

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ І. І. МЕЧНИКОВА

(повне найменування закладу вищої освіти)

Факультет математики, фізики та інформаційних технологій

(повне найменування факультету)

Кафедра комп'ютерних систем та технологій

(повна назва кафедри)

## Кваліфікаційна робота

на здобуття ступеня вищої освіти «магістр»

«Дослідження методів інтеграції апаратних та програмних систем  
у Веб-застосунки»

(тема кваліфікаційної роботи українською мовою)

«Research of methods of integration of hardware and software systems into Web  
applications»

(тема кваліфікаційної роботи англійською мовою)

Виконав(ла): здобувач(ка) денної/заочної форми навчання  
спеціальності 123 Комп'ютерна інженерія

(код, назва спеціальності)

Освітня програма \_\_\_\_\_

(назва)

Яблонський Кирило Михайлович

(прізвище, ім'я, по-батькові здобувача)

Керівник д.т.н., проф. Гунченко Ю.О

(науковий ступінь, вчене звання, прізвище, ініціали) (підпис)

Рецензент ст. викл. Мартинович Л.Я

(науковий ступінь, вчене звання, прізвище, ініціали)

Рекомендовано до захисту: Протокол засідання кафедри № _____ від _____. _____. 20__ р.	Захищено на засіданні ЕК № _____ протокол № __ від _____. _____. 20__ р.
Завідувач(ка) кафедри _____ <u>Гунченко Ю.О</u> (підпис) (прізвище, ім'я)	Оцінка _____ / _____ / _____ (за національною шкалою/шкалою ECTS/ бали) Голова ЕК _____ _____ (підпис) (прізвище, ім'я)

Одеса 2023

## АНОТАЦІЯ

Магістерська робота "Дослідження методів інтеграції апаратних та програмних систем у веб-застосунки" присвячена створенню інноваційної платформи для управління навчальним процесом, яка відповідає сучасним вимогам EdTech.

Предметом роботи є процес інтеграції апаратних та програмних рішень у створення ефективної та інтерактивної освітньої платформи. Проект зосереджений на створенні передової Learning Experience Platform (LXP) для університетів, спрямованої на трансформацію та підвищення ефективності навчального процесу.

Метою є створення системи, яка не тільки відповідає сучасним освітнім вимогам, але й пропонує інноваційні рішення для персоналізації навчання та підвищення залученості студентів.

Об'єктом проекту є комплексний навчальний процес у вищих навчальних закладах, який включає у себе управління курсами, інтерактивне навчання, оцінювання та адміністрування.

В роботі детально аналізується проблематика існуючих систем управління навчанням та висвітлюються ключові недоліки, такі як недостатня інтеграція з новітніми онлайн-інструментами, обмежена адаптивність до специфіки різних учбових закладів та низька масштабованість.

В рамках дослідження розроблено модульну систему на базі фреймворку Symfony, яка використовує Vue.js для створення реактивного користувацького інтерфейсу та Go для оптимізації серверних процесів з високою завантаженістю.

Результати дослідження вказують на значне підвищення ефективності управління навчальними процесами та зручності їх використання, що відкриває нові перспективи для розвитку освітнього середовища.

Ключові слова: інтеграція систем, веб-застосунки, EdTech, управління навчальним процесом, Symfony, Vue.js, Go, модульна архітектура, онлайн-навчання, високопродуктивні серверні процеси, інтерактивність.

## ANNOTATION

The master's thesis "Research on Methods of Integrating Hardware and Software Systems into Web Applications" is dedicated to the creation of an innovative platform for managing the educational process, which meets modern EdTech requirements.

The subject of the work is the process of integrating hardware and software solutions into the creation of an effective and interactive educational platform. The project focuses on creating an advanced Learning Experience Platform (LXP) for universities, aimed at transforming and enhancing the efficiency of the educational process.

The goal is to create a system that not only meets modern educational requirements but also offers innovative solutions for personalizing learning and increasing student engagement.

The object of the project is the comprehensive educational process in higher education institutions, which includes managing courses, interactive learning, assessment, and administration.

The work thoroughly analyzes the problems of existing learning management systems and highlights key shortcomings such as insufficient integration with cutting-edge online tools, limited adaptability to the specifics of various educational institutions, and low scalability.

A modular system based on the Symfony framework has been developed within the research, utilizing Vue.js to create a reactive user interface and Go for optimizing high-load server processes.

The research results indicate a significant increase in the efficiency of managing educational processes and the convenience of their use, opening new prospects for the development of the educational environment. Keywords: system integration, web applications, EdTech, educational process management, Symfony, Vue.js, Go, modular architecture, online learning, high-performance server processes, interactivity, personalized learning.

# ЗМІСТ

Стор.

1. ТЕОРЕТИЧНІ ОСНОВИ ІНТЕГРАЦІЇ АПАРАТНИХ ТА ПРОГРАМНИХ СИСТЕМ У ВЕБ-ЗАСТОСУНКИ.....	9
1.1. Огляд сучасного стану інтеграції систем в освіті.....	9
1.2. Аналіз існуючих підходів до інтеграції апаратних та програмних систем.....	13
1.3. Принципи та методології інтеграції веб-застосунків.....	19
1.4. Виклики та перспективи розвитку інтегрованих систем в освіті..	20
2. АНАЛІЗ ВИМОГ ТА РОЗРОБКА АРХІТЕКТУРИ СИСТЕМИ УПРАВЛІННЯ НАВЧАЛЬНИМ ПРОЦЕСОМ.....	22
2.1. Мета, об'єкт та задачі проекту.....	22
2.2. Визначення основних функціональних вимог.....	23
2.3. Аналіз потреб користувачів та стейкхолдерів.....	24
2.4. Моделювання процесів управління навчальним процесом.....	27
2.5. Проектування архітектури системи.....	29
2.6. Вибір технологічного стеку для реалізації платформи.....	32
2.7. Розробка модулів інтеграції апаратних та програмних рішень.....	38
3. ІНТЕГРАЦІЯ АПАРАТНИХ ТА ПРОГРАМНИХ СИСТЕМ ТА ЇХ ТЕСТУВАННЯ.....	42
3.1. Методи інтеграції апаратних засобів.....	42
3.2. Інтеграція програмних рішень та API.....	48
3.3. Забезпечення безпеки при інтеграції систем.....	50
3.4. Методи тестування інтегрованих систем.....	53
3.5. Оптимізація продуктивності та надійності системи .....	55
3.6. Валідація функціональності та користувацького досвіду.....	58
4. ВПРОВАДЖЕННЯ, ОЦІНКА ЕФЕКТИВНОСТІ СИСТЕМИ ТА ПЕРСПЕКТИВИ РОЗВИТКУ .....	62
4.1. Планування впровадження системи в освітній процес .....	62
4.2. Оцінка впливу системи на якість навчального процесу .....	66
4.3. Збір та аналіз зворотного зв'язку від користувачів .....	68
ВИСНОВОК.....	70
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	72
ДОДАТОК А Технічна документація системи.....	74
ДОДАТОК Б Користувацькі інструкції.....	76
ДОДАТОК В Код програми.....	78

## ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ

EdTech - Освітні технології (Educational Technology).

API - Інтерфейс програмування додатків (Application Programming Interface).

MVC - Модель-вид-контролер (Model-View-Controller).

Symfony - PHP фреймворк для веб-додатків.

Vue.js - JavaScript фреймворк для розробки інтерфейсів користувача.

Go - Мова програмування, розроблена в Google.

CRUD - Створення, читання, оновлення, видалення (Create, Read, Update, Delete).

UI/UX - Інтерфейс користувача / Досвід користувача (User Interface / User Experience).

CI/CD - Неперервна інтеграція / Неперервне розгортання (Continuous Integration / Continuous Deployment).

Selenium - Інструмент для автоматизації тестування веб-додатків.

New Relic - Платформа моніторингу та аналітики.

Prometheus - Система моніторингу та алертування.

Redis - Система керування базами даних в пам'яті.

Memcached - Розподілена система кешування в пам'яті.

Nginx - Веб-сервер та обратний проксі-сервер.

HAProxy - Програмне забезпечення для балансування навантаження.

Docker - Платформа для контейнеризації додатків.

Kubernetes - Система для автоматизації розгортання, масштабування та управління контейнеризованими додатками.

## ВСТУП

Сучасний світ стоїть на порозі нової ери освітніх технологій (EdTech), де інтеграція апаратних та програмних систем у веб-застосунки відіграє ключову роль. Впровадження інновацій у навчальний процес набуває нового значення, оскільки цифровізація освіти перетворюється з опціонального тренду на необхідність. Ця магістерська робота "Дослідження методів інтеграції апаратних та програмних систем у веб-застосунки" покликана вирішити ряд проблем, що стоять перед сучасними освітніми установами, а саме - розробка платформи, що втілює інноваційні підходи до управління навчальним процесом.

Метою цього дослідження є підвищення ефективності та поліпшення якості навчального процесу через розробку модульної системи, яка відповідає сучасним вимогам EdTech. Система розробляється з урахуванням специфіки різних освітніх установ та забезпечує високий рівень масштабованості та адаптивності. Актуальність проекту обумовлена потребою усунення недоліків існуючих систем управління навчанням, таких як недостатня інтеграція з сучасними онлайн-інструментами, обмежена гнучкість та відсутність ефективних можливостей для персоналізації навчального процесу.

Аналіз існуючих систем показує, що багато з них не здатні повною мірою задовольнити потреби сучасного навчального процесу. Проблеми сумісності, обмежена інтеграція з зовнішніми інструментами та недостатня гнучкість є лише деякими з викликів, з якими стикаються освітні установи. В цьому контексті розробка платформи, що відповідає вимогам сучасності, стає не тільки технічним завданням, але й стратегічним кроком на шляху до модернізації освітнього процесу.

В ході дослідження було вибрано комбінацію фреймворку Symfony для бекенду, Vue.js для фронтенду та Go для оптимізації серверних процесів. Такий підхід дозволив розробити систему, яка відрізняється високою продуктивністю, масштабованістю та зручністю інтерфейсу. Важливим аспектом є й інтеграція системи з широким спектром освітніх інструментів, що включає системи для відеоконференцій, інтерактивні дошки, тести та інші ресурси. Це не тільки забезпечує глибоку інтерактивність, але й розширює можливості для персоналізації навчання.

Однією з ключових особливостей запропонованої системи є її модульна архітектура, яка дозволяє легко адаптувати платформу до специфічних потреб різних навчальних закладів. Це, у свою чергу, відкриває нові перспективи для розвитку освітнього середовища, дозволяючи кожній установі індивідуалізувати свій навчальний процес.

Важливою частиною роботи є дослідження впливу системи на якість навчального процесу. Результати показали, що інтеграція сучасних технологічних рішень сприяє підвищенню ефективності управління навчанням та покращує зручність його використання. Це демонструє значний потенціал для покращення освітніх процесів за допомогою інтегрованих систем.

Дослідження також включає оцінку технічних аспектів розробки, зокрема використання високопродуктивних серверних процесів та реактивних фронтенд компонентів. Це дає можливість глибокого розуміння того, як технічні рішення впливають на загальну продуктивність та ефективність системи.

Висновки даної роботи можуть бути корисні не тільки для освітян, але й для розробників технологій, політиків та всіх зацікавлених у розвитку освітнього середовища. Це дослідження вказує на важливість інтеграції новітніх технологічних рішень у навчальний процес та підкреслює потенціал, який вони несуть для трансформації освіти.

# 1 ТЕОРЕТИЧНІ ОСНОВИ ІНТЕГРАЦІЇ АПАРАТНИХ ТА ПРОГРАМНИХ СИСТЕМ У ВЕБ-ЗАСТОСУНКИ

## 1.1. Огляд сучасного стану інтеграції систем в освіті

Сучасний світ освіти переживає епоху значних змін та інновацій, що в основному обумовлено стрімким розвитком інформаційних технологій. Ці технології відіграють не просто важливу, а ключову роль у формуванні та розвитку освітніх систем по всьому світу. Вони впливають на всі аспекти освіти – від методів викладання до управління освітніми установами, від створення навчальних матеріалів до способів їх доставки студентам.

Інтеграція апаратних та програмних систем у веб-застосунки в освітній сфері є одним із найважливіших напрямків розвитку сучасної освіти. Цей процес не тільки сприяє підвищенню ефективності навчання, але й відкриває нові можливості для персоналізації освітнього процесу, роблячи його більш доступним та інклюзивним. Інтеграція технологій у освіту не обмежується лише впровадженням нових інструментів; це також передбачає зміну підходів до навчання, розвиток нових педагогічних стратегій та адаптацію управлінських процесів.

В цьому розділі ми розглянемо сучасний стан інтеграції систем в освіті, аналізуючи як успіхи, так і виклики, з якими стикаються освітні установи. Ми дослідимо, як інтеграція технологій впливає на різні аспекти освітнього процесу, включаючи методи викладання, управління навчальними матеріалами, взаємодію між студентами та викладачами, а також на загальну організаційну структуру освітніх установ.

Особлива увага буде приділена тому, як інтеграція апаратних та програмних систем сприяє розвитку гнучких та адаптивних навчальних середовищ, які можуть задовольнити потреби різноманітного контингенту студентів. Ми також розглянемо виклики, пов'язані з цим процесом, включаючи питання безпеки даних, проблеми інтеграції з існуючими системами та необхідність постійного оновлення технологічної інфраструктури.



Крім того, ми оцінимо вплив цих інтеграційних процесів на педагогічні практики та освітні стратегії. Це включає аналіз того, як технології можуть сприяти більш ефективному та інтерактивному навчанню, а також розгляд можливостей для інновацій у педагогіці, які відкриваються завдяки цим технологіям.

Завершуючи, ми підведемо підсумки та визначимо основні напрямки подальших досліджень у цій області, з огляду на швидкі зміни в технологічному ландшафті та їх вплив на освітній процес.

### Значення інтеграції в сучасній освіті

Інтеграція апаратних та програмних систем у освітній процес є не просто тенденцією, а необхідністю сучасного освітнього середовища. Цей процес відіграє вирішальну роль у формуванні гнучких, адаптивних та інтерактивних методів навчання, які відповідають вимогам сучасного світу. Інтеграція технологій в освіту не тільки підвищує ефективність навчального процесу, але й сприяє розвитку критичного мислення, творчості та інших ключових навичок у студентів.

Важливість інтеграції систем у освіті полягає в тому, що вона дозволяє студентам отримувати знання в більш залучаючому та ефективному форматі. Завдяки інтеграції, освітні установи можуть пропонувати більш персоналізовані та гнучкі навчальні програми, які враховують індивідуальні особливості та потреби кожного студента. Це, в свою чергу, сприяє підвищенню мотивації та зацікавленості у навчанні.

