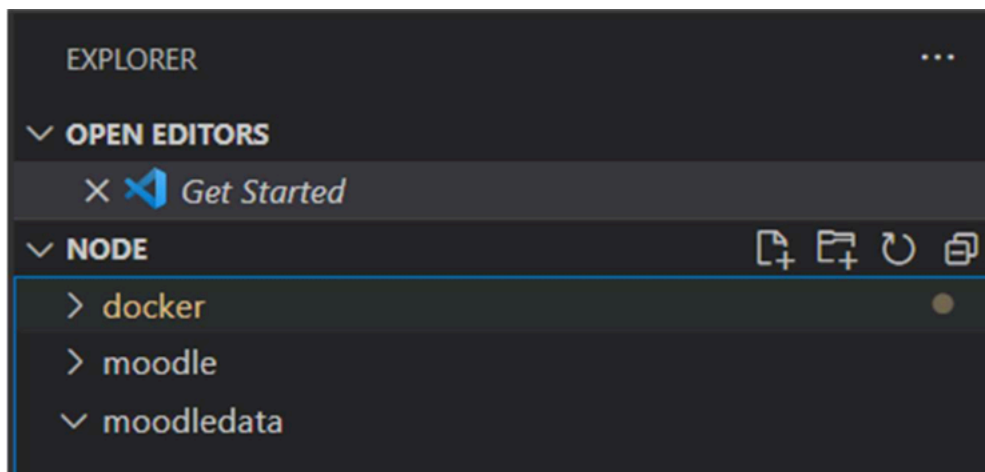


Рисунок 3.10 – Користувацький інтерфейс Docker з переліком контейнерів

Після успішного розгортання інфраструктури виконується підготовка файлової структури платформи Moodle. На початковому етапі здійснюється завантаження вихідних файлів із GitHub-репозиторію, після чого створюється каталог *moodledata*, призначений для зберігання кешу, користувацьких файлів та інших службових ресурсів системи.



3.11 Структура каталогів проєкту (docker, moodle, moodledata)

Наступним кроком є перевірка доступності системи за локальною веб-адресою, наприклад <http://kpi.prostir.loc/>. У разі коректного налаштування середовища при переході за вказаною адресою відображається початковий екран інсталятора платформи Moodle.

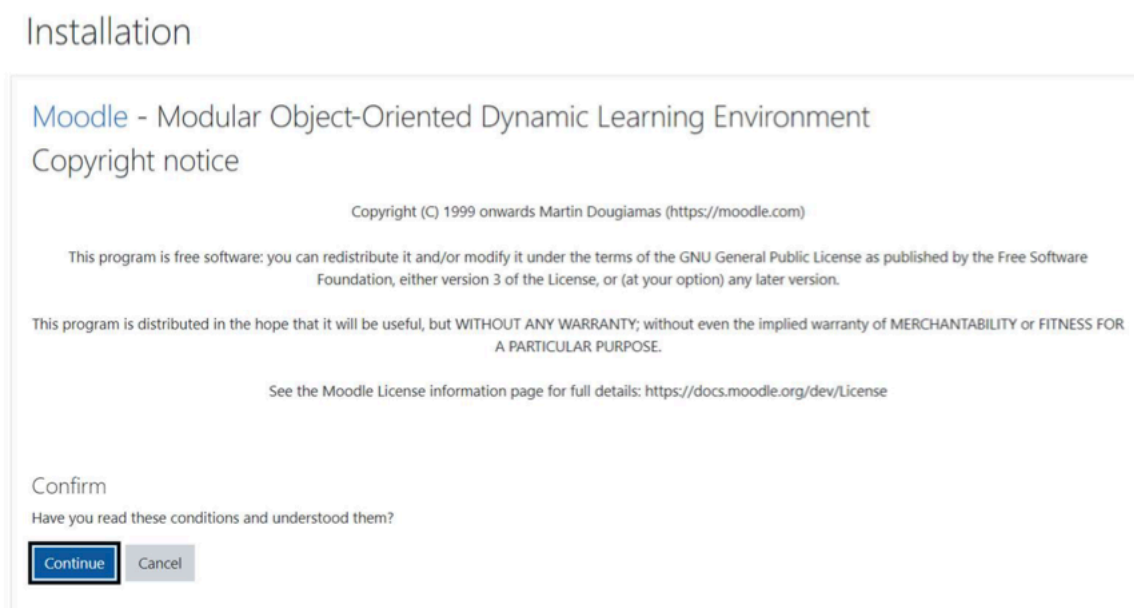


Рисунок 3.12 – Початковий екран встановлення Moodle

На початковому етапі встановлення платформи Moodle виконується перевірка готовності програмного середовища до подальшої роботи системи. Зокрема, здійснюється тестування коректності взаємодії між вебсервером, середовищем виконання PHP та контейнером бази даних, що дає змогу виявити можливі помилки конфігурації ще до початку формування структури даних. Такий підхід дозволяє мінімізувати ризики некоректної інсталяції та забезпечує стабільність подальшої експлуатації системи дистанційного навчання.

Застосування контейнеризованого середовища на цьому етапі має суттєву практичну перевагу, оскільки всі компоненти платформи функціонують у стандартизованому та ізольованому середовищі. Це забезпечує однакову поведінку інсталятора незалежно від апаратного забезпечення або операційної системи сервера. У разі необхідності процес встановлення може бути повторений або перенесений на інший сервер без виконання додаткових ручних налаштувань.

Таким чином, стартовий екран інсталяції Moodle виконує не лише ознайомчу, а й контрольну функцію, підтверджуючи готовність контейнеризованої інфраструктури до подальшого розгортання платформи. Успішне проходження цього етапу є необхідною умовою для коректного створення бази даних, налаштування облікових записів користувачів, навчальних курсів та інших функціональних компонентів системи дистанційного навчання.