Інтеграція технологій також відкриває доступ до нових форм навчання, таких як онлайн-курси, віртуальні лабораторії, інтерактивні симуляції та ігрові методики, які роблять процес навчання більш динамічним та захоплюючим. Це дозволяє студентам не тільки засвоювати теоретичні знання, але й розвивати практичні навички та вміння, які будуть корисними їм у майбутньому професійному житті.

Крім того, інтеграція апаратних та програмних систем сприяє розвитку цифрової грамотності серед студентів. У сучасному світі, де технології швидко розвиваються, володіння цифровими навичками є критично важливим. Студенти, які вчаться в середовищі, де активно використовуються різноманітні технології, стають більш підготовленими до роботи в сучасному цифровому світі.

Інтеграція систем у освіті також сприяє рівності у доступі до освіти. Завдяки онлайн-платформам та дистанційним технологіям, якісна освіта стає доступною для ширшого кола людей, незалежно від їх географічного розташування чи соціально-економічного статусу. Це відкриває нові можливості для людей, які раніше не мали доступу до вищої освіти, та сприяє розвитку інклюзивного освітнього середовища.

Враховуючи всі ці аспекти, можна сказати, що інтеграція апаратних та програмних систем у освітній процес є фундаментальним кроком у розвитку сучасної освіти. Вона не тільки підвищує якість та доступність освіти, але й готує студентів до ефективної роботи та життя у високотехнологічному суспільстві.

Історія інтеграції систем в освіті є свідченням еволюції технологій та їх впливу на освітній процес. Від перших кроків введення комп'ютерів у навчальні заклади до сучасних інноваційних технологій, цей шлях характеризується постійним розвитком та адаптацією до змінюваних потреб освіти.

Спочатку, введення комп'ютерів у навчальні заклади було спрямоване на навчання основам програмування та комп'ютерної грамотності. Це були прості комп'ютерні класи, де студенти могли здобувати базові навички роботи з комп'ютерною технікою. З часом, з появою інтернету та розвитком цифрових технологій, освітні установи почали інтегрувати більш складні системи. Електронні навчальні платформи, системи управління навчанням (LMS), інтерактивні дослідницькі інструменти стали частиною освітнього процесу.

Значний вплив на розвиток інтеграційних процесів у освіті мали технологічні інновації, такі як хмарні технології, мобільні додатки, штучний інтелект та великі дані. Ці технології дозволили створити більш гнучкі, адаптивні та персоналізовані системи навчання. Вони забезпечили можливість для студентів вчитися в умовах, які найкраще відповідають їхнім індивідуальним потребам та стилю навчання.

Сьогодні інтеграція систем у освіті охоплює широкий спектр технологій та методів. Вона включає в себе інтеграцію апаратного та програмного забезпечення, використання інтерактивних методів навчання, інтеграцію соціальних медіа та ігрових технологій у навчальний процес. Також важливим аспектом є створення персоналізованих навчальних траєкторій за допомогою аналітики даних та штучного інтелекту.

Завдяки цим інноваціям, освітній процес стає більш ефективним, залучаючим та відповідним до потреб сучасного світу. Студенти отримують можливість вчитися в більш інтерактивному та динамічному середовищі, що сприяє розвитку критичного мислення, креативності та навичок роботи з сучасними технологіями.

Інтеграція систем у освіті продовжує розвиватися, відкриваючи нові горизонти для навчання та викладання. Вона стає не тільки інструментом для підвищення якості освіти, але й засобом для підготовки студентів до життя та роботи у високотехнологічному суспільстві.

## 1.2. Аналіз існуючих підходів до інтеграції апаратних та програмних систем

Сучасні апаратні рішення в освіті представляють собою різноманітність технологій, які істотно впливають на методи та якість навчання. Ці рішення охоплюють широкий спектр пристроїв та обладнання, кожен з яких вносить свій вклад у формування сучасного освітнього середовища:

- Персональні комп'ютери та ноутбуки: Це основа багатьох освітніх систем. Вони забезпечують доступ до інтернету, освітніх програм та ресурсів, а також дозволяють студентам та викладачам виконувати різноманітні завдання, від обробки текстів до складних наукових розрахунків.
- Інтерактивні дошки: Ці пристрої перетворюють традиційне навчання, забезпечуючи більшу взаємодію та залученість учнів. Вони дозволяють викладачам використовувати мультимедійні матеріали, інтерактивні вправи та ігри, що робить навчання більш захоплюючим та ефективним.
- Планшети та смартфони: Ці мобільні пристрої розширюють можливості навчання за межами класної кімнати. Вони забезпечують студентам доступ до електронних книг, освітніх додатків та онлайн-курсів, а також дозволяють виконувати завдання та проекти в будь-якому місці.
- Віртуальна та доповнена реальність: Ці технології відкривають нові горизонти в освіті, дозволяючи студентам занурюватися в інтерактивні та реалістичні навчальні середовища. Вони можуть використовуватися для віртуальних екскурсій, наукових експериментів та імітації складних процесів.

- Спеціалізоване обладнання: До цієї категорії відносяться лабораторне обладнання, 3D-принтери, робототехніка та інші спеціалізовані пристрої, які використовуються для практичного навчання у різних наукових та технічних дисциплінах. Це обладнання дозволяє студентам отримувати практичний досвід та розвивати конкретні навички.
- Мережеве обладнання та інфраструктура: Це включає в себе Wi-Fi маршрутизатори, сервери та інші компоненти, які забезпечують підключення до інтернету та внутрішньої мережі освітнього закладу. Наявність надійної мережевої інфраструктури є критично важливою для підтримки всіх видів апаратних та програмних рішень.
- Безпека та зберігання даних: Системи безпеки, такі як антивірусне програмне забезпечення та файрволи, а також рішення для зберігання даних, включаючи хмарні сервіси та мережеві сховища, є важливими для захисту інформації та забезпечення безперебійної роботи освітніх систем.
- Адаптивність та масштабованість: Сучасні апаратні рішення повинні бути адаптивними та масштабованими, щоб відповідати зростаючим та змінюваним потребам освітнього середовища. Це означає можливість легко оновлювати та розширювати обладнання та інфраструктуру з мінімальними зусиллями.

Інтеграція цих апаратних рішень у освітній процес не тільки сприяє ефективності навчання, але й відкриває нові можливості для індивідуалізації та адаптації освітнього процесу до потреб кожного студента. Використання цих технологій дозволяє освітнім установам використовувати інтерактивні методи навчання, забезпечувати доступ до великої кількості навчальних ресурсів та підтримувати постійний зв'язок між студентами та викладачами.

Програмні системи є фундаментальною частиною сучасної освіти, відіграючи ключову роль у формуванні та розвитку освітніх процесів. Вони охоплюють широкий спектр платформ та додатків, кожен з яких вносить свій вклад у покращення та оптимізацію навчального досвіду.

- Системи управління навчанням (LMS): Платформи, такі як Moodle, Blackboard, або Canvas, є основою для організації курсів, відстеження прогресу студентів, проведення тестувань та управління навчальними ресурсами. Вони дозволяють викладачам створювати інтерактивні курси, включаючи відео, квізи, та дискусійні форуми, тим самим забезпечуючи гнучкість та доступність навчання.
- Програмне забезпечення для співпраці: Інструменти, такі як Google Classroom, Microsoft Teams, або Slack, сприяють ефективній комунікації та співпраці між студентами та викладачами. Вони дозволяють обмінюватися файлами, організовувати групові проекти, проводити відеоконференції та спільно працювати над документами в реальному часі.
- Електронні книги та освітні ресурси: Цифрові бібліотеки та онлайн-ресурси, такі як JSTOR, Google Scholar, або Khan Academy, надають студентам доступ до великої кількості навчальних матеріалів, наукових статей та інтерактивних курсів. Це сприяє самостійному навчанню та дослідженню.
- Інструменти для творчості та дизайну: Програми, такі як Adobe Creative Cloud, AutoCAD, або SketchUp, використовуються у курсах дизайну, архітектури, медіа та інших творчих дисциплінах. Вони дозволяють студентам розвивати та демонструвати свої творчі навички через професійні інструменти.

- Аналітичне та наукове програмне забезпечення: Програми, такі як MATLAB, SPSS, або R, використовуються для складних наукових розрахунків, статистичного аналізу та обробки даних. Вони є незамінними для студентів, які займаються науковими дослідженнями та аналітикою.
- Інструменти для програмування та розробки: Середовища розробки, такі як Visual Studio, Eclipse, або PyCharm, дозволяють студентам писати, тестувати та відлагоджувати код. Вони важливі для курсів з комп'ютерних наук, інженерії та програмування.
- Інструменти для віртуального навчання та симуляцій: Програми, які дозволяють створювати віртуальні лабораторії або симуляції, такі як Labster або PhET, забезпечують студентам можливість проводити експерименти та дослідження в безпечному та контрольованому середовищі.
- Інструменти для оцінювання та відстеження прогресу: Програмне забезпечення, яке дозволяє викладачам створювати тести, квізи та інші форми оцінювання, а також відстежувати прогрес студентів, є важливим для забезпечення об'єктивності та справедливості оцінювання.
- Інструменти для особистісного розвитку та самоосвіти: Програми, які допомагають студентам розвивати навички самоорганізації, управління часом, а також підтримують самоосвіту, такі як Duolingo, Coursera, або TED-Ed, відіграють важливу роль у формуванні всебічно розвиненої особистості.

Кожна з цих систем вносить свій вклад у створення гнучкого, інтерактивного та індивідуалізованого освітнього середовища, яке сприяє розвитку критичного мислення, креативності та навичок розв'язання проблем у студентів. Вони не тільки полегшують доступ до навчальних матеріалів, але й відкривають нові можливості для індивідуалізації навчання, забезпечуючи студентам необхідні інструменти для успішного навчання та професійного розвитку.

Веб-технології у сучасній освіті пропонують новітні підходи та методики навчання, які радикально змінюють традиційні освітні парадигми.

- Онлайн-платформи та курси: Платформи, такі як Coursera, Udey, або edX, пропонують широкий спектр онлайн-курсів на різноманітні теми, від базових академічних дисциплін до спеціалізованих професійних навичок. Ці курси часто включають відео-лекції, інтерактивні вправи, форуми для обговорень та онлайн-тести, забезпечуючи глибоке занурення в предмет.
- Інтерактивні вправи та симуляції: Веб-технології дозволяють створювати інтерактивні вправи та симуляції, які відтворюють реальні ситуації або експерименти. Це особливо корисно в наукових та технічних дисциплінах, де практичний досвід є ключовим.
- Відео-лекції та вебінари: Використання відео-лекцій та вебінарів дозволяє студентам вчитися у власному темпі, переглядаючи матеріали в зручний для них час. Це також сприяє більш широкому доступу до знань, оскільки студенти з будь-якої точки світу можуть долучитися до курсів.



- Інструменти для співпраці та комунікації: Веб-технології, такі як Slack, Zoom, або Google Meet, сприяють ефективній комунікації та співпраці між студентами та викладачами. Вони дозволяють проводити групові проекти, обговорення та консультації в онлайн-форматі.
- Віртуальні лабораторії та експерименти: Веб-технології дозволяють створювати віртуальні лабораторії, де студенти можуть проводити експерименти та дослідження без необхідності фізичної присутності в лабораторії. Це робить навчання більш доступним та безпечним.
- Персоналізоване навчання та адаптивні системи: Використання алгоритмів машинного навчання та штучного інтелекту дозволяє створювати персоналізовані навчальні траєкторії, які адаптуються до індивідуальних потреб та рівня знань кожного студента.
- Цифрова грамотність та навички: Веб-технології сприяють розвитку цифрової грамотності серед студентів, навчаючи їх ефективно використовувати різноманітні цифрові інструменти та ресурси для навчання, дослідження та спілкування.
- Відкритий доступ до освітніх ресурсів: Ініціативи, такі як Open Educational Resources (OER), надають вільний доступ до навчальних матеріалів, підручників та курсів, що сприяє демократизації освіти та рівному доступу до знань.
- Мобільне навчання та додатки: Розвиток мобільних технологій та додатків дозволяє студентам вчитися в будь-якому місці та в будь-який час, використовуючи смартфони та планшети для доступу до навчальних матеріалів та ресурсів.
- Інтеграція з соціальними медіа: Використання соціальних медіа в освітніх цілях, таких як створення

### 1.3. Принципи та методології інтеграції веб-застосунків

Сучасний світ веб-розробки пропонує різноманітні принципи та методології інтеграції, кожен з яких має свої особливості та сфери застосування. Важливо розуміти ці різні підходи для вибору найбільш ефективного рішення для конкретного проекту.

Мікросервіси - це підхід до розробки програмного забезпечення, який включає розбиття додатку на набір невеликих, незалежних сервісів, які спілкуються між собою за допомогою легкодоступних API. Цей підхід дозволяє командам працювати над різними частинами додатку незалежно, що пришвидшує розробку та спрощує управління змінами. Це забезпечує гнучкість, легке масштабування та високу надійність.

RESTful (Representational State Transfer) API є популярним методом інтеграції у веб-розробці. Цей підхід використовує HTTP запити для обміну даними між різними системами, що сприяє легкій інтеграції та високій сумісності між різними платформами.

Simple Object Access Protocol (SOAP) є більш формалізованим протоколом для обміну даними у розподіленій комп'ютерній мережі. Це забезпечує високий рівень безпеки та надійності, але може бути складнішим у реалізації порівняно з RESTful API.

GraphQL – це мова запитів для API та час виконання для виконання цих запитів з вашими даними. GraphQL надає повний та зрозумілий опис даних у вашому API, дає клієнтам можливість запитувати саме те, що їм потрібно, та спрощує еволюцію API з часом.

Webhooks дозволяють додаткам отримувати реальні відгуки про події, що відбуваються в інших додатках. Це корисно для створення реактивних систем, де один додаток може автоматично відповідати на зміни у іншому.

Інтеграційні платформи як сервіс (iPaaS) надають набір інструментів та сервісів для спрощення процесу інтеграції між різними хмарними та локальними сервісами. Це може включати інструменти для мапінгу даних, оркестрації робочих процесів тощо.

#### 1.4. Виклики та перспективи розвитку інтегрованих систем в освіті

У сучасному світі освіти, де інтеграція різноманітних технологій стає все більш важливою, освітні установи стикаються з рядом технічних та безпекових викликів (див. табл. 1.1). Однією з основних задач є забезпечення ефективної взаємодії та сумісності між різними цифровими інструментами та платформами. Це вимагає ретельного планування, координації та використання стандартів інтероперабельності.