РОЗДІЛ 4 ТЕСТУВАННЯ ТА ОЦІНЮВАННЯ ПРАЦЕЗДАТНОСТІ ПРОГРАМНОГО ПРОДУКТУ

4.1. Перевірка відповідності серверного середовища

Метою даного розділу є оцінювання працездатності розгорнутого середовища Moodle, аналіз функціонування всіх його компонентів і підтвердження правильності виконаних налаштувань системи. Процес тестування охоплює перевірку відповідності системним вимогам, коректності підключення до бази даних, а також створення облікового запису адміністратора, навчальних курсів і користувачів.

Після запуску інсталлятора платформи Moodle автоматично виконується діагностика серверної конфігурації. У межах цієї перевірки система аналізує:

- версію середовища виконання PHP;
- наявність і доступність необхідних програмних модулів;
- параметри підключення до бази даних;
- коректність роботи встановлених розширень.

Результати такої перевірки дозволяють підтвердити, що контейнеризоване середовище повністю відповідає технічним вимогам платформи Moodle та готове до подальшої експлуатації.

Server checks				
Name	Information	Report	Plugin	Status
unicode		❗ must be installed and enabled		OK
database	mysql (8.0.27)	❗ version 5.6 is required and you are running 8.0.27		OK
php		❗ version 7.2.0 is required and you are running 7.4.26		OK
pcreunicode		❗ should be installed and enabled for best results		OK
php_extension	iconv	❗ must be installed and enabled		OK
php_extension	mbstring	❗ must be installed and enabled		OK
php_extension	curl	❗ must be installed and enabled		OK
php_extension	openssl	❗ must be installed and enabled		OK
php_extension	tokenizer	❗ should be installed and enabled for best results		OK
php_extension	xmlrpc	❗ should be installed and enabled for best results		OK
php_extension	soap	❗ should be installed and enabled for best results		OK
php_extension	ctype	❗ must be installed and enabled		OK
php_extension	zip	❗ must be installed and enabled		OK
php_extension	zlib	❗ must be installed and enabled		OK
php_extension	gd	❗ must be installed and enabled		OK
php_extension	simplexml	❗ must be installed and enabled		OK
php_extension	spl	❗ must be installed and enabled		OK

Рисунок 4.1 – Перевірка відповідності серверного середовища

Окрім перевірки наявності необхідних програмних компонентів, на даному етапі тестування здійснюється оцінювання узгодженості налаштувань між окремими контейнерами системи. Зокрема, аналізується коректність взаємодії між контейнером вебсервера, середовищем виконання PHP та контейнером бази даних MySQL, що є визначальним чинником стабільної роботи платформи дистанційного навчання. Використання автоматизованих засобів перевірки дозволяє знизити ймовірність виникнення помилок, пов'язаних із ручним налаштуванням серверного середовища.

Застосування контейнеризованої інфраструктури позитивно відображається на результатах тестування, оскільки всі необхідні PHP-розширення та параметри конфігурації були задані ще на етапі формування Docker-образів. Це забезпечує відтворюваність середовища виконання та гарантує відповідність технічним вимогам платформи Moodle незалежно від апаратної платформи або операційної системи сервера. У разі масштабування системи чи розгортання додаткових вузлів перевірка серверного середовища здійснюється за аналогічною схемою, без необхідності виконання додаткових налаштувань.

Таким чином, результати перевірки серверного середовища свідчать про те, що розроблена контейнеризована архітектура повністю відповідає технічним вимогам платформи Moodle та готова до подальших етапів конфігурації й експлуатації. Успішне завершення цього етапу формує надійну основу для тестування функціональних можливостей системи, роботи з обліковими записами користувачів, навчальними курсами та освітнім контентом у межах навчального процесу.

4.2. Налаштування конфігураційного файлу Moodle

На даному етапі виконується редагування файлу `config.php`, який є основним конфігураційним файлом платформи Moodle та містить ключові параметри функціонування системи. У цьому файлі визначаються налаштування підключення до бази даних, шляхи до системних і користувацьких каталогів, а також базові параметри безпеки та доступу. Коректне заповнення конфігураційних параметрів є необхідною умовою для стабільної роботи платформи в контейнеризованому середовищі.

Внесення змін до файлу `config.php` забезпечує узгоджену взаємодію між вебсервером, середовищем виконання PHP та контейнером бази даних, які функціонують як окремі компоненти системи. Завдяки цьому досягається правильна синхронізація роботи сервісів, коректна обробка запитів і доступ до даних, що є критично важливим для подальшого розгортання та експлуатації системи дистанційного навчання.