Таблиця 1.1 – Основні виклики та перспективи

Виклики	Перспективи
Забезпечення сумісності технологій	Інноваційні технології (ШІ, машинне навчання)
Кібербезпека	Гнучкість та адаптивність систем
Технічна підтримка та обслуговування	Співпраця між установами

Зі зростанням залежності від цифрових технологій, збільшуються й ризики, пов'язані з кібербезпекою. Проблема кібербезпеки стає критичною, оскільки інституції повинні забезпечувати захист від кібератак, гарантувати конфіденційність та захист даних студентів та співробітників. Це вимагає інвестицій у захисні технології, підготовку персоналу та створення ефективних систем управління ризиками.

Технічна підтримка та обслуговування технологічного обладнання та програмного забезпечення також є важливими викликами. Необхідність у постійному оновленні та модернізації обладнання, а також у підтримці на високому рівні функціонування цифрових систем, вимагає значних ресурсів та спеціалізованих знань.

Щодо перспектив розвитку інтеграції в освіті, то вони включають активне використання новітніх технологій. Штучний інтелект, машинне навчання та великі дані відіграють ключову роль у радикальній трансформації освітнього процесу. Вони дозволяють персоналізувати навчання, забезпечуючи адаптивність та індивідуалізацію освітніх траєкторій студентів.

Розвиток гнучких та адаптивних навчальних систем, які можуть швидко реагувати на зміни у навчальному процесі, є ще однією важливою перспективою. Це забезпечує можливість швидко адаптувати освітні програми під потреби студентів та ринку праці.

Розвиток міжнародних та міжінституційних партнерств сприяє обміну досвідом, ресурсами та кращими практиками в галузі інтеграції технологій, відкриваючи нові можливості для співпраці та інновацій.

У контексті міжнародного досвіду та кращих практик, вивчення успішних моделей інтеграції технологій у освітній процес із різних країн стає надзвичайно корисним. Це включає участь у глобальних освітніх проектах та програмах, що забезпечують обмін знаннями та досвідом між установами та країнами.

Таким чином, аналіз викликів та перспектив розвитку інтегрованих систем в освіті вказує на важливість комплексного підходу. Це включає забезпечення технічної сумісності та безпеки, використання інноваційних технологій для персоналізації навчання, розвиток гнучких та адаптивних систем, а також активну участь у міжнародних освітніх проектах та програмах.

## **2 АНАЛІЗ ВИМОГ ТА РОЗРОБКА АРХІТЕКТУРИ СИСТЕМИ УПРАВЛІННЯ НАВЧАЛЬНИМ ПРОЦЕСОМ**

### **2.1 Визначення мети, об'єкта та задач проекту**

Проект зосереджений на створенні передової Learning Experience Platform (LXP) для університетів, спрямованої на трансформацію та підвищення ефективності навчального процесу. Метою є створення системи, яка не тільки відповідає сучасним освітнім вимогам, але й пропонує інноваційні рішення для персоналізації навчання та підвищення залученості студентів. Об'єктом проекту є комплексний навчальний процес у вищих навчальних закладах, який включає у себе управління курсами, інтерактивне навчання, оцінювання та адміністрування. Предметом роботи є процес інтеграції апаратних та програмних рішень у створення ефективної та інтерактивної освітньої платформи.

Задачі проекту охоплюють широкий спектр діяльності, від аналізу існуючих освітніх процесів до впровадження інноваційних технологій. Важливою частиною є розробка архітектури системи, яка повинна бути гнучкою, масштабованою та інтеграційно спроможною. Це включає в себе створення інтуїтивно зрозумілого інтерфейсу, адаптованого до потреб різних користувачів, від студентів до адміністраторів. Інтеграція з існуючими освітніми системами та базами даних є ключовою для забезпечення безперервності та ефективності навчального процесу. Крім того, проект передбачає впровадження штучного інтелекту для аналізу даних та персоналізації навчального досвіду, а також забезпечення високого рівня безпеки та приватності інформації.

## 2.2 Аналіз основних функціональних вимог

Аналіз функціональних вимог включає в себе детальне дослідження потреб усіх учасників освітнього процесу. Важливою частиною є збір та аналіз зворотного зв'язку від студентів, викладачів та адміністративного персоналу, що дозволяє виявити ключові аспекти, які потребують удосконалення. Основні функціональні вимоги охоплюють створення інтерактивних та гнучких навчальних модулів, які можуть бути адаптовані до індивідуальних освітніх потреб. Це включає в себе розробку інструментів для управління курсами, розкладами, оцінюванням та іншими адміністративними завданнями.

Інтеграція з існуючими освітніми системами та базами даних є критичною для забезпечення сумісності та ефективності. Система повинна бути здатна обробляти великі обсяги даних, забезпечувати стабільну роботу під час пікових навантажень та мати високий рівень доступності. Впровадження штучного інтелекту для аналізу даних та персоналізації навчального досвіду дозволить створити більш ефективну та залучену навчальну середу. Також важливим є забезпечення безпеки та конфіденційності інформації, що вимагає впровадження сучасних стандартів безпеки та регулярного оновлення системи.

Ці вимоги формують основу для розробки системи, яка відповідатиме потребам сучасного освітнього середовища та забезпечить високу якість навчання. Розробка такої системи вимагає комплексного підходу, який включає в себе технічну експертизу, глибоке розуміння освітніх процесів та вміння адаптувати інноваційні технології до потреб користувачів.

## 2.3 Аналіз потреб користувачів та стейкхолдерів

Розробка системи управління навчальним процесом (СУНП) вимагає детального розуміння та аналізу потреб усіх зацікавлених сторін (рис. 3.2). Цей процес не тільки включає ідентифікацію основних вимог, але й глибоке дослідження специфічних потреб різних груп користувачів, включаючи студентів, викладачів, адміністративний персонал, батьків, та інших користувачів системи.

### Ідентифікація стейкхолдерів



Рисунок 2.1 – Зацікавлені сторони

**Студенти:** Як основні користувачі системи, студенти мають унікальні потреби та очікування від СУНП.

**Викладачі:** Важливі учасники освітнього процесу, викладачі використовують систему для управління курсами та взаємодії зі студентами.

**Адміністративний персонал:** Відповідають за адміністративні аспекти навчального процесу, включаючи реєстрацію, звітність та управління даними.

**Батьки:** Зацікавлені в моніторингу успішності та прогресу своїх дітей.

**Інші користувачі:** Включають технічний персонал, партнерів освітнього закладу, та потенційних інвесторів. Аналіз потреб користувачів.

Аналіз потреб користувачів (див. рис. 2.2).



Рисунок 2.2 – Потреби стейкхолдерів

Студенти: Потребують індивідуалізації навчання, інтерактивних елементів, гейміфікації та мобільного доступу до навчальних матеріалів. Важливою є можливість адаптації навчального процесу під особисті потреби та стиль навчання.

Викладачі: Необхідні інструменти для ефективного управління курсами, аналітики та звітності. Важливим є наявність засобів для забезпечення зворотного зв'язку та спілкування зі студентами.

Адміністративний персонал: Потребують автоматизації рутинних адміністративних завдань, ефективних систем управління даними та забезпечення безпеки персональних даних.

Батьки: Хочуть мати доступ до інформації про академічний прогрес та відвідуваність своїх дітей, а також можливість спілкування з викладачами.

Інші користувачі: Технічний персонал зацікавлений у надійності системи та доступності технічної підтримки. Партнери та інвестори можуть бути зацікавлені в інтеграції з іншими системами та масштабованості СУНП.



Методи збору інформації (див. рис. 2.3).



Рисунок 2.3 – Методи збору інформації

- Опитування та анкетування: Збір даних про потреби та вподобання користувачів через опитування.
- Інтерв'ю: Проведення глибинних інтерв'ю з представниками різних груп стейкхолдерів.
- Фокус-групи: Організація дискусійних зустрічей з ключовими користувачами для детального аналізу їхніх потреб.
- Аналіз вторинних даних: Вивчення існуючих досліджень, звітів, та статистики.

Аналіз потреб користувачів та стейкхолдерів є важливим кроком у розробці ефективної СУНП. Це дозволяє створити систему, яка відповідає реальним потребам усіх зацікавлених сторін, забезпечуючи високу якість та ефективність навчального процесу.

## 2.4. Моделювання процесів управління навчальним процесом

Моделювання процесів управління навчальним процесом в контексті створення LXP для університетів є вирішальним етапом, що задає вектор для всієї подальшої розробки системи. У цій роботі ми прагнемо досягти глибокого розуміння та інтеграції різноманітних освітніх процесів, забезпечуючи при цьому необхідну гнучкість і адаптивність системи до широкого спектру вимог та потреб сучасних університетів.

Починаючи з аналізу існуючих процесів управління навчальним процесом (див. рис. 2.4), ми зосереджуємо увагу на кількох ключових аспектах. Планування навчальних програм передбачає розробку інтерактивних і гнучких навчальних планів, що інтегрують мультимедійні матеріали та інтерактивні засоби навчання, а також адаптацію курсів до змінних вимог ринку праці та академічних стандартів.



Рисунок 2.4 – Існуючі навчальні процеси

Особлива увага приділяється процесу реєстрації та запису на курси, де ми розробляємо модульну систему, що інтегрується з персоналізованими профілями студентів та включає систему рекомендацій для курсів, базуючись на академічних інтересах та історії успіхів студента.

Управління навчальними матеріалами передбачає централізацію та організацію ресурсів для забезпечення легкого доступу, а також інтеграцію з зовнішніми освітніми ресурсами та платформами. Важливим елементом є також ведення та моніторинг відвідуваності, який передбачає автоматизацію процесів контролю за допомогою інтегрованих мобільних додатків та використання аналітичних інструментів.

Оцінювання та зворотний зв'язок включає розробку модульних систем оцінювання та інтеграцію інструментів для збору та аналізу зворотного зв'язку від студентів. Аналіз успішності студентів та генерація звітів для адміністрації та викладачів є частиною звітності та аналітики.

У процесі моделювання ми використовуємо BPMN для стандартизації візуалізації процесів управління навчальним процесом та розробки діаграм потоків даних. Сценарії використання розробляються з урахуванням різних потреб користувачів, включаючи студентів, викладачів та адміністрацію (рис. 2.5).



Рисунок 2.5 – Методологія моделювання

Інтеграція з іншими системами, інформаційні системи університетів та зовнішні сервіси, є необхідною для забезпечення широких можливостей та гнучкості LXP. Пілотне тестування, збір зворотного зв'язку та ітеративні покращення забезпечують оптимізацію процесів на основі аналізу отриманих даних.

Загалом, моделювання процесів управління навчальним процесом є ключовим елементом у розробці ефективної, гнучкої та адаптивної системи LXP для університетів, забезпечуючи при цьому високу якість освітнього процесу та задоволення потреб сучасних студентів та викладачів.

## 2.5. Проектування архітектури системи

Розробка архітектури LXP для університетів вимагає інтеграції сучасних підходів та кращих практик у сфері програмного забезпечення. Ми використовуємо принципи Domain-Driven Design (DDD) та Command Query Responsibility Segregation (CQRS) для створення гнучкої, масштабованої та ефективної системи, яка відповідає потребам сучасних освітніх установ.

У процесі розробки архітектури нашої LXP, ми акцентуємо увагу на декількох ключових принципах, які забезпечують гнучкість, ефективність та легкість управління системою:

– Модульність

Ми розбиваємо систему на незалежні модулі, що полегшує розробку, тестування та подальшу підтримку. Кожен модуль відповідає за свою частину функціональності та може розвиватися автономно.

– Мікросервісна архітектура

Використання мікросервісів дозволяє забезпечити високу доступність та легкість масштабування. Кожен мікросервіс є самостійним і може бути розгорнутий незалежно від інших.

– Інтеграція з зовнішніми сервісами

Система повинна підтримувати інтеграцію з різноманітними зовнішніми API та сервісами, що розширює її функціональні можливості та забезпечує більшу гнучкість.

Використання сучасних підходів

**Domain-Driven Design (DDD):** Ми використовуємо DDD для визначення областей бізнесу (Bounded Contexts) та створення багатoshарових моделей для кожної бізнес-області. Це дозволяє нам краще управляти складністю бізнес-правил та забезпечити високий рівень адаптації системи до потреб користувачів.

**Command Query Responsibility Segregation (CQRS):** Застосування CQRS дозволяє нам розділити логіку обробки запитів на читання та запис. Це забезпечує більшу оптимізацію продуктивності та гнучкість у виборі технологій для різних типів запитів.

У процесі розробки архітектури LXP, ми зосереджуємося на декількох критично важливих аспектах (див. рис. 2.6), які забезпечують ефективність та надійність системи.

**Масштабованість та Висока  
Доступність**



**Безпека**



**Інтеграція з Іншими Системами**



Рисунок 2.6 – Аспекти ефективності та надійності системи

Система повинна забезпечувати горизонтальне масштабування та балансування навантаження між серверами та сервісами. Це дозволяє системі легко адаптуватися до змін у навантаженні та забезпечувати високу доступність послуг.

Шифрування даних, аутентифікація, авторизація та захист від загроз є ключовими елементами нашої системи. Ми використовуємо надійні механізми для забезпечення конфіденційності та цілісності даних.

API-орієнтований підхід, веб-хуки, події та стандартизація даних дозволяють нашій системі легко інтегруватися з іншими системами, забезпечуючи широкий спектр можливостей.

## – Тестування та Моніторинг

Автоматизоване тестування, моніторинг стану системи та її компонентів, логування та аналітика є важливою частиною нашого процесу розробки. Вони дозволяють нам вчасно виявляти та вирішувати проблеми, забезпечуючи високу якість та надійність системи.

Підводячи підсумок, розробка архітектури LXP, що базується на сучасних підходах та кращих практиках, є ключовим фактором успіху системи. Використання DDD та CQRS, забезпечення масштабованості, високої доступності, безпеки, та інтеграції з іншими системами, а також ефективне тестування та моніторинг, дозволяють створити потужну, гнучку та надійну платформу для управління навчальним процесом.

### 2.6. Вибір технологічного стеку для реалізації платформи

У контексті магістерської роботи "Дослідження методів інтеграції апаратних та програмних систем у веб-застосунки", вибір технологічного стеку для розробки інноваційної платформи управління навчальним процесом відіграє ключову роль. Основна увага приділяється використанню Symfony для бекенду, Vue.js для фронтенду та Go для оптимізації високонавантажених серверних процесів.