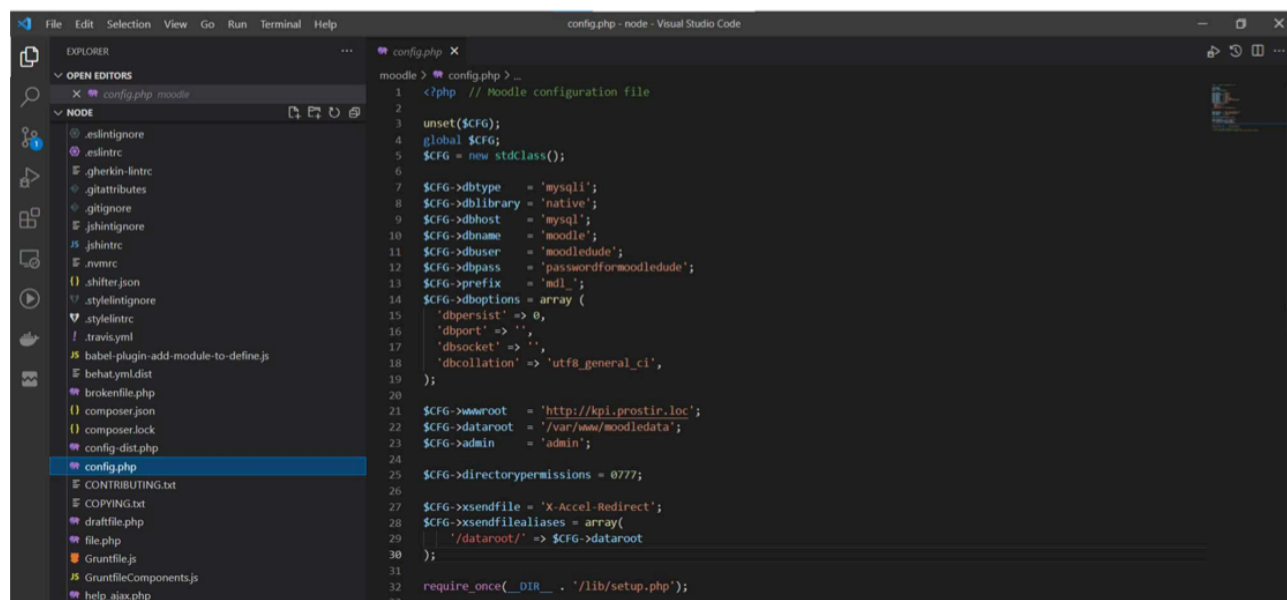


Рисунок 4.2 – Конфігураційний файл Moodle

Після завершення налаштування конфігураційного файлу здійснюється фінальний етап встановлення системи. У процесі інсталяції платформа Moodle автоматично формує структуру таблиць у базі даних, створює необхідні системні компоненти та виконує ініціалізацію основних модулів. Успішне завершення цього етапу підтверджується відповідним повідомленням системи, що свідчить про коректне встановлення та готовність платформи до подальшої роботи.

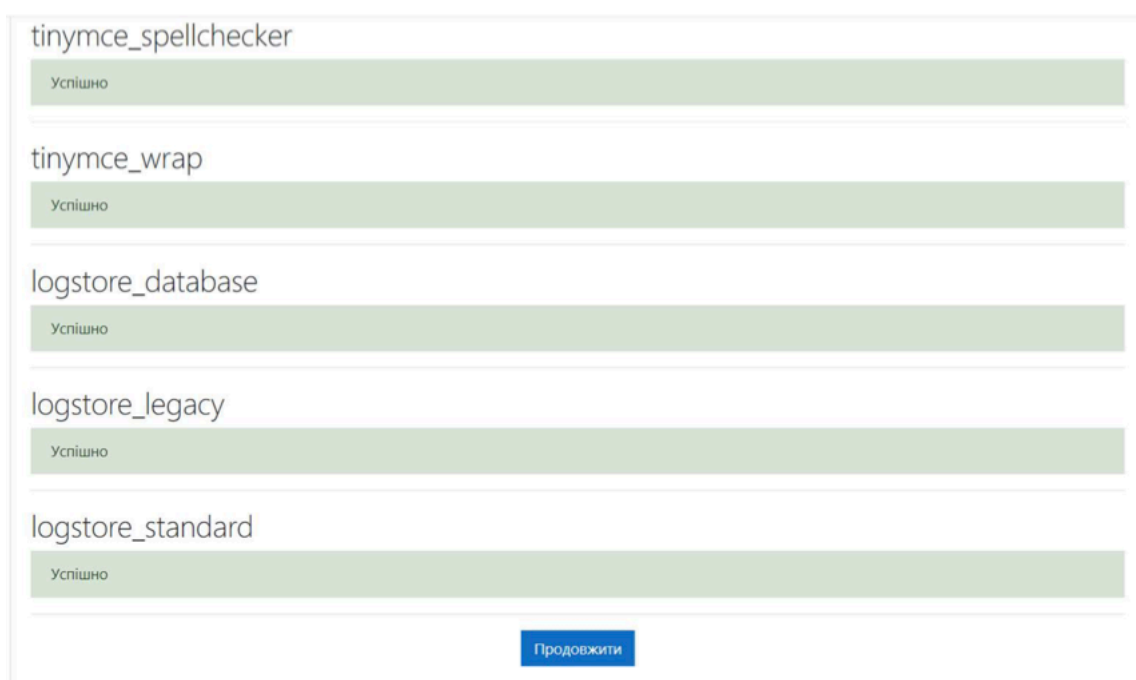


Рисунок 4.3 – Завершення інсталяції системи

Окрім контролю наявності необхідних програмних компонентів, на даному етапі тестування здійснюється оцінка узгодженості конфігурацій між окремими контейнерами системи. Зокрема, перевіряється правильність взаємодії між контейнером вебсервера, середовищем виконання PHP та контейнером бази даних MySQL, що є ключовою умовою стабільного функціонування платформи дистанційного навчання. Використання автоматизованих засобів перевірки дає змогу суттєво зменшити ймовірність виникнення помилок, пов'язаних із ручним налаштуванням серверного середовища.

Контейнеризована інфраструктура позитивно впливає на результати тестування, оскільки всі необхідні PHP-розширення та параметри конфігурації були визначені ще на етапі формування Docker-образів. Це забезпечує відтворюваність середовища виконання та гарантує відповідність технічним вимогам платформи Moodle незалежно від апаратної платформи або операційної системи сервера. У випадку масштабування системи чи розгортання додаткового вузла перевірка серверного середовища здійснюється за аналогічною схемою без потреби виконання додаткових ручних налаштувань.