## – Backend

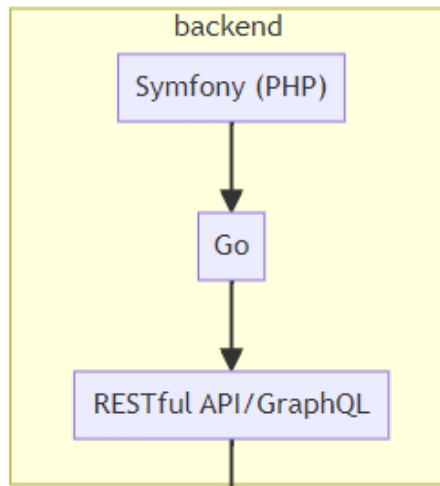


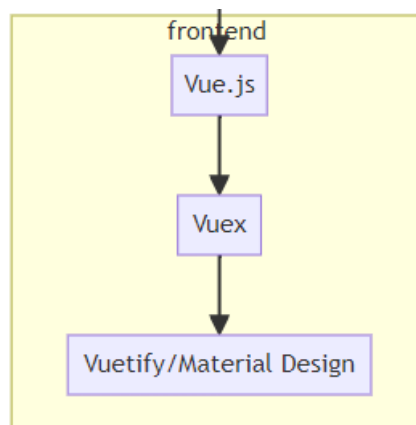
Рисунок 2.7 – Backend технології

**Symfony (PHP):** Основа бекенду, Symfony забезпечує модульну архітектуру, що дозволяє легко інтегрувати різноманітні компоненти та сервіси. Його використання сприяє підвищенню продуктивності, забезпечуючи гнучкість та масштабованість системи.

**Go:** Використання мови Go для розробки окремих мікросервісів або компонентів, які вимагають високої продуктивності та ефективності обробки великих обсягів даних.

**RESTful API/GraphQL:** Створення API для забезпечення взаємодії між фронтендом та бекендом, використовуючи REST або GraphQL для гнучкості та ефективності.

– Frontend





## Рисунок 2.8 – Frontend технології

Vue.js: Вибір Vue.js для фронтенду забезпечує створення інтуїтивно зрозумілих, динамічних користувацьких інтерфейсів. Його легкість та гнучкість ідеально підходять для реалізації інтерактивних елементів платформи.

Vuex для управління станом: Використання Vuex у поєднанні з Vue.js для ефективного управління станом додатку.

Vuetify/Material Design: Використання готових компонентів Vuetify для швидкої та зручної верстки, заснованої на принципах Material Design.

– Бази даних

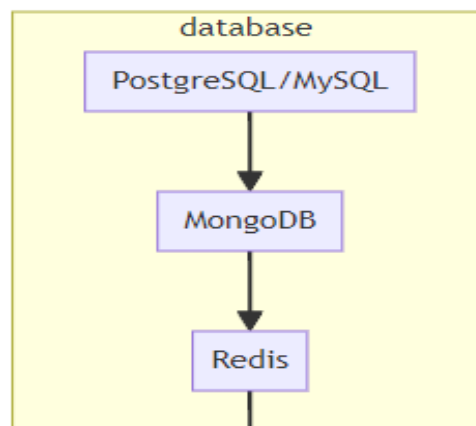


Рисунок 2.9 – Технології для баз даних

PostgreSQL/MySQL: Реляційні бази даних для зберігання структурованих даних, інтегрованих з Symfony.

MongoDB: Документо-орієнтована база даних для гнучкого зберігання даних, які не вимагають строгих схем.

Redis: Використання Redis для кешування та зберігання сесій, що підвищує швидкість відгуку системи.

– Мікросервісна архітектура та інтеграція

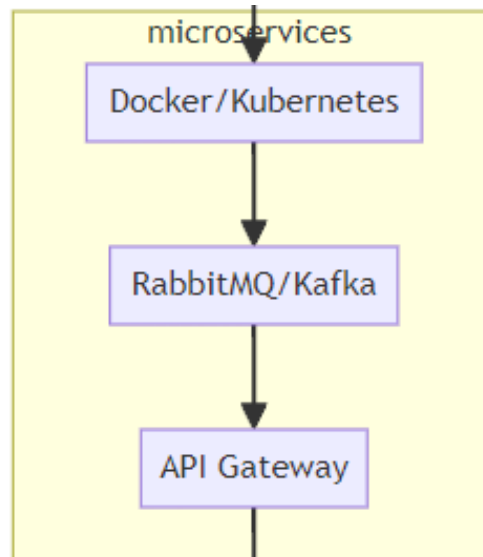


Рисунок 2.10 – Технології архітектури та інтеграції

**Docker/Kubernetes:** Контейнеризація та оркестрація мікросервісів з використанням Docker та Kubernetes для забезпечення масштабованості та високої доступності.

**RabbitMQ/Kafka:** Системи для обробки потокових даних та асинхронної комунікації між сервісами.

**API Gateway:** Централізоване управління API для маршрутизації запитів та забезпечення безпеки.

- Безпека

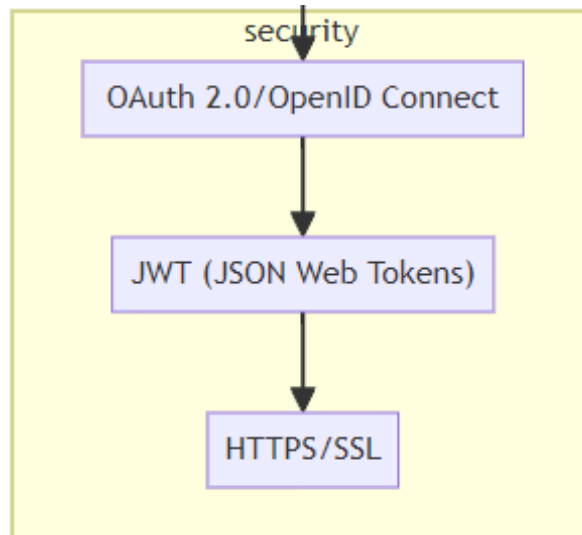


Рисунок 2.11 – Технології безпеки

OAuth 2.0/OpenID Connect: Стандарти для аутентифікації та авторизації користувачів.

JWT (JSON Web Tokens): Технологія для безпечної передачі даних між сторонами.

HTTPS/SSL: Забезпечення безпечного з'єднання та шифрування даних.

При розробці інноваційної платформи для управління навчальним процесом, важливо розглянути альтернативні технологічні стеки та зрозуміти, чому обраний нами стек (Symfony, Vue.js, Go) є оптимальним вибором (див. рис. 2.12).

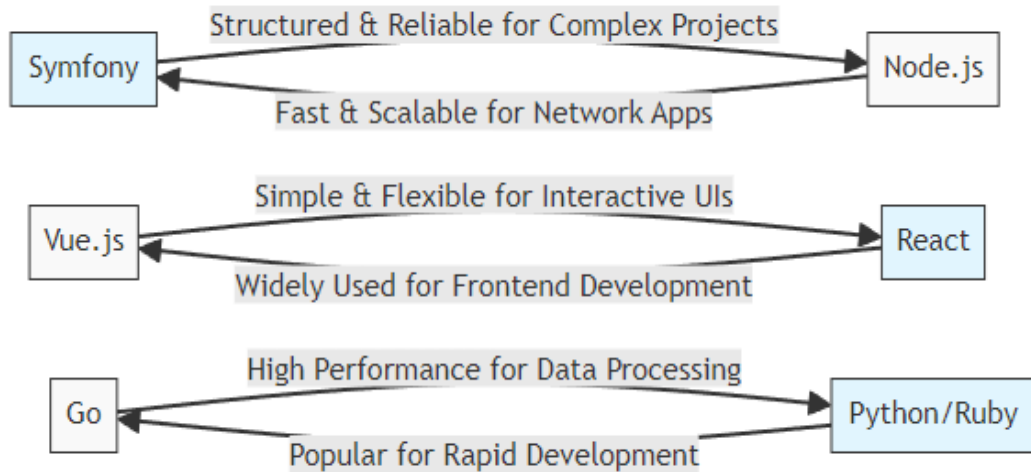


Рисунок 2.12 – Порівняння технологічних стеків

У процесі проектування архітектури системи LXP для університетів ми зіткнулися з викликом вибору технологічного стеку, який би найкраще відповідав нашим цілям та вимогам. Вибір між Symfony та Node.js, Vue.js проти React, а також Go в порівнянні з Python/Ruby був заснований на ряді ключових факторів, таких як продуктивність, масштабованість, гнучкість, інтеграція, безпека та надійність.

#### Symfony vs Node.js:

Ми обрали Symfony замість Node.js, оскільки він пропонує більшу структурованість та надійність для великих та складних проектів. Symfony забезпечує більш зрілу екосистему та ефективнішу підтримку для розробки модульних та масштабованих веб-додатків. Такий вибір виявився особливо важливим у контексті розробки складної системи управління навчальним процесом, де Symfony забезпечив більшу стабільність та легшість у підтримці коду.

#### Vue.js vs React:

Хоча React є популярним вибором у розробці фронтенду, ми вирішили використати Vue.js через його простоту та гнучкість, що ідеально підходить для швидкої розробки інтерактивних інтерфейсів. Vue.js дозволяє реалізувати динамічні користувацькі інтерфейси з меншим обсягом коду порівняно з React, що є вирішальним для платформи з високим рівнем інтерактивності.

#### Go vs Python/Ruby:

Вибір Go замість Python або Ruby був обумовлений потребою в високій продуктивності та ефективності обробки великих обсягів даних. Go особливо корисний для розробки мікросервісів, які вимагають високої продуктивності та легкої масштабованості. Це важливо для обробки великих даних та забезпечення високої продуктивності серверних процесів нашої LXP системи.

#### Переваги Обраного Технологічного Стеку:

##### – Масштабованість та Продуктивність

Використання Symfony та Go дозволяє нам створити систему, що легко масштабується та забезпечує високу продуктивність. Це критично важливо для обробки великих обсягів даних та великої кількості одночасних користувачів.

##### – Гнучкість та Швидкість Розробки:

Vue.js сприяє швидкій та гнучкій розробці фронтенду, дозволяючи нам швидко адаптуватися до змінних вимог та потреб користувачів.

– Інтеграція та Сумісність:

Обраний стек забезпечує легку інтеграцію з різноманітними зовнішніми сервісами та платформами, що є ключовим для створення комплексної системи управління навчальним процесом.

– Безпека та Надійність:

Symfony та Go відомі своєю високою безпекою та надійністю, що є критично важливим для захисту конфіденційності даних та стабільності роботи системи.

Обраний технологічний стек (Symfony, Vue.js, Go) не тільки вирішує існуючі виклики традиційних систем, але й забезпечує високий рівень інтерактивності та персоналізації навчання, роблячи нашу LXP систему інноваційною та ефективною. Обраний технологічний стек (Symfony, Vue.js, Go) забезпечує оптимальне поєднання продуктивності, масштабованості, гнучкості та безпеки, що є важливим для створення ефективної та надійної платформи управління навчальним процесом. Цей стек дозволяє не тільки вирішити існуючі проблеми традиційних систем, але й забезпечити високий рівень інтерактивності та персоналізації навчання.

## 2.. Розробка модулів інтеграції апаратних та програмних рішень

У рамках розробки нашої передової LXP, ми зосередили увагу на створенні шести ключових модулів, кожен з яких відіграє вирішальну роль у функціонуванні системи та забезпечує її комплексність та інтегрованість. Ці модулі разом формують міцну основу для ефективного управління навчальним процесом у сучасному освітньому середовищі (рис. 2.13).



Рисунок 2.13 – Основні модулі освітнього середовища

### 1. Модуль Управління Користувачами

Роль: Цей модуль відповідає за реєстрацію, аутентифікацію, та управління профілями користувачів (студентів, викладачів, адміністраторів).

Інтеграція: Цей модуль можливо буде об'єднати з існуючими базами даних університетів через API, що дозволяє синхронізувати інформацію про користувачів та їх ролі в освітньому процесі.

### 2. Модуль Курсів та Навчальних Матеріалів

Роль: Модуль забезпечує створення, редагування, та управління курсами та навчальними матеріалами (рис. 2.14).

Інтеграція: Цей модуль буде інтегровано з внутрішніми ресурсами університету та зовнішніми освітніми платформами для забезпечення доступу до різноманітних навчальних ресурсів.

### 3. Модуль Планування та Розкладу

Роль: Відповідає за планування академічних курсів, складання розкладу занять, та адміністрування академічних подій.

Інтеграція: Інтегрований з календарями та планувальниками, що дозволяє автоматично синхронізувати академічні події з особистими календарями користувачів.

### 4. Модуль Оцінювання та Відгуків

Роль: Забезпечує механізми для оцінювання студентів, збору зворотного зв'язку, та аналітики успішності.

Інтеграція: Цей модуль інтегрований з системами оцінювання та аналітики, що дозволяє викладачам та адміністрації отримувати детальний огляд академічних досягнень студентів.

### 5. Модуль Комунікацій

Роль: Відповідає за забезпечення комунікації між учасниками навчального процесу.

Інтеграція: Інтегровано з інструментами електронної пошти, месенжерами, та форумами для забезпечення ефективного обміну інформацією та співпраці.

### 6. Модуль Безпеки та Шифрування

Роль: Забезпечує захист даних, конфіденційність інформації, та цілісність системи.

Інтеграція: Інтеграція з сучасними методами шифрування та системами безпеки забезпечує високий рівень захисту персональних та навчальних даних.



**Courses Statistics**

Search:  2 selected 29.10.2022 ...

Course	Enrollments	Total Progress	Completed	Certificates Issued	Income
Profession UX UI Designer	41	86%	31	29	\$3500
JavaScript	2	100%	1	1	\$500
Business English	200	50%	110	100	\$5.000

Рисунок 2.13 – Інтерфейс системи

Кожен модуль, починаючи від управління користувачами до безпеки та шифрування, був розроблений з метою максимальної ефективності та інтеграції, враховуючи специфічні потреби освітнього середовища. Інтеграція кожного модуля з зовнішніми та внутрішніми системами, використання передових методів шифрування та безпеки, а також забезпечення гнучкості та масштабованості системи є ключовими елементами, що дозволяють нашій LXP системі бути не тільки функціональною, але й інноваційною та адаптивною до потреб сучасного освітнього середовища.

## **3 ІНТЕГРАЦІЯ АПАРАТНИХ ТА ПРОГРАМНИХ СИСТЕМ ТА ЇХ ТЕСТУВАННЯ**

### **3.1. Методи інтеграції апаратних засобів**

Інтеграція апаратних засобів у систему управління навчальним процесом вимагає ретельного підходу та використання передових методів. Цей процес включає в себе вибір відповідного обладнання, розробку інтерфейсів для взаємодії з програмним забезпеченням та забезпечення стабільної та ефективної роботи інтегрованих систем. Розглянемо докладніше кожен з методів інтеграції апаратних засобів в контексті системи LXP.