Таким чином, результати перевірки серверного середовища свідчать про те, що розроблена контейнеризована архітектура повністю відповідає технічним вимогам платформи Moodle та є готовою до подальших етапів конфігурації й експлуатації. Успішне завершення цього етапу формує надійну основу для тестування функціональних можливостей системи, організації роботи з користувачами, навчальними курсами та освітнім контентом у межах навчального процесу.

4.3. Створення адміністратора системи

Після завершення процесу інсталяції платформи відкривається інтерфейс створення облікового запису адміністратора системи. На даному етапі визначаються основні параметри адміністративного доступу, зокрема персональні дані користувача та налаштування, необхідні для подальшого керування системою.логін і пароль,

- електронна пошта,
- ім'я користувача,
- налаштування автентифікації.

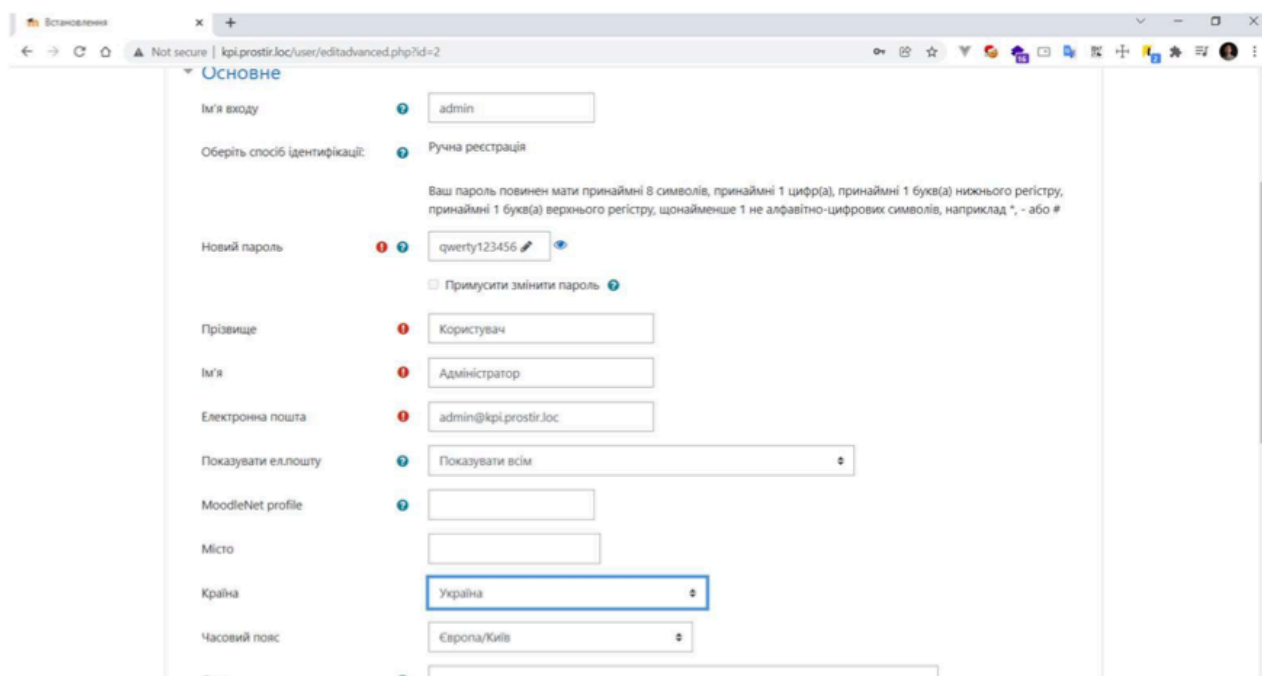


Рисунок 4.4 – Інтерфейс створення адміністратора

Для перевірки коректності роботи системи створюється тестовий курс. На цьому етапі перевіряється:

- робота інтерфейсу Moodle,
- можливість додавання курсів,
- функціонування елементів навчального матеріалу.