Безпроводна інтеграція, яка включає використання технологій, таких як Wi-Fi, Bluetooth, та NFC (Near Field Communication), є важливою частиною сучасних технологічних екосистем, особливо у контексті інтеграції мобільних пристроїв, носимих гаджетів та інтерактивних дошок. Цей метод дозволяє створити гнучке та мобільне з'єднання між різними пристроями без необхідності фізичних кабелів, що є ідеальним рішенням для розподілених та динамічних середовищ, які зустрічаються у навчальних заходах та корпоративних налаштуваннях.

Одним з ключових переваг безпроводної інтеграції є її гнучкість. Технології, такі як Wi-Fi, дозволяють підключення без прив'язки до фізичного розташування, що сприяє легкій організації робочих місць або навчальних зон. Bluetooth забезпечує зручність підключення носимих пристроїв, таких як смарт-годинники та фітнес-трекери, до основної системи, а NFC може бути використано для швидкої та безпечної аутентифікації користувачів у системі.

Однак, безпроводна інтеграція також має свої виклики. Найважливішим з них є потреба у стабільному та надійному бездротовому з'єднанні, оскільки перерви у з'єднанні можуть призвести до переривання важливих процесів. Крім того, питання безпеки та конфіденційності даних є критичними, особливо у випадку передачі чутливої інформації через бездротові мережі. Забезпечення безпеки даних вимагає комплексного підходу, що включає застосування сучасних методів шифрування, безпечного аутентифікаційного протоколу та регулярного моніторингу мережевої активності.

Інтеграція через інтерфейсні шини та порти, зокрема через такі стандартні порти як USB і HDMI, є фундаментальною технікою у сучасному світі технологій. Цей метод дозволяє з'єднувати зовнішні апаратні пристрої з основною системою, забезпечуючи надійне фізичне з'єднання та високу швидкість передачі даних, що є критично важливим для багатьох застосувань, від простих периферійних пристроїв до складних мультимедійних систем.

USB-порти, завдяки своїй універсальності та широкій підтримці, стали основою для підключення великої кількості пристроїв, включаючи зовнішні накопичувачі, клавіатури, миші та багато інших периферійних пристроїв. Це забезпечує легкість у використанні та сумісність з різними пристроями. HDMI, з іншого боку, є стандартом для передачі відео та аудіо сигналів, що дозволяє з'єднувати комп'ютери, монітори, телевізори та інші мультимедійні пристрої, забезпечуючи високоякісну передачу контенту.

Основним недоліком є обмежена мобільність через фізичні кабелі, які можуть ускладнювати розташування обладнання та його використання у гнучких або динамічних середовищах. Крім того, виникають питання сумісності із різними стандартами і версіями портів, особливо коли мова йде про інтеграцію старих та нових пристроїв, що може вимагати використання адаптерів або конвертерів. Мережева інтеграція, що базується на використанні мережевих протоколів, таких як TCP/IP, є одним з найбільш фундаментальних підходів у сучасних інформаційних системах. Цей метод дозволяє з'єднати різні пристрої, як-от комп'ютери, сервери, та периферійні пристрої, у єдину мережу, забезпечуючи їх спільну роботу та обмін даними. Основною перевагою мережевої інтеграції є її універсальність і масштабованість, що дозволяє підключати та керувати пристроями в різних локаціях.

Однією з ключових особливостей мережевої інтеграції є можливість дистанційного доступу та управління. Це означає, що системні адміністратори можуть віддалено налаштовувати пристрої, оновлювати програмне забезпечення, та здійснювати моніторинг стану мережі. Для бізнесу та освітніх установ це відкриває широкі можливості для оптимізації робочих процесів та підвищення ефективності управління ресурсами.

Мережева інтеграція стикається з низкою викликів, особливо що стосується безпеки мережі. З огляду на зростаючу кількість кібератак, забезпечення безпеки даних та систем в мережі є пріоритетним завданням. Це включає захист від несанкціонованого доступу, шифрування даних та регулярне оновлення захисних механізмів.

Іншим важливим аспектом є комплексність налаштування мережевих параметрів. Правильна конфігурація мережі, включаючи налаштування маршрутизаторів, комутаторів та інших мережевих пристроїв, є ключовою для забезпечення стабільності та високої продуктивності мережі. Це вимагає глибоких знань у сфері мережевих технологій та системного адміністрування.

API-базована інтеграція, яка передбачає використання програмних інтерфейсів додатків (APIs), сьогодні є одним з ключових елементів у створенні гнучких і масштабованих систем. Цей метод дозволяє ефективно інтегрувати різні апаратні та програмні компоненти, створюючи єдину функціональну систему. Основною перевагою API-інтеграції є її висока гнучкість, оскільки вона дозволяє розробникам створювати взаємодії між різними системами та платформами, що значно розширює можливості застосування інтегрованих рішень.

Завдяки API, можна легко з'єднати програмне забезпечення з різноманітним апаратним обладнанням, надаючи можливість контролювати фізичні пристрої через програмні інтерфейси. Це особливо важливо в таких галузях, як автоматизація процесів, інтелектуальне управління будівлями, та розумне виробництво, де потрібна швидка реакція та адаптація до змін у робочому середовищі.

Проте, API-базована інтеграція також стикається з певними викликами. Одним із основних є потреба у розробці та підтримці API, що може вимагати значних ресурсів, особливо у випадку великих та складних систем. Додатково, питання контролю доступу та безпеки інтерфейсів є критичними, оскільки необхідно забезпечити захист від несанкціонованого доступу та можливих зловмисних атак.

Узагалі, API-базована інтеграція відіграє фундаментальну роль у сучасній цифровій інфраструктурі, надаючи можливість для створення більш гнучких, інтерактивних та інтегрованих рішень. Її ефективне використання може значно підвищити якість та ефективність робочих процесів, відкриваючи нові горизонти для інновацій та розвитку.

Інтеграція через хмарні сервіси, яка забезпечує синхронізацію даних між різними пристроями та системами, стає все більш популярною в сучасному цифровому світі. Цей метод використовує потужності хмарних обчислень для створення масштабованих та гнучких рішень, які можуть адаптуватися до змінних потреб бізнесу та користувачів. Однією з ключових переваг хмарної інтеграції є централізоване управління даними, яке дозволяє користувачам легко доступатися до необхідної інформації та ресурсів з будь-якого місця та на будь-якому пристрої.

Використання хмарних сервісів для інтеграції має істотні переваги у вигляді гнучкості та масштабованості. Це означає, що системи можуть бути швидко адаптовані до зростаючих потреб, наприклад, шляхом додавання додаткового місця для зберігання даних або обчислювальних ресурсів. Такий підхід ідеально підходить для бізнес-систем, які вимагають високої гнучкості та можливості швидкого масштабування.

Основними проблемами та викликами для цього методу будуть якість інтернет-з'єднання та безпека.

Модульна інтеграція, заснована на використанні модульних систем та стандартів, стає все більш поширеною у сучасних технологічних рішеннях. Основна ідея цього методу полягає у створенні систем, де окремі компоненти або модулі можуть легко замінюватися або оновлюватися без необхідності переробки всієї системи. Це досягається завдяки стандартизації компонентів, що дозволяє різним виробникам виробляти взаємозамінні частини.

Однією з основних переваг модульної інтеграції є легкість заміни та оновлення компонентів. Наприклад, у комп'ютерному обладнанні це може означати можливість легкої заміни процесора, оперативної пам'яті або накопичувачів без необхідності заміни всієї материнської плати або корпусу. Такий підхід не лише знижує витрати на оновлення та обслуговування обладнання, але й сприяє сталому використанню ресурсів, оскільки зменшує необхідність у повній заміні пристроїв.

Стандартизація компонентів відіграє ключову роль у цьому процесі. Вона дозволяє компаніям розробляти продукти, які можуть легко інтегруватися з іншими пристроями або компонентами, не зважаючи на виробника. Це створює більш гнучкі та адаптивні системи, які можуть відповідати змінним потребам користувачів.

Необхідність узгодження стандартів інтерфейсів між різними виробниками – один з основних викликів модульної інтеграції. Несумісність стандартів може обмежити вибір компонентів або знизити ефективність інтеграції. Крім того, стандартизація може накласти обмеження на інновації, оскільки компаніям може бути складно відступити від усталених стандартів для впровадження нових технологій.

Інтеграція апаратних засобів у систему управління навчальним процесом є складним, але важливим завданням. Використання передових методів інтеграції, як показано на прикладах реальних кейсів, дозволяє створити ефективну та гнучку систему, яка відповідає потребам сучасного освітнього середовища.

### 3.2. Інтеграція програмних рішень та API

Для навчальної платформи LXP, що базується на Symfony, інтеграція програмних рішень та API є в край необхідною для реалізації ефективного та інноваційного навчального середовища. Вона охоплює широкий спектр сценаріїв взаємодії, які значно підвищують можливості платформи, забезпечуючи гнучкість та зручність для користувачів.

#### – Інтеграція систем відеоконференцій

Використання платформ для відеоконференцій, як-от Zoom або Microsoft Teams, є ключовим для сучасного дистанційного навчання. Інтеграція цих інструментів через API, розроблені на Symfony, дозволяє нам легко організувати відеозустрічі, розподіляти запрошення та збирати важливу інформацію про участь і взаємодію студентів (див. рис. 3.1). Це забезпечує гладке впровадження відеоконференцій у навчальний процес, підвищуючи його ефективність.

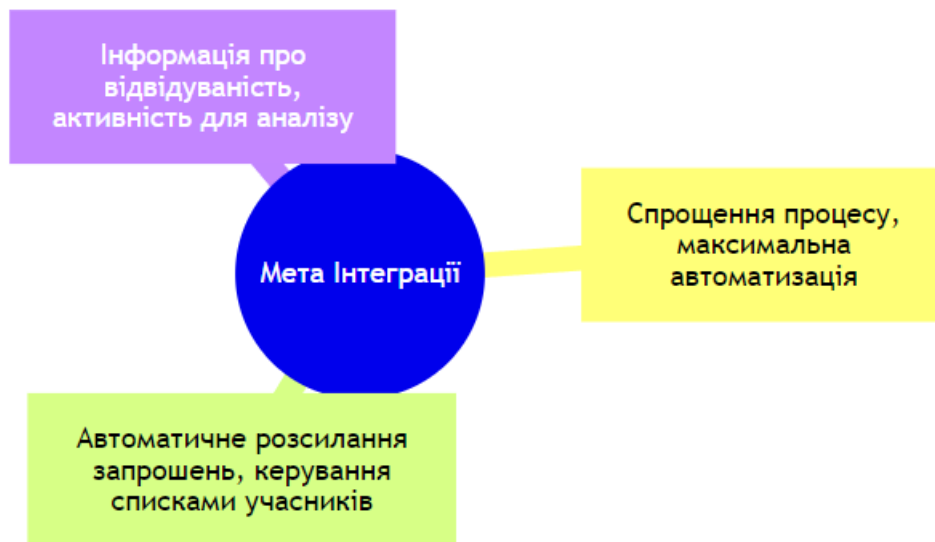


Рисунок 3.1 – Мета інтеграції систем відеоконференцій



Розглянемо процес інтеграції більш детально та виділемо ключові етапи інтеграції.

Вибір API для Інтеграції: Оскільки наша система розроблена на Symfony, першим кроком є вибір відповідних API від Zoom та Microsoft Teams, які дозволяють інтегрувати їх функціонал в нашу платформу.

Розробка Модуля Інтеграції: Далі наша команда розробляє спеціалізований модуль на Symfony, який використовуватиме ці API для створення та управління відеоконференціями. Модуль буде включати функції для автоматичного створення відеозустрічей на основі навчального розкладу, відправлення автоматичних запрошень учасникам та збору статистики участі.

Інтеграція з Календарем та Управлінням Курсами: Модуль буде інтегрований з календарем навчальних заходів та системою управління курсами, щоб забезпечити автоматичне планування та відстеження відеоконференцій.

Тестування та Валідація: Перед запуском модуль проходитиме ретельне тестування, щоб забезпечити його стабільну та безпечну роботу.

Впровадження та Навчання Користувачів: Після успішного тестування модуль впроваджується у загальну систему. Викладачі та студенти отримують інструкції та підтримку щодо використання нових функцій.

Ця інтеграція має на меті значно полегшити процес проведення відеоконференцій, зробивши їх більш доступними та зручними для викладачів та студентів. Автоматизація рутинних процесів та збір даних про відвідуваність дозволить підвищити ефективність навчання та забезпечити більшу залученість студентів.

Інтеграція Інтерактивних Дошок'

Інтеграція інтерактивних дошок, таких як Smart Board, відіграє важливу роль у залученні студентів і підвищенні взаємодії в класі. Реалізація модулів на Symfony для інтеграції цих технологій дозволяє ефективно інтегрувати інтерактивний контент у навчальний процес, забезпечуючи більш динамічне та залучаюче навчання.

#### Інтеграція з Хмарними Сервісами

Використання хмарних сервісів, як-от Google Drive або Microsoft OneDrive, для зберігання та спільної роботи над навчальними матеріалами є важливим для сучасного навчального процесу. Інтеграція цих платформ забезпечує легкий доступ до матеріалів, підвищуючи гнучкість та співпрацю між студентами та викладачами.

#### Інтеграція з Мобільними Пристроями

Забезпечення сумісності нашої LXP з мобільними пристроями є критично важливим для зручності користувачів. Розробка API на Symfony для мобільних додатків дозволяє забезпечити безперебійну синхронізацію даних та оптимізацію інтерфейсів, роблячи навчання доступним та зручним незалежно від місця та часу.

Процес інтеграції охоплює аналіз та планування для визначення ключових потреб, розробку та налаштування API, тестування для забезпечення стабільності та безпеки, а також впровадження та постійний моніторинг інтегрованих рішень.

#### 3.3. Забезпечення безпеки при інтеграції систем

У процесі інтеграції систем на нашій освітній платформі LXP, розробленій на базі Symfony, ми приділяємо особливу увагу питанням безпеки, оскільки обробка конфіденційних даних студентів та викладачів вимагає високого рівня захисту. Наш підхід до забезпечення безпеки охоплює кілька ключових аспектів:

- Багаторівнева аутентифікація (MFA): Введення MFA забезпечує додатковий рівень захисту, вимагаючи від користувачів підтвердження особи через кілька методів, таких як пароль та SMS-код.
- OAuth 2.0 та OpenID Connect: Використання цих стандартів гарантує безпечну аутентифікацію та авторизацію, дозволяючи безпечно інтегрувати нашу систему з обліковими записами університету.
- Шифрування даних у спокої та в русі: Всі дані, що зберігаються на серверах LXP, а також ті, що передаються, захищені шифруванням, з використанням протоколу TLS для передачі.
- Використання сучасних алгоритмів шифрування: Наша система застосовує надійні алгоритми, такі як AES-256, для максимального захисту даних.
- Брандмауери та системи виявлення вторгнень (IDS/IPS): Захист інфраструктури здійснюється за допомогою брандмауерів та систем виявлення та запобігання вторгнень, що забезпечує надійний захист від зовнішніх атак.
- Регулярні оновлення та патчі: Система постійно оновлюється, застосовуючи безпекові патчі для усунення потенційних вразливостей.
- Принцип найменшого привілею: Обмеження доступу до системи та даних відповідно до ролей користувачів, що дозволяє забезпечити контрольований доступ.

- Аудит та моніторинг: Систематичний аудит дій користувачів у системі допомагає виявляти та вживати заходів проти підозрілої активності.
- Обмеження доступу до API: Контроль доступу до API здійснюється через токени доступу, що забезпечує додатковий рівень захисту.
- Рейт-ліміти та квоти: Встановлення обмежень на частоту запитів до API допомагає запобігати DDoS-атакам та іншим формам зловмисної активності.
- Регулярне резервне копіювання: Важливі дані регулярно копіюються для забезпечення їх відновлення у випадку втрати.
- План відновлення після аварій: Розроблений план дій для швидкого відновлення системи у разі аварійних ситуацій.
- GDPR та інші нормативні акти: Наша система відповідає вимогам GDPR та інших відповідних законодавчих актів, забезпечуючи захист персональних даних користувачів.



Рисунок 3.2 – Необхідні заходи безпеки

Всі ці заходи безпеки впроваджені для забезпечення надійного функціонування нашої LXP (див. рис. 3.2), включаючи інтеграцію з зовнішніми системами та обробку конфіденційних даних. Вони створюють міцний фундамент для безпечного та ефективного навчального середовища.

#### 3.4. Методи тестування інтегрованих систем

У нашій роботі над інтеграцією різноманітних систем у освітню платформу, розроблену на Symfony, ми приділяємо велику увагу тестуванню, щоб забезпечити високий стандарт якості та надійності. Основним нашим інструментом є Codeception (рис. 3.3), який використовується для автоматизації всього процесу тестування, включаючи юніт, інтеграційне та системне тестування.



## Automated Testing Using Codeception

Рисунок 3.3 – Codeception

Починаючи з юніт-тестів, ми переконуємося, що кожен найдрібніший елемент нашої системи, будь то функція чи метод, працює бездоганно. Ці тести дозволяють нам ідентифікувати та виправити будь-які проблеми на ранніх стадіях розробки, значно скорочуючи час та зусилля, необхідні для виявлення помилок.

Перехід до інтеграційного тестування, ми зосереджуємося на взаємодії між різними модулями та компонентами системи. Це важливо, адже ми повинні бути впевнені, що наші інтегровані системи співпрацюють гармонійно, без конфліктів чи збоїв у їх взаємодії. Codeception тут знову стає незамінним, дозволяючи нам автоматизувати цей процес.

Системне тестування дозволяє нам оцінити роботу інтегрованої системи як єдиного цілого. Ми перевіряємо всі аспекти системи, від її функціональності та інтерфейсів до взаємодії з користувачами. Через acceptance тести ми визначаємо, чи задовольняє наш продукт очікуванням користувачів, що є вирішальним для успішного впровадження системи.

Наша команда регулярно переглядає та оновлює тестові сценарії, щоб вони відповідали останнім змінам у системі. Це постійне оновлення та тестування забезпечує високий рівень якості коду та гарантує стабільність та надійність нашої системи.

Використання цих методів тестування гарантує, що кожен елемент нашої інтегрованої системи працює належним чином, забезпечуючи стабільність, безпеку та високу продуктивність LXP.

### 3.5. Оптимізація продуктивності та надійності системи

Оптимізація продуктивності та надійності системи є важливою частиною розробки високоефективних веб-застосунків. Це включає в себе ряд заходів, спрямованих на підвищення швидкості роботи, зменшення часу відгуку, забезпечення стабільності та запобігання відмовам.

У процесі розробки LXP, ми зіткнулися з викликом оптимізації продуктивності та надійності системи, що є критично важливими аспектами для забезпечення ефективного веб-застосунку. Для досягнення цих цілей, ми впровадили цілеспрямований набір заходів, охоплюючи все від аналізу продуктивності до масштабування системи, використання сучасних технологій та методик.

Основою нашого підходу до аналізу продуктивності було використання інструментів моніторингу, таких як New Relic та Prometheus. Це дозволило нам відстежувати продуктивність системи в реальному часі та швидко реагувати на будь-які проблеми. Профілювання системи виявило "вузькі місця", що допомогло нам ефективно оптимізувати час виконання різних компонентів.

В контексті оптимізації продуктивності Symfony-базованих веб-додатків, аналіз продуктивності системи стає ключовим аспектом для забезпечення ефективної роботи та відповідності сучасним вимогам веб-розробки. Використання інструментів моніторингу, таких як New Relic та Prometheus, є фундаментальним для забезпечення цієї цілі.

При використанні New Relic та Prometheus у Symfony-базованих додатках, основна увага зосереджена на моніторингу продуктивності у реальному часі. Ці інструменти надають цінну інформацію щодо часу відгуку сервера, часу виконання запитів до бази даних та використання системних ресурсів. Це дозволяє оперативно виявляти недоліки у продуктивності та ідентифікувати "вузькі місця" у системі.

#### – Профілювання для оптимізації продуктивності

Профілювання системи в рамках Symfony-базованих додатків включає аналіз часу виконання різних компонентів додатку. Це може включати аналіз ефективності ORM запитів, виконання контролерів та сервісів, а також обробку шаблонів. Профілювання допомагає виявити частини коду, які потребують оптимізації, та відіграє ключову роль у поліпшенні загальної продуктивності додатку.

#### – Оптимізація баз даних

Оптимізація запитів до бази даних є важливою складовою підвищення продуктивності Symfony-додатків. Це включає індексацію баз даних, оптимізацію запитів, а також впровадження стратегій реплікації та шардування для покращення доступу до даних та забезпечення високого рівня надійності.

При розробці зосередилися на індексації та оптимізації SQL-запитів, щоб зменшити час відгуку. Реплікація та шардування даних були впроваджені для підвищення швидкості доступу та надійності. Ці заходи значно покращили продуктивність бази даних, що є критично важливим для великомасштабних систем.



- Кешування

Кешування на рівні сервера та клієнта грає важливу роль у зменшенні навантаження на сервери та покращенні часу відгуку. Використання розподілених кеш-систем, таких як Redis або Memcached, у Symfony-додатках забезпечує швидкий доступ до часто використовуваних даних, значно підвищуючи продуктивність.

Використання кешування допомогло зменшити навантаження на сервер. Розподілені кеш-системи, такі як Redis або Memcached, були використані для швидкого доступу до часто використовуваних даних.

- Оптимізація коду

Проведення рефакторингу та оптимізації коду, дозволить підвищити загальну ефективність системи. Асинхронне програмування було використане для поліпшення відгуку системи та забезпечення більш плавної взаємодії з користувачем.

Мінімізація та оптимізація ресурсів фронтенду, а також застосування стратегій лінивого завантаження допомагають покращити швидкість завантаження сторінок у Symfony-базованих додатках. Це забезпечує кращий досвід користувача та підвищує загальну продуктивність системи.

- Масштабування

Використання мікросервісної архітектури у сучасних веб-додатках, зокрема у контексті горизонтального масштабування, представляє собою стратегічний підхід до створення гнучких та надійних систем. Ця архітектура відрізняється від монолітного підходу тим, що вона розбиває додаток на менші, незалежні сервіси, кожен з яких відповідає за виконання певної функціональності.

– Забезпечення надійності

Регулярне резервне копіювання та плани відновлення після збоїв забезпечили високий рівень надійності системи. Використання кластерів та високодоступних систем забезпечило стабільність системи навіть при відмовах окремих компонентів.

– Автоматизація та моніторинг

Автоматизація розгортання з використанням Docker та Kubernetes спростила процеси розгортання та масштабування. Продовжений моніторинг стану системи, аналіз логів та виявлення аномалій дозволили нам оперативно реагувати на потенційні проблеми.

Оптимізація продуктивності та надійності системи вимагає комплексного підходу, який включає в себе аналіз продуктивності, оптимізацію баз даних, кешування, оптимізацію коду, масштабування, балансування навантаження, оптимізацію фронтенду, забезпечення надійності, автоматизацію та моніторинг. Ці заходи дозволяють не тільки підвищити продуктивність та швидкість роботи системи, але й забезпечити її стабільність та надійність, що є критично важливим для успішного функціонування веб-застосунків у сучасному цифровому світі.

### 3.6. Валідація функціональності та користувацького досвіду

Валідація функціональності та користувацького досвіду є вирішальним етапом у розробці нашої LXP для університетів (табл. 3.1). Цей процес не тільки забезпечує відповідність продукту всім технічним вимогам, але й гарантує, що він є інтуїтивно зрозумілим, зручним та ефективним для кінцевих користувачів.

Таблиця 3.1 – Етапи тестування функціональності та користувацького досвіду

Етап Тестування	Опис
Модульне Тестування	Перевірка окремих компонентів системи на відповідність технічним параметрам.
Інтеграційне Тестування	Тестування взаємодії між різними модулями та системами.
Системне Тестування	Оцінка повної функціональності системи в цілому.
Тестування UI/UX	Перевірка елементів інтерфейсу та загального досвіду користувачів.
Автоматизація Тестування	Використання інструментів як Selenium для автоматизації тестування.
Аналітика та Зворотний Зв'язок	Відстеження поведінки користувачів та збір зворотного зв'язку.
Адаптивність та Доступність	Тестування системи на різних пристроях та розмірах екранів.
Сценарії Використання	Моделювання реальних ситуацій для перевірки ефективності системи.
Тестування на Стрес	Оцінка стабільності системи при високих навантаженнях.

На початковому етапі валідації проводиться модульне тестування, яке зосереджено на перевірці окремих компонентів системи. Використовуючи технології Symfony та інші інструменти розробки, кожен модуль перевіряється на відповідність заданим технічним параметрам. Це дозволяє виявити та виправити помилки на ранніх етапах розробки, значно знижуючи ризики та витрати, пов'язані з подальшими змінами в системі.

Після модульного тестування проводиться інтеграційне тестування, яке забезпечує перевірку взаємодії між різними модулями та системами. Цей етап важливий для виявлення проблем інтеграції, особливо в контексті взаємодії з зовнішніми API та сервісами. Використання мікросервісної архітектури та інтеграція з Vue.js та Go дозволяють ефективно тестувати взаємодію між різними частинами системи.

Системне тестування оцінює повну функціональність системи в цілому. На цьому етапі перевіряється, як система працює в умовах, максимально наближених до реального використання. Це включає тестування всіх інтегрованих функцій, взаємодії з користувацьким інтерфейсом та загальної продуктивності системи.

Тестування UI та UX має велике значення для забезпечення зручності та інтуїтивності використання платформи. Використання інструментів як Selenium дозволяє автоматизувати тестування користувацького інтерфейсу, забезпечуючи швидке виявлення та виправлення проблем. Оцінюється кожен елемент інтерфейсу на предмет зрозумілості, легкості використання та загального досвіду користувачів при роботі з системою.

Автоматизація тестування є ключовим елементом у забезпеченні високої якості продукту. Використання CI/CD дозволяє автоматизувати процеси тестування та розгортання, забезпечуючи стабільність та якість системи. Це також сприяє швидкому впровадженню змін та оновлень у систему.

Використання аналітичних даних та зворотного зв'язку від користувачів є важливим для оптимізації системи. Аналітика дозволяє відстежувати поведінку користувачів, виявляти проблемні місця та аналізувати ефективність різних функцій. Зворотний зв'язок, зібраний через опитування та інтерв'ю, допомагає зрозуміти потреби та переваги користувачів.

Тестування на адаптивність та доступність забезпечує, що система ефективно працює на різних пристроях та розмірах екранів, а також доступна для всіх категорій користувачів. Це включає перевірку на сумісність з різними браузерами та операційними системами, а також забезпечення доступності для людей з обмеженими можливостями.

Розробка сценаріїв використання, що моделюють реальні ситуації, допомагає перевірити ефективність системи в різних умовах. Тестування на стрес оцінює стабільність системи при високих навантаженнях та в екстремальних умовах, забезпечуючи надійність та витривалість системи. Необхідно розробляти сценарії використання, що моделюють реальні ситуації, з якими можуть зіткнутися користувачі. Це допомагає перевірити ефективність системи в реальних умовах. Тестування на стрес дозволяє нам оцінити стабільність системи при високих навантаженнях та в екстремальних умовах.

Валідація функціональності та користувацького досвіду є важливою частиною процесу розробки, яка дозволяє не тільки виявити та виправити технічні недоліки, але й оптимізувати взаємодію користувачів з системою. Це забезпечує високу якість кінцевого продукту та задоволеність користувачів, що є ключовим для успіху будь-якого веб-застосунку.

## 4 ВПРОВАДЖЕННЯ, ОЦІНКА ЕФЕКТИВНОСТІ СИСТЕМИ ТА ПЕРСПЕКТИВИ РОЗВИТКУ

### 4.1. Планування впровадження системи в освітній процес

Впровадження системи управління навчальним процесом в освітній контекст вимагає ретельного планування та стратегічного підходу. Це забезпечує плавний перехід та максимальну ефективність використання системи. Основна мета полягає у максимізації користі від використання системи для всіх учасників навчального процесу.

- Розробка плану впровадження

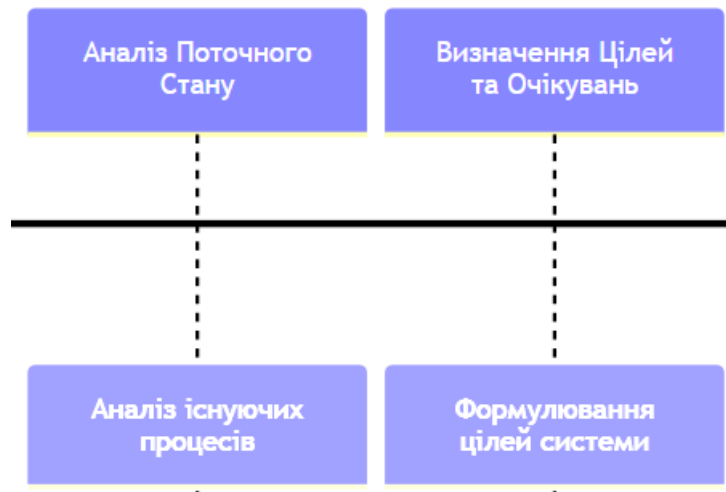


Рисунок 4.1 – Основні етапи плану впровадження

Першим кроком є аналіз існуючих процесів та систем у навчальному закладі. Це дозволяє визначити потреби, можливості для впровадження нової системи та виявити потенційні виклики.