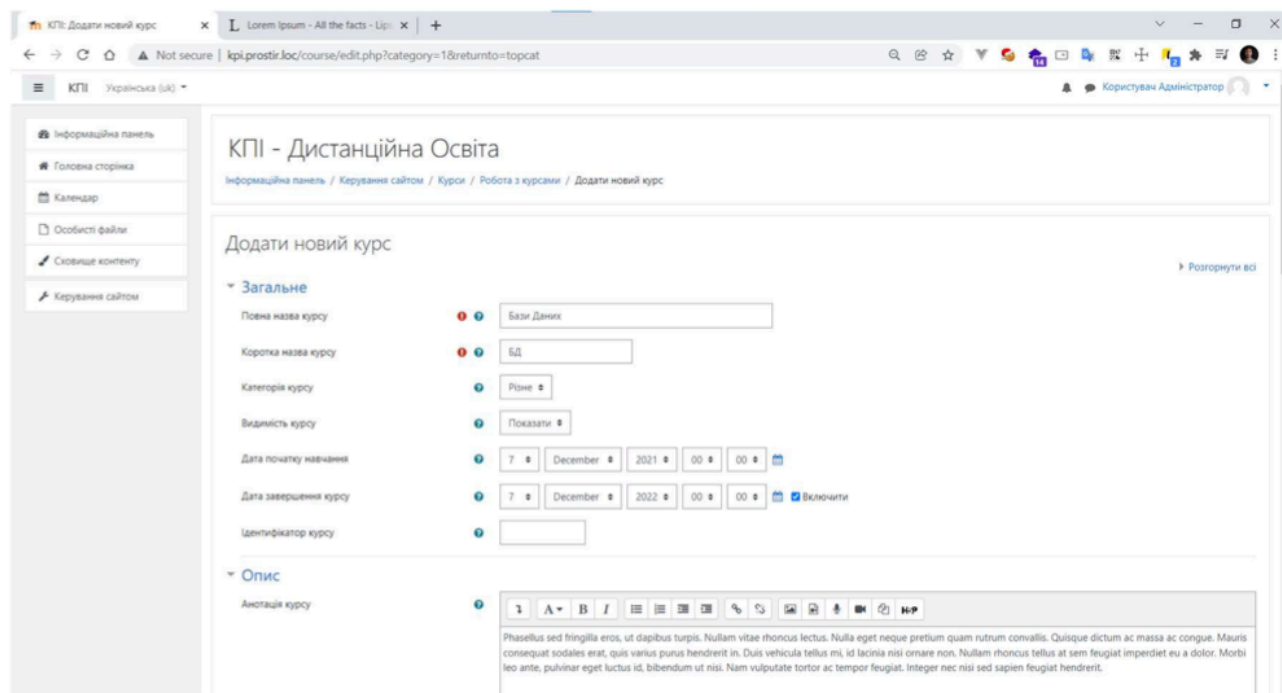


Рисунок 4.5 – Інтерфейс створення нового курсу

Завершальним етапом тестування є створення нового облікового запису користувача, призначення йому відповідної ролі та перевірка процесу авторизації в системі. Виконання цього кроку підтверджує коректну роботу бази даних, вебсервера та внутрішніх механізмів платформи Moodle, а також готовність системи до повноцінної експлуатації в межах освітнього процесу.

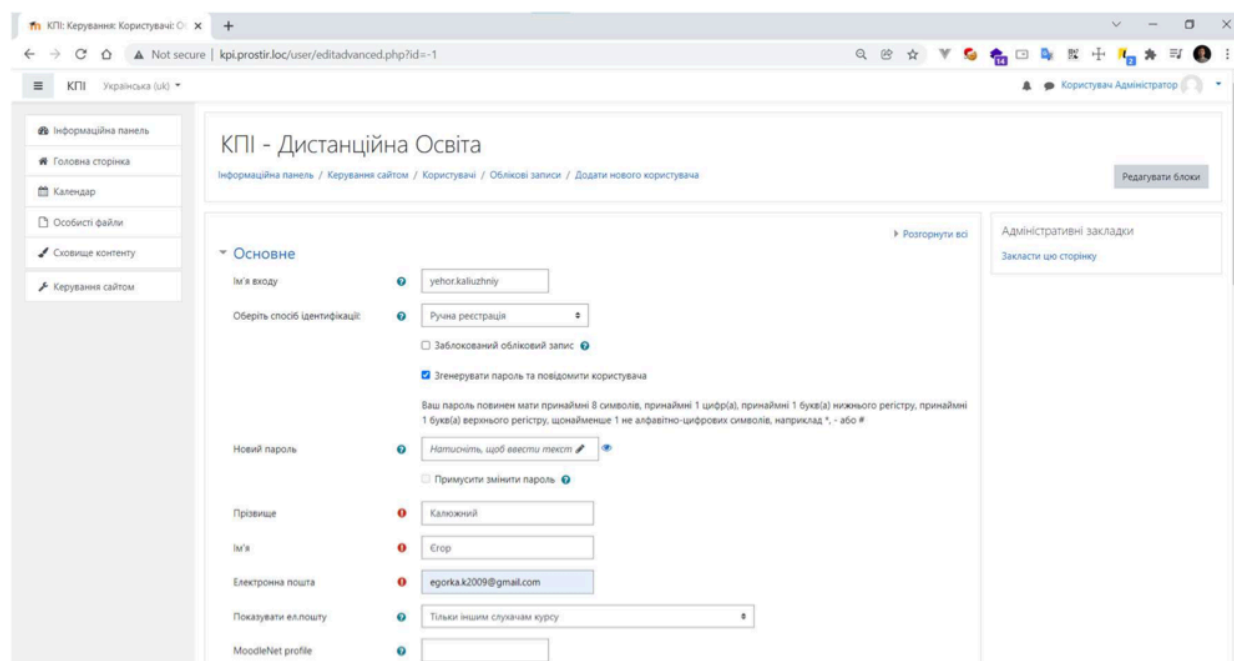


Рисунок 4.6 – Інтерфейс створення нового користувача

На рисунку наведено інтерфейс створення нового користувача в системі дистанційного навчання Moodle після успішного завершення інсталяції та виконання базових налаштувань. На відповідній сторінці відображено адміністративну форму, яка дозволяє ввести основні дані користувача, зокрема ім'я для входу, спосіб автентифікації, пароль, прізвище, ім'я, адресу електронної пошти та параметри профілю. Крім того, передбачені функції примусової зміни пароля під час першого входу, блокування облікового запису, призначення ролей і налаштування параметрів безпеки.

Наявність та коректна робота цього інтерфейсу свідчать про повну працездатність системи, правильне підключення до бази даних і готовність платформи до роботи з реальними користувачами. Таким чином, завершальний етап конфігурації підтверджує готовність веборієнтованого середовища Moodle до подальшої експлуатації в межах освітнього процесу.

РОЗДІЛ 5

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

5.1. Структурно-функціональний аналіз технологічного процесу

Розробка та вживання ефективних заходів запобігання аварійним і травмонебезпечним ситуаціям можливі лише при завчасному виявленні тих небезпек, з яких починаються процеси їх формування. Оскільки небезпечні умови не завжди завчасно можна виявити, а для вивчення небезпечних дій іноді потрібно багато часу, щоб зібрати статичний матеріал, то і методи виявлення цих небезпек повинні бути відповідно диференційовані (табл. 5.1).