Чітке формулювання цілей та очікувань від нової системи є ключовим для забезпечення її ефективності. Це включає визначення конкретних завдань, які має вирішити система, та бажаних результатів її впровадження.

– Залучення зацікавлених сторін

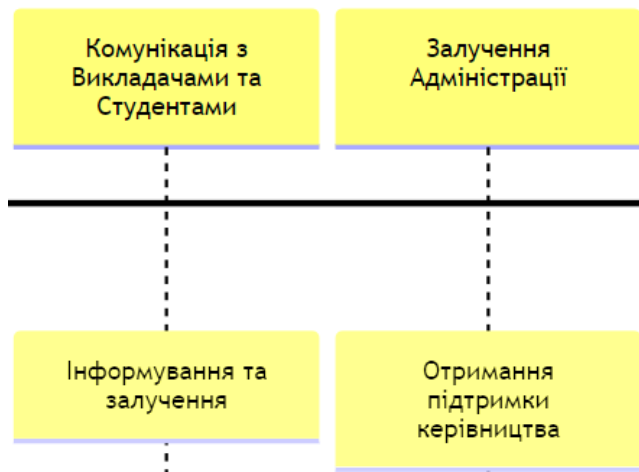


Рисунок 4.2 – Етапи залучення зацікавлених сторін

Інформування та залучення викладачів та студентів є важливим для забезпечення їх підтримки та активної участі у процесі впровадження.

Отримання підтримки від вищого керівництва є критичним для забезпечення необхідних ресурсів та політичної волі для впровадження системи.

– Технічна підготовка

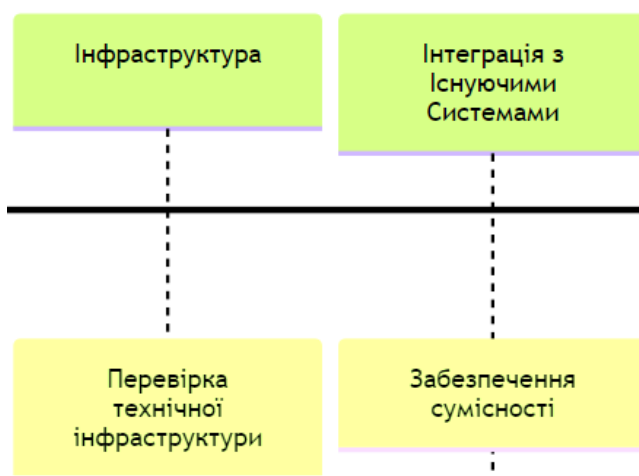


Рисунок 4.3 – Етапи технічної підготовки

Переконайтеся, що навчальний заклад має необхідну технічну інфраструктуру для підтримки нової системи, включаючи апаратне забезпечення та мережеву інфраструктуру.

Забезпечення сумісності нової системи з існуючими інформаційними системами навчального закладу є важливим для плавної інтеграції та ефективної взаємодії.

– Навчання та підтримка

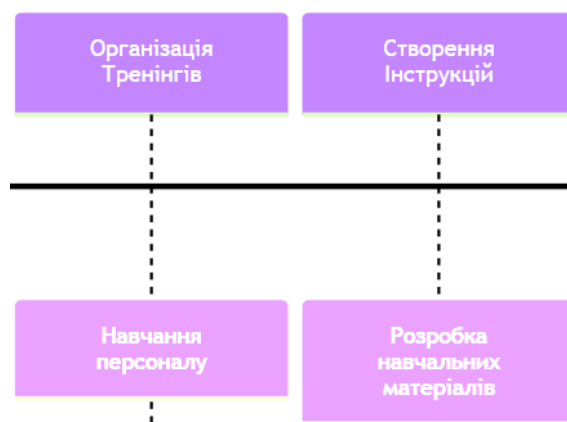


Рисунок 4.4 – Етапи навчання та підтримки користувачів

Навчання викладачів та адміністративного персоналу користуванню новою системою є важливим для забезпечення їх здатності ефективно використовувати її можливості.

Забезпечення доступу до детальних інструкцій та навчальних матеріалів допомагає користувачам самостійно ознайомитися з функціоналом системи.

– Пілотне впровадження



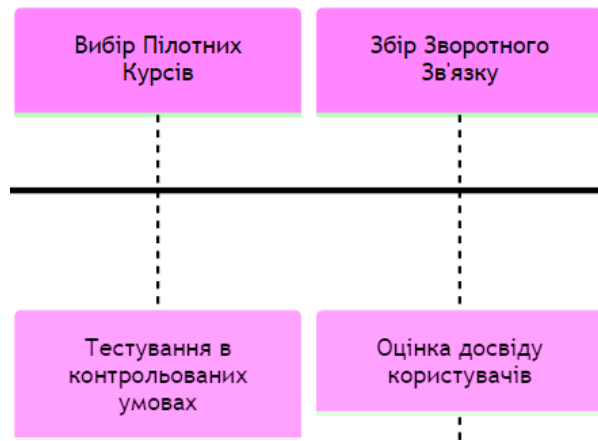


Рисунок 4.5 – Етапи пілотного впровадження

Початок впровадження з обмеженої кількості курсів дозволяє тестувати систему в контрольованих умовах та адаптувати її до потреб учбового закладу.

Оцінка досвіду користувачів та внесення необхідних коректив на основі зібраного зворотного зв'язку є важливим для оптимізації системи.

– Масштабування та повне впровадження

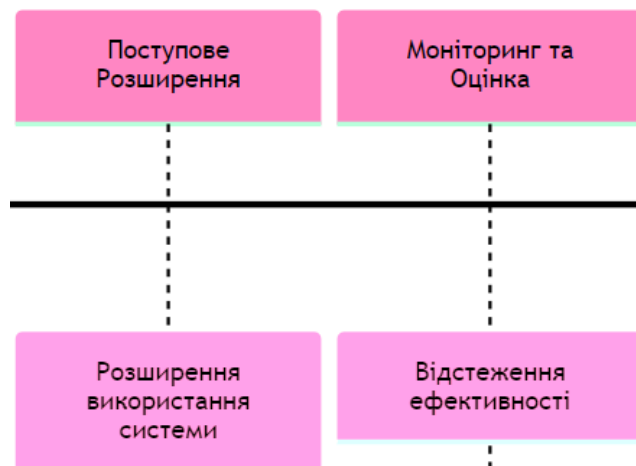


Рисунок 4.6 – Етапи масштабування

Розширення використання системи на більшу кількість курсів та програм дозволяє поступово інтегрувати її в усі аспекти навчального процесу.

Постійне відстеження ефективності системи та її впливу на навчальний процес є ключовим для забезпечення її тривалої ефективності.

– Забезпечення підтримки та оновлень

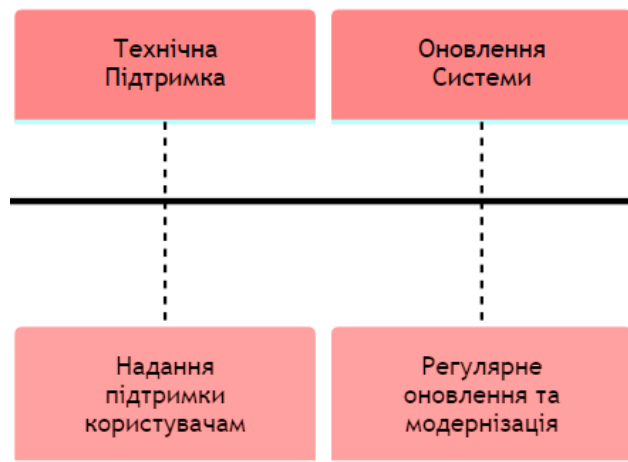


Рисунок 4.7 – Етапи забезпечення підтримки та оновлень

Надання постійної технічної підтримки користувачам є важливим для забезпечення безперебійної роботи системи.

Регулярне оновлення та модернізація системи забезпечує її актуальність та відповідність сучасним вимогам та тенденціям у сфері освіти.

Впровадження системи управління навчальним процесом в освітній контекст є складним, але водночас важливим завданням, що вимагає ретельного планування, залучення всіх зацікавлених сторін, а також постійного моніторингу та підтримки. Такий підхід забезпечує плавний перехід та максимальну користь від використання нової системи для всіх учасників навчального процесу.

4.2. Оцінка впливу системи на якість навчального процесу

Впровадження інноваційної системи управління навчальним процесом вносить значні зміни в освітнє середовище. Оцінка цього впливу є ключовою для розуміння ефективності системи та її впливу на якість навчання.

Після впровадження системи, важливо аналізувати, як вона впливає на різні аспекти навчального процесу. Це включає оцінку зручності використання інтерфейсу, доступності навчальних матеріалів, ефективності комунікаційних каналів між студентами та викладачами, а також загальної організації навчального процесу.

Одним з ключових показників ефективності системи є залученість студентів. Система повинна сприяти активній участі студентів у навчальному процесі, полегшувати доступ до ресурсів та підвищувати мотивацію до навчання. Оцінка залученості може включати аналіз активності студентів в системі, частоту використання навчальних матеріалів та участь у дискусіях та інших інтерактивних елементах.

Також важливо оцінити, як система впливає на роботу викладачів. Це може включати аналіз зручності планування курсів, ефективності оцінювання та зворотного зв'язку, а також загальної ефективності викладання. Викладачі повинні мати можливість легко керувати курсами, ефективно комунікувати зі студентами та використовувати аналітичні інструменти для оцінки успішності студентів.

Для оцінки впливу системи, необхідно збирати та аналізувати велику кількість даних. Це може включати дані про відвідуваність, успішність студентів, активність використання ресурсів та зворотний зв'язок від користувачів. Використання сучасних аналітичних інструментів дозволяє глибоко зануритися в ці дані та отримати цінні інсайти.

Оцінка задоволеності користувачів є важливою частиною процесу. Це включає проведення опитувань та інтерв'ю зі студентами та викладачами для збору зворотного зв'язку про їхній досвід використання системи. Такий зворотний зв'язок допомагає виявити сильні та слабкі сторони системи та визначити можливості для подальшого вдосконалення.

На основі зібраних даних та зворотного зв'язку, команда розробників може адаптувати та покращувати систему. Це може включати оптимізацію інтерфейсу, вдосконалення функціональності, а також розробку нових інструментів та можливостей, щоб краще задовольнити потреби користувачів.

Для забезпечення тривалої ефективності системи, важливо проводити постійний моніторинг її впливу на навчальний процес. Це дозволяє вчасно виявляти та реагувати на будь-які проблеми, а також адаптувати систему до змінюваних потреб та тенденцій в освіті.

Оцінка впливу системи на якість навчального процесу є комплексним завданням, яке вимагає збору та аналізу великої кількості даних, а також активної участі всіх зацікавлених сторін. Цей процес дозволяє не тільки оцінити ефективність системи, але й виявити можливості для її покращення та адаптації до потреб сучасного освітнього середовища.

#### 4.3. Збір та аналіз зворотного зв'язку від користувачів

Збір та аналіз зворотного зв'язку від користувачів є важливою частиною процесу вдосконалення та розвитку нашої платформи для управління навчальним процесом. Це дозволяє не тільки оцінити задоволеність користувачів, але й виявити потенційні проблеми та можливості для покращення.

Для збору зворотного зв'язку використовуються різноманітні методи, включаючи онлайн-опитування, інтерв'ю, фокус-групи, а також аналіз відгуків у соціальних мережах та на форумах (див. табл. 4.1). Онлайн-опитування є особливо ефективними, оскільки вони дозволяють швидко та ефективно збирати велику кількість даних.

Таблиця 4.1 – Етапи тестування функціональності та користувацького досвіду

Аспект Оцінки	Методи Оцінки
Зручність Використання Інтерфейсу	Перевірка інтуїтивності, легкості навігації
Ефективність Комунікаційних Каналів	Аналіз зв'язку між студентами та викладачами
Організація Навчального Процесу	Вивчення доступності ресурсів, гнучкості курсів
Залученість Студентів	Аналіз активності, участі у дискусіях
Ефективність Викладання	Оцінка зручності планування курсів, оцінювання
Аналіз Даних та Зворотній Зв'язок	Збір аналітичних даних, відстеження поведінки
Оцінка Задоволеності Користувачів	Проведення опитувань, інтерв'ю
Адаптація та Покращення	Оптимізація інтерфейсу, розробка нових функцій
Довгостроковий Моніторинг	Відстеження впливу системи, адаптація до змін

Отриманий зворотний зв'язок аналізується для виявлення ключових тенденцій та проблем. Важливо звертати увагу не тільки на кількісні, але й на якісні аспекти зворотного зв'язку. Аналіз коментарів та відгуків допомагає глибше зрозуміти досвід користувачів та їхні потреби.

На основі аналізу зворотного зв'язку команда розробників вносить необхідні зміни та покращення в систему. Це може включати оптимізацію інтерфейсу, вдосконалення функціональності, а також розробку нових інструментів та можливостей.

Залучення користувачів до процесу розвитку системи є важливим аспектом. Це може включати створення спільнот користувачів, де вони можуть ділитися ідеями та пропозиціями, а також участь у бета-тестуванні нових функцій.

Для більш глибокого аналізу зворотного зв'язку використовуються сучасні аналітичні інструменти. Це дозволяє автоматизувати процес збору та аналізу даних, а також отримувати більш точні та об'єктивні результати.

## ВИСНОВОК

Ця магістерська робота присвячена дослідженню інтеграції апаратних та програмних систем у веб-застосунки з метою розвитку сучасної платформи для управління навчальним процесом. Метою дослідження є створення інноваційної, ефективної та гнучкої системи, що відповідає сучасним вимогам та тенденціям в галузі EdTech.

Проведено глибокий аналіз сучасного стану інтеграції систем в освіті, визначено ключові принципи та методології інтеграції, а також висвітлено виклики та перспективи розвитку інтегрованих систем в освіті. Це дало змогу глибше зрозуміти контекст і потреби освітніх установ та виявити потенціал для впровадження нових технологічних рішень.

Основна увага була приділена визначенню функціональних вимог, аналізу потреб користувачів та розробці модульної архітектури системи. Вибір технологічного стеку, включаючи Symfony, Vue.js та Go, був обумовлений необхідністю забезпечити високу продуктивність, масштабованість та гнучкість системи.

Значну увагу було приділено методам інтеграції апаратних засобів, забезпеченню безпеки при інтеграції систем та оптимізації продуктивності та надійності. Валідація функціональності та користувацького досвіду дозволила оцінити ефективність інтегрованої системи та її прийняття кінцевими користувачами.

Планування впровадження системи, оцінка її впливу на якість навчального процесу, збір та аналіз зворотного зв'язку від користувачів дали змогу зробити висновки щодо реального впливу системи на освітній процес. Пропозиції щодо подальшого розвитку та масштабування системи підкреслюють її потенціал і відкривають шляхи для майбутніх досліджень.