Таблиця 4.1. - Моделі формування та виникнення травмонебезпечних і аварійних ситуацій

Вид робіт, виробничий підрозділ, робоче місце, виробниче обладнання, склад агрегату	Виробнича небезпека			Можливі наслідки	Заходи запобігання небезпечним ситуаціям
	Небезпечна умова (НУ)	Небезпечна дія (НД)	Небезпечна ситуація (НС)		
Виконання робіт із електрооблад нанням	Не вимкнено живлення. Відсутність заземлення.	Нехтування правилами ТБ	Ураження струмом	Травма (Т)	Проведення повторного інструктажу з ТБ. Розробка нових способів захисту. Встановлення заземлення.
<div style="display: flex; justify-content: center; align-items: center; gap: 20px;"> <div style="text-align: center;">НД ↓</div> <div style="text-align: center;">НУ → НС → Т</div> </div>					

Відповідно до аналізу небезпечних умов, які існують у виробничому процесі виокремлено такі наступні за характером дії на працівника їх групи:

- характеризують стан або рівень безпеки обладнання, які використовуються.
- сприяють виникненню технологічних помилок обслуговуючого персоналу впродовж виробничого процесу;

- створювати умови та можливість проникнення працівника в небезпечну зону;
- приводять до виникнення небезпечних дій (внаслідок низького рівня професійної підготовки працівників та організації навчання з охорони праці).

Моделі формування та виникнення травмонебезпечних і аварійних ситуацій в комп'ютерному кабінеті представлено у вигляді моделі формування та виникнення травмонебезпечних і аварійних ситуацій – табл. 5.1.

5.2. Розрахунок освітлення приміщення комп'ютерного кабінету.

Освітленість виробничих приміщень може бути штучною і природною. Природне освітлення при правильному обладнанні найбільш сприятливе для людини. Основні вимоги для освітлення наступні:

- освітлення повинне бути достатнім для швидкого і легкого розпізнання об'єктів роботи;
- освітлення повинно бути рівномірне без різких тіней;
- джерело світла не повинно осліплювати працівника;
- рівень освітленості не повинен обмежуватись часом.

Природне освітлення забезпечується обладнанням вікон (бокове освітлення) фонарів і світильних покриттів приміщень (верхнє освітлення). Природне освітлення нормується коефіцієнтом природної освітленості. Коефіцієнт природної освітленості – це процентне відношення фактичної освітленості E_v в будь-якій точці приміщення до освітленості E_n розсіяної світлом небозводу точки, яка лежить на відкритій місцевості. Розрахунок природного освітлення через бокові вікна по нормам освітленості ведеться для самої дальньої від вікон точки, тобто знаходять мінімальне значення стик коефіцієнта природної освітленості:

$$e_{\min} = \frac{F_b}{F_H} \cdot 100. \quad (4.1)$$

Значення коефіцієнта природної освітленості визначається не менше чим в п'яти точках. Значення коефіцієнта природної освітленості для сільськогосподарських виробничих приміщень в даному випадку ремонтній майстерні, беремо $e_{\min} = 5\%$.

Розрахунок природного освітлення зводиться до визначення площі світлових променів.

Сумарну площу світлових променів $\sum F_o (m^2)$ по коефіцієнту природної освітленості для бокових променів визначаємо по формулі:

$$\sum F_o = \frac{F_H \cdot e_{\min} \cdot r_o \cdot K}{100 \cdot \tau \cdot \Gamma_1}, \quad (4.2)$$

де F_H – площа підлоги; e_{\min} – величина мінімального коефіцієнта природного освітлення; τ – загальний коефіцієнт світловикористання віконного отвору із врахуванням його забруднення, $\tau = 0,25$; r_o – світлова характеристика вікна, $r_o = 9,5$; Γ_1 – коефіцієнт, який враховує підвищення освітленості за рахунок світла, яке відбивається від стін і стелі, $\Gamma_1 = 1,2$; K – коефіцієнт, який враховує затінення вікон сусідніми приміщеннями і загорожею, $K = 1$.

$$\sum F_o = \frac{36 \cdot 0,5 \cdot 9,5 \cdot 1}{100 \cdot 0,25 \cdot 1,2} = 5,7 m^2.$$

Кількість світлових променів визначимо:

$$N = \frac{\sum F_o}{F_o}, \quad (4.3)$$

де F_o – площа вікна згідно стандарту, m^2 .

$$N = \frac{5,7}{6} = 0,95.$$

Приймаємо кількість вікон – одне вікно.

При розрахунку природного освітлення найбільш поширеним і простим є метод світлового потоку. При цьому методі розраховуємо світловий потік F_l (Лк), який повинна випромінювати кожна лампа (при заданій кількості ламп).

$$F_l = \frac{k \cdot S_n \cdot E}{n_l \cdot \eta \cdot r^2}, \quad (4.4)$$

де k – коефіцієнт запасу, $k = 1,3$; S_n – площа підлоги, м²; $S_n = 36$ м².

E – нормативна освітленість, $E = 300$ Лк; n_l – кількість встановлених ламп, $n_l = 6$ од; η – коефіцієнт використання світлового потоку, $\eta = 0,25$; r – коефіцієнт нерівномірності освітленості, $r = 0,545$.

Коефіцієнт запасу (K) враховує можливість забруднення світильників пилом, що залежить від характеру виробництва.

Розрахунок штучного освітлення починаємо із визначення висоти розташування світильника і їх кількості. Висоту h_n (м) розташування світильників над робочим місцем знаходимо за формулою:

$$h_n = H - (h_1 + h_2), \quad (4.5)$$

де H – висота приміщення, м; h_1 – віддаль від підлоги до освітлювальної поверхні, м; h_2 – віддаль від стелі до світильника, м.

$$h_n = 4,5 - (2,2 + 1,5) = 0,8 \text{ м.}$$

При симетричному розміщенні світильників по вершинах квадратів їх кількість визначається за формулою:

$$n_c = \frac{S_n}{l^2}, \quad (4.6)$$

де l – віддаль між світильниками, м.

Підставивши значення отримаємо:

$$n_c = \frac{36}{9} = 4 \text{ од.}$$

Тоді світловий потік буде становити

$$F_n = \frac{1,3 \cdot 36 \cdot 300}{4 \cdot 0,25 \cdot 0,545} = 2576,2 \text{ Лк.}$$

При світловому потоці 2576,2 Лк для заданої лампи вибираємо тип і потужність.

Вибираємо тип лампи – люмінесцентну, потужністю 40Вт.

5.3. Безпека в надзвичайних ситуаціях

Забезпечення захисту населення і території у разі загрози і виникнення надзвичайних ситуацій є одним з найважливіших завдань держави.

Захист населення є системою загально-державних заходів, які реалізуються центральними і місцевими органами виконавчої влади, виконавчими органами влад, органами управління з питань надзвичайних ситуацій та цивільного захисту населення, підпорядкованими їм системами, та підприємств, що забезпечують виконання організаційних, інженерно – технічних, санітарно – гігієнічних, проти епідемічних та інших заходів у сфері запобігання та ліквідації наслідків надзвичайних ситуацій.

Загрози життєво важливих інтересів громадян, держави, суспільства поділяють на зовнішні та внутрішні, виконують під час надзвичайних ситуацій техногенного та природного характеру та воєнних конфліктах.

Принципи захисту випливають з основних положень Женевської конвенції щодо захисту жертв війни та додаткових протоколів до неї, можливого характеру воєнних дій, реальних можливостей держави щодо створення матеріальної бази захисту. З метою захисту населення, зменшення втрат та шкоди економіці в разі виникнення надзвичайних ситуацій має право проводитись спеціальний комплекс заходів.

ВИСНОВКИ

У ході виконання магістерської кваліфікаційної роботи було вирішено науково-технічну задачу створення та дослідження інформаційної системи дистанційного навчання з можливістю формування єдиного інформаційного простору та відновлення параметрів інформаційних об'єктів на основі методу k -найближчих сусідів.

У першому розділі проведено аналітичний огляд сучасного стану дистанційної освіти, досліджено її методи, форми та інструменти, а також визначено ключові тенденції цифровізації освітнього процесу. Особливу увагу приділено порівнянню синхронних, асинхронних та змішаних форматів навчання, що дало змогу обґрунтувати актуальність розробки системи, здатної інтегрувати різні підходи та забезпечувати гнучкість освітнього середовища.

У другому розділі сформовано концепцію єдиного інформаційного простору системи дистанційного навчання та запропоновано метод відновлення параметрів інформаційних об'єктів на основі алгоритму k -найближчих сусідів. Побудовано онтологічну модель системи, визначено взаємозв'язки між навчальними об'єктами, розроблено функціональні моделі та структуру бази даних. Запропонований підхід забезпечує автоматизацію процесів обробки, аналізу та корекції даних у системі, що підвищує її стабільність та інформативність.

У третьому розділі розроблено архітектуру програмної реалізації вузла системи на базі контейнеризації. Здійснено проектування та розгортання програмного середовища Moodle у багатоконтейнерній інфраструктурі, що включає вебсервер, сервер виконання PHP-коду та сховище даних. Використання Docker та Docker-Compose забезпечило модульність, масштабованість, високу доступність і можливість незалежного оновлення окремих компонентів системи. На основі побудованої архітектури виконано налаштування сервера, бази даних, конфігураційних файлів і запуск навчального порталу.

У четвертому розділі проведено інсталяцію та тестування програмного продукту. Перевірено відповідність серверного середовища вимогам Moodle, виконано початкові налаштування системи, створено адміністратора, користувачів, курси та ролі. Результати тестування підтвердили працездатність розробленого середовища, коректність взаємодії між контейнерами та стабільність роботи веборієнтованої системи при виконанні основних функцій дистанційного навчання.

Отже, у роботі досягнуто поставленої мети — розроблено та впроваджено інформаційний вузол системи дистанційного навчання на базі архітектури контейнеризації та методу відновлення параметрів інформаційних об'єктів. Запропоновані технічні та методологічні рішення забезпечують можливість побудови масштабованого, надійного та гнучкого освітнього середовища, яке здатне інтегрувати навчальні ресурси різних закладів у спільний інформаційний простір. Отримані результати можуть бути використані для подальшої модернізації системи, розширення її функціоналу та впровадження інтелектуальних механізмів адаптивного навчання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Класифікація веб-сайтів [Електронний ресурс]. – Режим доступу: <https://sites.google.com/site/yaremusinform/home> (дата звернення: 15.10.2025).
2. Negative outcomes of Internet use: A qualitative analysis in the homes of families with different educational backgrounds [Electronic resource]. – Mode of access: <https://www.tandfonline.com/doi/full/10.1080/01972243.2019.1649774> (date of access: 15.10.2025).
3. Communication: Online vs. Face-to-Face Interactions [Electronic resource]. – Mode of access: <https://psychminds.com/communication-online-vs-face-to-face-interactions/> (date of access: 15.10.2025).
4. What is the World Wide Web? [Electronic resource]. – Mode of access: <https://www.bbc.co.uk/bitesize/topics/zkcqn39/articles/z2nbgk7> (date of access: 15.11.2025).
5. What is HTML? [Electronic resource]. – Mode of access: <https://www.yourhtmlsource.com/starthere/whatishtml.html> (date of access: 15.11.2025).
6. What is CSS, and why is it important? [Electronic resource]. – Mode of access: <https://www.bigcommerce.com/ecommerce-answers/what-css-and-why-it-important/> (date of access: 15.11.2025).
7. JavaScript (JS) [Electronic resource]. – Mode of access: <https://www.techopedia.com/definition/3929/javascript-js> (date of access: 15.11.2025).
8. JavaScript [Електронний ресурс]. – Режим доступу: <https://scriptdev.ru/ts/007/> (дата звернення: 15.11.2025).
9. What is TypeScript and why should you use it? [Electronic resource]. – Mode of access:

<https://learn.coderslang.com/0056-what-is-typescript-and-why-should-you-use-it/> (date of access: 15.11.2025).