Розробка такої системи є важливим кроком у напрямку модернізації навчального процесу, пропонуючи ефективне рішення, що відповідає потребам сучасних освітніх установ. Результати дослідження вказують на значний потенціал інтегрованих систем для підвищення ефективності управління навчанням, забезпечення гнучкості та адаптивності в освітніх процесах, а також на можливості їх подальшого розвитку та удосконалення.

Враховуючи динаміку розвитку технологій та зростаючі потреби в освітній сфері, важливо продовжувати дослідження та розробку нових підходів та рішень. Ця робота надає важливий внесок у розвиток сфери освіти, пропонуючи новітні ідеї та рішення для створення більш ефективних та інтерактивних освітніх систем.



**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Martin Fowler. Patterns of Enterprise Application Architecture. – Addison-Wesley Professional, 2002. – 533 с.
2. Sam Newman. Building Microservices: Designing Fine-Grained Systems. – O'Reilly Media, 2015. – 280 с.
3. Eric Evans. Domain-Driven Design: Tackling Complexity in the Heart of Software. – Addison-Wesley Professional, 2003. – 560 с.
4. Інтеграція апаратних та програмних систем у веб-застосунки. – Режим доступа:  
<https://www.techopedia.com/2/28587/integration/hardware-software-integration-in-web-applications>
5. Vue.js. Офіційна документація. – Режим доступа:  
<https://vuejs.org/v2/guide/>
6. Go Programming Language. Effective Go. – Режим доступа:  
[https://golang.org/doc/effective\\_go](https://golang.org/doc/effective_go)
7. Symfony Framework Documentation. – Режим доступа:  
<https://symfony.com/doc/current/index.html>
8. Петерсон, Л. Комп'ютерні мережі. / Л. Петерсон, Б. Дейвіс. – Видавництво: Вільямс, 2016. – 608 с.
9. Кібербезпека в освіті: Виклики та рішення. – Режим доступа:  
<https://cybersecurity-edu.org/challenges-and-solutions/>
10. Машинне навчання та його застосування в освіті. – Режим доступа:  
<https://www.edtechreview.in/trends-insights/insights/3324-machine-learning-application-in-education>

11. Міжнародні стандарти інтеграції освітніх технологій. – Режим доступу: <https://www.iso.org/education>
12. Безпека API та її значення у веб-застосунках. – Режим доступу: [https://www.owasp.org/index.php/API\\_Security](https://www.owasp.org/index.php/API_Security)
13. Оцінка ефективності веб-застосунків. – Режим доступу: <https://www.webapp-performance-evaluation.com>
14. Концепція мікросервісів та їх впровадження. – Режим доступу: <https://microservices.io>
15. Системи керування базами даних: Теорія та практика. / М. Стоунбрейкер, Д.Мур. – Видавництво: Комп'ютерПресс, 2017. – 512 с.
16. Штучний інтелект у освіті: Перспективи розвитку. – Режим доступу: <https://www.ai-in-education.org/future-perspectives>

## Додаток А

### Технічна документація системи

#### А.1 Опис модулів

- Модуль управління користувачами

Функції:

Реєстрація та аутентифікація користувачів.

Налаштування ролей (студент, викладач, адміністратор).

Управління правами доступу для різних ролей.

Можливість налаштування індивідуальних профілів.

- Модуль управління курсами

Функції:

Створення та редагування курсів.

Управління навчальними матеріалами (додавання тексту, мультимедіа).

Відстеження прогресу студентів та ведення статистики.

Інтеграція з інтерактивними дошками та іншими освітніми інструментами.

- Модуль поставки контенту

Функції:

Підтримка мультимедійного контенту.

Сумісність з форматами SCORM та xAPI.

Персоналізація навчального досвіду на основі інтерактивних вікторин та завдань.

- Модуль оцінки та вікторин

Функції:

Створення різноманітних типів вікторин та тестів.

Автоматична оцінка та зворотний зв'язок.

Аналітика успішності студентів.

– Модуль співпраці та комунікації

Функції:

Форуми для обговорення та чати для миттєвого спілкування.

Віртуальні аудиторії для інтерактивних сесій.

Спільні документи та інструменти для групової роботи.

– Модуль аналітики та звітності

Функції:

Збір та аналіз даних про взаємодію та успішність студентів.

Візуалізація результатів у формі звітів та графіків.

Інструменти для моніторингу прогресу та виявлення проблемних областей.

– Модуль сумісності та мобільності

Функції:

Адаптація платформи для різних пристроїв та операційних систем.

Оптимізація інтерфейсу для мобільних пристроїв.

Підтримка офлайн-доступу до матеріалів.

– Модуль безпеки та приватності

Функції:

Захист даних користувачів та шифрування з'єднань.

Регулярне оновлення безпеки та моніторинг системи.

Дотримання стандартів конфіденційності та захисту персональних даних.

## Додаток Б

### Користувацькі інструкції

#### Б. 1 – інструкція до Mentor Api Application

##### 1. Конфігурація системи:

APP\_ENV: Визначає поточне середовище сервіса. Можливі значення: prod, stage, dev, test.

APP\_DEBUG: Включення або виключення debug-режиму. Значення: 0 або 1.

APP\_SECRET: Необхідно для збільшення ентропії у безпекових операціях. Це обов'язковий параметр Symfony. Наприклад: 823bhjsdaf0123123basd.

##### 2. База даних:

COMMAND\_DATABASE\_HOST: Наприклад: engine.db, localhost.

COMMAND\_DATABASE\_PORT: Наприклад: 5432.

COMMAND\_DATABASE\_NAME: Наприклад: engine.

COMMAND\_DATABASE\_USER: Наприклад: engine\_usr.

COMMAND\_DATABASE\_PASSWORD: Наприклад: 123qwe.

Параметри бази даних для тестового середовища (test env):

TEST\_DATABASE\_HOST

TEST\_DATABASE\_PORT

TEST\_DATABASE\_NAME

TEST\_DATABASE\_USER

TEST\_DATABASE\_PASSWORD

#### 4. Встановлення та запуск:

- Копіювання конфігурації: Скопіюйте файл конфігурації за допомогою команди `cp .env.dist .env`.
- Збірка Docker: Виконайте команду `docker-compose build` для збірки Docker-контейнерів.
- Запуск проекту: Використовуйте команду `make up` для запуску проекту.
- Вхід у консоль PHP контейнера: Для виконання команд у контейнері використовуйте `make exec`. Після цього встановіть залежності за допомогою `composer install` та проведіть міграцію бази даних командою `bin/console doctrine:migration:migrate`.

**ДОДАТОК В****Вихідний програмний код****Г.1 – класс сутності**

```
<?php
namespace App\Entity;

use App\Repository\CourseRepository;
use Doctrine\Common\Collections\ArrayCollection;
use Doctrine\Common\Collections\Collection;
use Doctrine\DBAL\Types\Types;
use Doctrine\ORM\Mapping as ORM;

#[ORM\Entity(repositoryClass: CourseRepository::class)]
class Course
{
    #[ORM\Id]
    #[ORM\GeneratedValue]
    #[ORM\Column]
    private ?int $id = null;

    #[ORM\Column(length: 255)]
    private ?string $name = null;

    #[ORM\Column(type: Types::TEXT, nullable: true)]
    private ?string $description = null;

    #[ORM\Column]
    private ?float $price = null;

    #[ORM\Column]
    private ?\DateTimeImmutable $createdAt = null;
```

```
#[ORM\Column]
private ?\DateTimeImmutable $updatedAt = null;

#[ORM\Column(nullable: true)]
private ?\DateTimeImmutable $deletedAt = null;

#[ORM\OneToMany(mappedBy: 'course', targetEntity: Module::class)]
private Collection $modules;

#[ORM\OneToMany(mappedBy: 'course', targetEntity:
CourseTranslation::class)]
private Collection $courseTranslations;

public function __construct()
{
    $this->modules = new ArrayCollection();
    $this->courseTranslations = new ArrayCollection();
}

public function getId(): ?int
{
    return $this->id;
}

public function getName(): ?string
{
    return $this->name;
}

public function setName(string $name): self
{
    $this->name = $name;

    return $this;
}
```



```
}

public function getDescription(): ?string
{
    return $this->description;
}

public function setDescription(?string $description): self
{
    $this->description = $description;

    return $this;
}

public function getPrice(): ?float
{
    return $this->price;
}

public function setPrice(float $price): self
{
    $this->price = $price;

    return $this;
}

public function getCreatedAt(): ?\DateTimeImmutable
{
    return $this->createdAt;
}

public function setCreatedAt(\DateTimeImmutable $createdAt): self
{
    $this->createdAt = $createdAt;
}
```

```
        return $this;
    }

    public function getUpdatedAt(): ?\DateTimeImmutable
    {
        return $this->updatedAt;
    }

    public function setUpdatedAt(\DateTimeImmutable $updatedAt): self
    {
        $this->updatedAt = $updatedAt;

        return $this;
    }

    public function getDeletedAt(): ?\DateTimeImmutable
    {
        return $this->deletedAt;
    }

    public function setDeletedAt(?\DateTimeImmutable $deletedAt):
self
    {
        $this->deletedAt = $deletedAt;

        return $this;
    }

    public function getModules(): Collection
    {
        return $this->modules;
    }
}
```

```

public function addModule(Module $module): self
{
    if (!$this->modules->contains($module)) {
        $this->modules->add($module);
        $module->setCourse($this);
    }

    return $this;
}

public function removeModule(Module $module): self
{
    if ($this->modules->removeElement($module)) {
        // set the owning side to null (unless already changed)
        if ($module->getCourse() === $this) {
            $module->setCourse(null);
        }
    }

    return $this;
}

/**
 * @return Collection<int, CourseTranslation>
 */
public function getCourseTranslations(): Collection
{
    return $this->courseTranslations;
}

public function addCourseTranslation(CourseTranslation
$courseTranslation): self
{

```

```
        if (!$this->courseTranslations->contains($courseTranslation))
    {
        $this->courseTranslations->add($courseTranslation);
        $courseTranslation->setCourse($this);
    }

    return $this;
}

    public function removeCourseTranslation(CourseTranslation
$courseTranslation): self
    {
        if
($this->courseTranslations->removeElement($courseTranslation)) {
            // set the owning side to null (unless already changed)
            if ($courseTranslation->getCourse() === $this) {
                $courseTranslation->setCourse(null);
            }
        }

        return $this;
    }
}
```

## Г. 2 – код CRUD контроллеру з використанням CQRS

```
<?php

namespace App\Controller\Api\Admin;

use App\Controller\BaseRestController;
use App\Exception\Admin\Crud\EntityNotFoundException;
use App\Exception\ValidationException;
use App\Service\Common\Validator\ValidatorService;

use
App\Service\Language\Factory\Command\CreateLanguageCommandFactory;

use
App\Service\Language\Factory\Command\UpdateLanguageCommandFactory;

use
App\Service\Language\Factory\Request\CreateLanguageRequestFactory;

use App\Service\Language\LanguageCommandService;
use App\Service\Language\LanguageQueryService;
use Symfony\Component\HttpFoundation\JsonResponse;
use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;

class LanguageController extends BaseRestController
{
    #[Route('/admin/language', name: 'admin_language_list',
methods: ['GET'])]

    public function list(Request $request, LanguageQueryService
$languageQueryService): JsonResponse
```

```

{
    return $this->returnResponse($request,
$languageQueryService->getList());
}

/**
 * @throws ValidationException
 */

#[Route('/admin/language/create', name:
'admin_language_create', methods: ['POST'])]
public function create(
    Request $request,
    CreateLanguageRequestFactory $requestFactory,
    ValidatorService $validator,
    LanguageCommandService $commandService,
    CreateLanguageCommandFactory $commandFactory
): JsonResponse {
    $requestData = $request->toArray();

    $requestDTO = $requestFactory->create($requestData);

    $validator->validateWithThrowsException($requestDTO);

    $command = $commandFactory->create($requestData);

    $commandService->create($command);

    return $this->returnResponse($request, []);
}

```

```

    }

        #[Route('/admin/language/{identifier}', name:
'admin_language_show', methods: ['GET'])]

    public function show(

        int $identifier,

        Request $request,

        LanguageQueryService $queryService,

    ): JsonResponse {

        $language = $queryService->getItem($identifier);

        return $this->returnResponse($request, $language);

    }

/**

 * @throws ValidationException

 */

        #[Route('/admin/language/update', name:
'admin_language_update', methods: ['POST'])]

    public function update(

        Request $request,

        UpdateLanguageCommandFactory $requestFactory,

        ValidatorService $validator,

        LanguageCommandService $commandService,

        UpdateLanguageCommandFactory $commandFactory

    ): JsonResponse {

        $requestData = $request->toArray();

```

```

$requestDTO = $requestFactory->create($requestData);

$validator->validateWithThrowsException($requestDTO);

$command = $commandFactory->create($requestData);

try {
    $commandService->update($command);
} catch (EntityNotFoundException $e) {
    return $this->returnError($e->getMessage(),
Response::HTTP_NOT_FOUND);
}

return $this->returnResponse($request, []);
}

#[Route('/admin/language/{identifier}/delete', name:
'admin_language_delete', methods: ['DELETE'])]
public function delete(
    int $identifier,
    Request $request,
    LanguageCommandService $languageCommandService,
): JsonResponse {
    try {
        $languageCommandService->delete($identifier);
    } catch (EntityNotFoundException $e) {
        return $this->returnError($e->getMessage(),
Response::HTTP_NOT_FOUND);
}

```



```

    }

    return $this->returnResponse($request, []);
}
}

```

### Г. 3 – код Google OAuth2.0 контролеру

```
<?php
```

```

namespace App\Controller\Auth;

use KnpU\OAuth2ClientBundle\Client\ClientRegistry;
use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\Routing\Annotation\Route;

class GoogleController extends AbstractController
{
    /**
     * Link to this controller to start the "connect" process.
     *
     * @Route("/connect/google", name="connect_google_start")
     */
    public function connectAction(ClientRegistry $clientRegistry)
    {
        // will redirect to Google!
        return $clientRegistry
            ->getClient('google_main') // key used in
            config/packages/knpu_oauth2_client.yaml
            ->redirect(
                [
                    'profile',
                    'email', // the scopes you want to access
                ]
            );
    }
}

```

```
        ]
    );
}

/**
 * After going to Google, you're redirected back here
 * because this is the "redirect_route" you configured
 * in config/packages/knpu_oauth2_client.yaml.
 *
 * @Route("/connect/google/check", name="connect_google_check")
 */
public function connectCheckAction(Request $request,
ClientRegistry $clientRegistry)
{
    return $this->redirectToRoute('index');
}
}
```