10. Angular [Electronic resource]. – Mode of access: <https://www.typescriptlang.org/docs/handbook/angular.html> (date of access: 15.11.2025).

11. 8 Proven Reasons You Need Angular for Your Next Development Project [Electronic resource]. – Mode of access: <https://www.grazitti.com/blog/8-proven-reasons-you-need-angular-for-your-next-development-project/> (date of access: 15.11.2025).

12. What is API: Definition, Types, Specifications, Documentation [Electronic resource]. – Mode of access: <https://www.altexsoft.com/blog/engineering/what-is-api-definition-types-specifications-documentation/> (date of access: 15.12.2025).

13. REST API [Электронный ресурс]. – Режим доступа: <https://blog.skillfactory.ru/glossary/rest-api/> (дата звернення: 15.12.2025).

14. Murray N., Coury F., Lerner A., Taborda C. *ng-book: The Complete Guide to Angular*. – 5th ed. – Technical Editor: Frode Fikke. – 2018. – 626 p.

15. What Are Angular Services and Why Should You Use Them? [Electronic resource]. – Mode of access: <https://chudovo.com/what-are-angular-services-and-why-should-you-use-them/> (date of access: 15.12.2025).

16. What is Bootstrap and Why You Should Use It in Web Development [Electronic resource]. – Mode of access: <https://www.yogihosting.com/what-is-bootstrap/> (date of access: 15.12.2025).

17. Why You Should Use NestJS for Backend Development? [Electronic resource]. – Mode of access: <https://enlear.academy/why-you-should-use-nestjs-as-your-backend-framework-bd1ff1acce5d> (date of access: 15.12.2025).

18. Magolan G., Bell J., Guijarro D., Peretti A., Housley P. *Nest.js: A Progressive Node.js Framework*. – Kindle Edition. – Bleeding Edge Press, 2018. – 350 p.
19. What is MongoDB? A Quick Guide for Developers [Electronic resource]. – Mode of access: <https://www.infoworld.com/article/3623357/what-is-mongodb-a-quick-guide-for-developers.html> (date of access: 15.12.2025).
20. Bradshaw S., Brazil E., Chodorow K. *MongoDB: The Definitive Guide*. – 3rd ed. – O'Reilly Media, 2019. – 514 p.
21. Page Regions [Electronic resource]. – Mode of access: <https://www.w3.org/WAI/tutorials/page-structure/regions/> (date of access: 15.12.2025).
22. What Is a User Interface? (Definition, Types and Examples) [Electronic resource]. – Mode of access: <https://www.indeed.com/career-advice/career-development/user-interface> (date of access: 15.12.2025).

Додаток А

Фрагменти конфігураційних файлів контейнеризованої системи Moodle

А.1. Фрагмент Dockerfile для контейнера PHP-FPM

```
ARG LARADOCK_PHP_VERSION
ARG BASE_IMAGE_TAG_PREFIX=latest
FROM laradock/php-fpm:${BASE_IMAGE_TAG_PREFIX}-${LARADOCK_PHP_VERSION}

ARG INSTALL_XSL=false
RUN if [ ${INSTALL_XSL} = true ]; then \
    apt-get -y install libxslt-dev && \
    docker-php-ext-install xsl \
;fi
```

А.2. Фрагмент Dockerfile для контейнера Nginx

```
FROM nginx:alpine

COPY nginx.conf /etc/nginx/
RUN apk add --no-cache curl

ARG PHP_UPSTREAM_CONTAINER=php-fpm
ARG PHP_UPSTREAM_PORT=9000

RUN echo "upstream php-upstream { server
${PHP_UPSTREAM_CONTAINER}:${PHP_UPSTREAM_PORT}; }" \
    > /etc/nginx/conf.d/upstream.conf \
    && rm /etc/nginx/conf.d/default.conf
```

А.3. Конфігураційний файл Nginx для Moodle

```
server {
    listen 80 default_server;
    server_name kpi.prostir.loc;

    root /var/www/moodle;
    index index.php;

    location / {
        try_files $uri $uri/ /index.php?$args;
    }

    location ~ /\.php$ {
```

```

        include fastcgi_params;
        fastcgi_pass php-upstream;
        fastcgi_param SCRIPT_FILENAME
$document_root$fastcgi_script_name;
    }

    location /dataroot/ {
        internal;
        alias /var/www/moodledata/;
    }
}

```

A.4. Фрагмент Dockerfile для контейнера MySQL

```

ARG MYSQL_VERSION
FROM mysql:${MYSQL_VERSION}

COPY my.cnf /etc/mysql/conf.d/my.cnf
RUN chmod 0444 /etc/mysql/conf.d/my.cnf

EXPOSE 3306
CMD ["mysqld"]

```

A.5. Пример фрагмента config.php Moodle

```

$CFG->dbtype      = 'mysqli';
$CFG->dbname      = 'moodle';
$CFG->dbuser      = 'root';
$CFG->dbpass      = 'passwordformoodle';
$CFG->wwwroot     = 'http://kpi.prostir.loc';
$CFG->dataroot    = '/var/www/moodledata';
$CFG->directorypermissions = 0777;

```